Gilbert Babin
Katarina Stanoevska-Slabeva
Peter Kropf (Eds.)

E-Technologies: Transformation in a Connected World

LNBIP 78

5th International Conference, MCETECH 2011
Les Diablerets, Switzerland, January 2011
Revised Selected Papers

Springer

Lecture Notes
in Business Information Processing 78

Gilbert Babin
Katarina Stanoevska-Slabeva
Peter Kropf (Eds.)

E-Technologies: Transformation in a Connected World

5th International Conference, MCETECH 2011
Les Diablerets, Switzerland, January 23-26, 2011
Revised Selected Papers

Springer

Volume Editors

Gilbert Babin
HEC Montréal
Information Technologies
Montréal, QC, H3T 2A7, Canada
E-mail: Gilbert.Babin@hec.ca

Katarina Stanoevska-Slabeva
University of Neuchâtel
Academy of Journalism and Media
2000 Neuchâtel Switzerland
E-mail: Katarina.Stanoevska@unine.ch

Peter Kropf
University of Neuchâtel
Institute of Computer Science
2000 Neuchâtel Switzerland
E-mail: Peter.Kropf@unine.ch

ISSN 1865-1348 e-ISSN 1865-1356
ISBN 978-3-642-20861-4 e-ISBN 978-3-642-20862-1
DOI 10.1007/978-3-642-20862-1
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2011926504

ACM Computing Classification (1998): J.1, K.4.4, H.4

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

The Internet and the Web are continuously evolving. Their changing incarnations such as Web 2.0, services and service-oriented architectures, cloud computing, and convergence with mobile Internet are transforming the way traditional activities are undertaken and are having a dramatic impact on many aspects of modern society. Companies, governments, and users are continuously challenged to follow up and take advantage of the potential benefits and power of digital technologies. Successful transformation to meet new challenges and opportunities is a multi-faceted problem, involving vision and skills, but also many technological, managerial, economic, organizational, and legal issues.

In this fifth edition of the International MCETECH conference, held in Les Diablerets (Switzerland) during January 23–26, 2001, researchers and practitioners were asked to think about this notion of transformation and its relationship to e-commerce. A total of 32 papers were submitted on topics ranging from process modelling to e-business, including presentations on eHealth, eEducation, and eGovernment. Out of these papers, ten were accepted with minor revisions. The authors of eight papers were asked to revise and resubmit their papers. We assigned a committee member to each of these papers to guide the authors in their review process. This resulted in an additional seven papers being accepted in the proceedings. The final program included 17 papers.

The main scientific conference program of MCETECH 2011 was held in parallel with the CUSO Winter School in Computer Science that focussed on managing and engineering complex systems.

We thank all the authors who submitted papers, the Program Committee members, and the external reviewers. We express our gratitude to the Steering Committee Chair, Hafedh Mili, for his enthusiasm and his invaluable help in preparing this conference. We also thank all the local people who were instrumental in making this edition of MCETECH another very successful event. In particular, we are very grateful to Alain Sandoz, who was responsible for the local arrangements. Furthermore, we thank Étienne Rivière, who organized publicity, and the many students who volunteered on the organization team.

January 2011

Gilbert Babin
Katarina Stanoevska-Slabeva
Peter Kropf

Organization

The 5th International MCETECH Conference on e-Technologies (MCETECH 2011) was sponsored by Université de Neuchâtel, HEC Montréal, Université du Québec à Montréal, and Biodiversity Conservation Information Systems.

Conference Chair

Peter Kropf	Université de Neuchâtel, Switzerland

Steering Committee Chair

Hafedh Mili	Université du Québec à Montréal, Canada

Program Committee Co-chairs

Gilbert Babin	HEC Montréal, Canada
Katarina Stanoevska-Slabeva	Université de Neuchâtel, Switzerland

Program Committee

Kamel Adi	Université du Québec en Outaouais, Canada
Esma Aimeur	Université de Montréal, Canada
Jörn Altmann	Seoul National University, Korea
Daniel Amyot	University of Ottawa, Canada
Andrea Back	University of St.Gallen, Switzerland
Jean-Paul Barthès	Université de Technologie de Compiègne, France
Sarita Bassil	Marshall University, USA
Morad Benyoucef	University of Ottawa, Canada
Yasmine Charif	Université du Québec à Montréal, Canada
Bernard Corbineau	Université Paris-Est Marne-la-Vallée, France
Peter Emmel	SAP, Germany
Michael Franz	University of California, Irvine, USA
Stéphane Gagnon	Université du Québec en Outaouais, Canada
Olivier Glassey	IDHEAP Lausanne, Switzerland
Christoph Glauser	IFAAR Bern, Switzerland
Ning Gu	Fudan University, China
Felix Hampe	University of Koblenz-Landau, Germany
Paul Hofmann	SAP, USA
Gregory Kersten	Concordia University, Canada

Lamia Labed Jilani	Institut supérieur de Gestion, Tunis, Tunisia
Anne-Françoise Le Meur	Université de Lille, France
Luigi Logrippo	Université du Québec en Outaouais, Canada
Simone Ludwig	University of Saskatchewan, Canada
Morteza Niktash	Public Works and Government Services, Canada
Susanne Patig	University of Bern, Switzerland
Liam Peyton	University of Ottawa - COGNOS, Canada
Andreja Pucihar	University of Maribor, Slovenia
Roy Rada	University of Maryland, USA
Reinhard Riedl	Berner Fachhochschule, Switzerland
Étienne Rivière	Université de Neuchâtel, Switzerland
Alain Sandoz	Université de Neuchâtel, Switzerland
Hans Schotten	University of Kaiserslautern, Germany
Michael Spahn	SAP, Germany
Thomas Tran	University of Ottawa, Canada
Ulrich Ultes-Nitche	Université de Fribourg, Switzerland
Theodora Varvarigou	National Technical University of Athens, Greece
Michael Weiss	Carleton University, Canada
Yuhong Yan	Concordia University, Canada
Hans-Dieter Zimmermann	University of Applied Sciences St. Gallen, Switzerland
Christian Zirpins	University College London, UK

Publicity Committee Chair

Étienne Rivière	Université de Neuchâtel, Switzerland

Local Arrangements Committee Chair

Alain Sandoz	Université de Neuchâtel, Switzerland

Table of Contents

Session 5: eHealth, eEducation, and eGovernment II

Session 6: Internet-Based Collaborative Work

Session 7: eBusiness

Invited Papers

A Systematic Approach to Web Application Penetration Testing Using TTCN-3

Bernard Stepien, Pulei Xiong, and Liam Peyton

School of Information Technology and Engineering,
University of Ottawa, Canada
{bernard,xiong,lpeyton}@site.uottawa.ca

Abstract. Penetration testing is critical for ensuring web application security. It is often implemented using traditional 3GL web test frameworks (e.g. HttpUnit, HtmlUnit). There is little awareness in the literature that a test specification language like TTCN-3 can be effectively combined with such frameworks. In this paper, we identify the essential aspects of TTCN-3 for penetration testing and how best to use them. These include separating abstract test logic from concrete data extraction logic, as well as support for templates, matching test oracles and parallel test components. The advantages of leveraging TTCN-3 together with 3GL web test frameworks for penetration testing is demonstrated and evaluated using example scenarios. The work was performed with a prototype TTCN-3 tool that extends the TTCN-3 model architecture to support the required integration with 3GL web test frameworks. A concrete proposal for modifying the TTCN-3 standard to support this refinement is described.

Keywords: web application security, model-based testing, penetration testing, test specification, TTCN-3.

1 Introduction

Web application vulnerabilities have been exploited since the early '90s against user oriented applications such as email, online shopping, and Web banking. Testing for web application vulnerabilities continues to be a significant problem, as more and more user-oriented applications are deployed to the web such as Facebook and Twitter Blog. It is often implemented using traditional 3GL web test frameworks (e.g. HttpUnit [11], HtmlUnit [10], JUnit [12]). There is little awareness in the literature that a test specification language like TTCN-3 [5] can be effectively combined with such frameworks. In this paper, we identify the essential aspects of TTCN-3 for penetration testing and how best to use them. These include separating abstract test logic from concrete data extraction logic, as well as support for templates, matching test oracles and parallel test components.

The use of a test specification language like TTCN-3 can improve the quality of the test oracles or assertions. General purpose language (GPL) approaches to penetration testing tend to be problematic because test oracles have to be pre-defined, and verification is limited to spot checking on a limited number of web page elements. As a result of this, confidence is reduced on the completeness of the test results which

G. Babin, K. Stanoevska-Slabeva, P. Kropf (Eds.): MCETECH 2011, LNBIP 78, pp. 1–16, 2011.

means there is always a lingering concern that some vulnerability has gone undetected. As well, there is poor test automation re-usability both for a given application (regression testing) and between various applications (conceptual generalization).

The advantages of leveraging TTCN-3 together with 3GL web test frameworks for penetration testing is demonstrated and evaluated in this paper in section 2 (separating test logic from data extraction logic) and section 3 (test oracles) using two concrete attacks against the web application vulnerabilities: SQL Injection [17] (an attack occurs on server-side) and Persistent Cross Site Scripting [21](an attack occurs on client-side).

The work was performed with a prototype TTCN-3 tool that extends the TTCN-3 model architecture to support the required integration with 3GL web test frameworks. A concrete proposal for modifying the TTCN-3 standard to support this refinement is described in section 4. An existing TTCN-3 vendor has already incorporated the changes in the latest version of their tool.

2 Background and Related Work

A vulnerability is a bug or misconfiguration that can be exploited [14]. Penetration testing detects vulnerabilities in a system by attempting to recreate what a real attacker would do [22]. Penetration testing is often implemented using a general purpose language combined with test frameworks such as Metasploit [15], AttackAPI [3], as well as special browser extensions e.g. Firebug [8] and GreaseMonkey [9]. Usually a general purpose programming language is used with the frameworks and specialized tools to automate test execution.

A description of current approaches to penetration testing can be found in [16, 1, 18]. There are many factors that affect test case coverage and quality of testing. In [2] it was found that the tester's knowledge, skills and experience are factors. The resources available to testers are also relevant [4]. Other research has proposed test methodology changes to ensure testing is conducted more systematically and efficiently by, for example, integrating penetration testing into a security-oriented development life cycle [19]. Our use of test specifications written in TTCN-3 fits well with this approach. [13] provides an example of passive intrusion testing using TTCN-3 and [20] provides an analysis of the problem but no actual implementation examples and discussion. In [24], penetration testing is driven by a process that starts with threat modeling.

3 Problem Description

There are two basic tasks in web application testing:

- Specifying test cases, including test actions usually in the form of http requests with test oracles implemented as assertions
- Extracting data from responses written or dynamically generated in HTML format

A number of frameworks [10, 11] are available that can create test cases by simulating user actions in a web browser by among other things executing scripting functions

on the client side. They also have features to reduce the effort of data extraction and requests submission. However, the two above tasks end up being intermingled as shown for example on the HtmlUnit website [10] where the implementation details of extracting the title from the HTML code are intermingled with specifying the expected response that constitutes the test logic.

```
public void testHtmlUnitHomePage() throws Exception {
    final WebClient webClient = new WebClient();
    final URL url = new URL("http://HtmlUnit.sourceforge.net");
    final HtmlPage page = (HtmlPage)webClient.getPage(url);

   assertEquals( "HtmlUnit - Welcome to HtmlUnit", page.getTitleText());
  }
```

Such a strong coupling of data extraction and test assertion tasks (see figure 1a) makes test cases more complex to write and harder to understand. It also makes test specification heavily dependent on the particular tools used for data extraction, which can be problematic if one wishes to migrate to a different tool. In industrial applications we have worked on, we have found that data extraction framework related statements represented about 80% of the total test script source code.

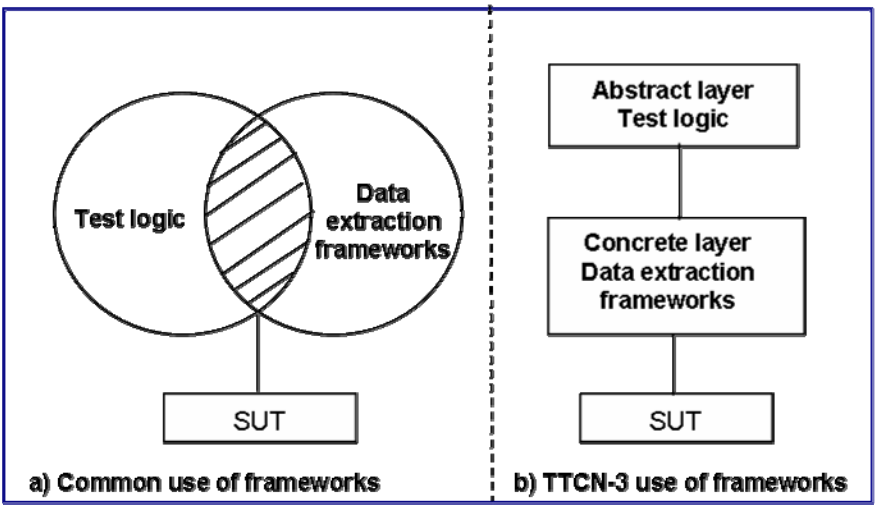


Fig. 1. Use of data extraction frameworks approaches

The use of a test language like TTCN-3 addresses this issue by clearly separating test logic from data extraction logic (see figure 1b). There is an abstract layer where test assertions are specified and a concrete layer that handles communication with the SUT including data extraction. While the abstract layer uses a very powerful matching mechanism that allows composing very complex assertions, the concrete layer can use any general purpose language (GPL) features to perform its tasks, including the type of frameworks used in our example. The significant advantage is that test logic

written in the TTCN-3 abstract layer becomes independent of the framework used for data extraction and communication.

Another important, but perhaps less obvious, advantage of separating test logic from data extraction logic is that it promotes a more efficient programming style by factoring out repeated operations. In theory, one could manually create such factoring using a general purpose language, but with TTCN-3 it is ensured by the TTCN-3 model architecture. As was shown in [23], the first task when testing using TTCN-3 consists of modeling a web page using an abstract data type definition. It consists of mapping each abstract data type to a function in the codec that performs the coding and decoding of concrete data. Consequently, the above example of testing for a title page will result in only a single function that handles the title page extraction.

In the following abstract representation of a web page found in [23]:

```
type record WebPageType {
        integer statusCode,
        charstring title,
        charstring content,
        LinkListType links optional,
        FormSetType forms optional,
        TableSetType tables optional
}
```

The above *WebPageType* data type drives the codec to invoke the concrete *decodePage*() method that processes a web response and populates an instance of the data type. Each element is processed and for example the title of the web page is obtained using the *getTitleText*() method of the *HtmlUnit* framework *WebClient* class.

```
public RecordValue decodePage(HtmlPage theCurrentPage) {

   RecordValue theResponseValue = (RecordValue)
     typeServer.getTypeForName("HtmlTypes.WebPageType").newInstance();

   …
   String title = theCurrentPage.getTitleText(); // HtmlUnit
   CharstringValue titleValue = (CharstringValue)
                            typeServer.getCharstring().newInstance();
   titleValue.setString(title);
   theResponseValue.setField("title", titleValue);
   …
}
```

The difference with the traditional GPL/frameworks approach is that the above *decodePage()* method is well separated from the test logic because it is located in the concrete layer that is generic in the sense that it is not dependant on a particular web application and thus makes it fully re-usable for any other web application. This is precisely not the case when data extraction and test logic functionalities are intermingled. The additional benefit is that these data extraction functionalities are completely transparent at the TTCN-3 abstract layer. Thus, the additional benefit of using TTCN-3 is to further factor out some code and place it in a framework. The abstract data typing and related codec for web pages inherently constitutes a framework that can be endlessly re-used for various web pages within a web application or for any

new web application as we had already done in [23]. Thus, the only things that remain to be coded by the tester are the assertions that are specified exclusively in the abstract layer using the central TTCN-3 concept of template. Here it is important to stress that, in theory, the test coder could have written a data extraction framework of his own in a GPL, but this is rarely done as there is no built in structured support for it. In TTCN-3 the separation of test logic in an abstract layer from data extraction logic forces the coder to use this more efficient structuring mechanism.

4 Specification of Test Oracles in TTCN-3

Test oracles are specified in TTCN-3 using the concept of a template. A TTCN-3 template is a mechanism for combining multiple assertions into a single operation, thus it is a structured assertion. For example, for web pages, test code written in a traditional general purpose language would use a sequence of independent assertions that will stop at the first failure of one single assertion while TTCN-3 can relate assertions to abstract data types that represent all the essential elements of a web page thus extending the concept of structured data types to the concept of structured assertion. Thus, all assertions composing a template are verified at once and they are all verified whether one fails or not. This gives a full picture of what could be wrong in a given web response. A TTCN-3 template can be best described by comparing it to an XML document with the difference that it contains assertions rather than data. The TTCN-3 matching mechanism also enables one to specify a single structured assertion for an entire web page. This can include complex tables, links or forms using our abstract type for web pages. For example, a test oracle for a web page table can be hard coded as follows. The value assignments (using the ":=" operator) implicitly mean should be equal to, and are therefore an assertion in the traditional JUnit sense:

```
template TableType statements_table_t := {
   rows := {
     {cells := {"date", "description", "amount", "kind"}},
     {cells := {"2009-07-10", "check # 235", "2491.89", "DB"}},
     {cells := {"2009-07-02", "salary ACME", "5000.23", "CR"}},
     {cells := {"2009-06-28", "transfer to savings", "500.0", "DB"}}
   }
}
```

However, hard-coded templates such as this one can require tedious efforts to create and require significant maintenance effort, which makes them not very re-usable.

In the rest of this section we explain the principle of self-definition for test oracles in 4.1, the sharing of test oracles among actors in 4.2 to address parallel execution, and their foundation for reuse in 4.3.

4.1 Test Oracle Self-defining Principle

To illustrate our approach we use a simple penetration testing example that consists of checking if one can illegally login to an application and thus land inside the application that is characterized by a specific web page. This consists in submitting a form filled with the login information and perform an assertion on the content of the specific web page.

We avoid hard coding test oracles by leveraging the concept of template in TTCN-3. The TTCN-3 template is used both for matching incoming data against a test oracle and passing data from the concrete layer to an abstract variable that can be in turn used as a template to be matched against further responses data. In the context of a web application, an abstract response that is the result of one login is used as a test oracle to be matched against the response for another login attempt on the same login form. This approach appears seemingly trivial at the abstract level but relies on a complex infrastructure provided by TTCN-3. It consists of two steps:

- Perform a login using a legitimate user id and password and obtain the normal response page content that we save by assigning it to a TTCN-3 template variable.
- Perform a login using an illegitimate password (SQL injection or stolen from a cookie as in XSS) on the same user id and use the response content from the legitimate login that was stored in a variable as a test oracle against the response from the illegitimate login.

If the legitimate response content of the first login attempt response matches the response to the illegitimate login, there is potential penetration vulnerability. The interesting aspect of this approach is that at no time do we need to explicitly specify the test oracle for the response, thus by definition, no hard coding needs to be performed. As a matter of fact, the tester does not even need to know what the content of the response exactly is. Minimal hard coding could still be used to avoid false positives. For example, checking a return code of 200 [25] or checking the title of the page could increase confidence. This can be easily implemented due to the fact that the template used is now stored as a structured variable where fields can be modified using invariant values such as the 200 return code. All of this is based on the ambiguity between variables that in a GPL can only contain data and that in TTCN-3 can contain templates that are really functions that perform the matching. The template variable allows modifying individual functions rather than just plain data.

Another important difference with traditional testing is that responses are fully assembled in the TTCN-3 concrete layer by the codec that is part of our TTCN-3 framework and thus does not need to be developed for each web application. It can be re-used for any other web application.

The capability of TTCN-3 to assign a complex assertion to a variable and then reuse the variable to actually perform the matching with incoming data is central to our design and has an obvious advantage. In addition to this advantage, the use of a variable as test oracle has one additional advantage; the specified behavior becomes generic and can be used in various web applications without having to rewrite anything. The only code that needs to be rewritten is the login request which is always minimal because it consists in our case only in rewriting the request login template by providing the user id and password and the related form information. This process can be easily automated using TTCN-3. This is the result of TTCN-3's separation of concern between test behavior and conditions governing behavior that are implemented using the TTCN-3 concept of template. In this case, behavior remains constant while conditions change from application to application.

In our example, the test case to test would be divided into the definition of the templates followed by the definition of the test case itself.

First the specification of the legitimate and illegitimate login form templates, data types defined in [23], that themselves use other template definitions:

```
template BrowseFormType legitimateLoginForm_t := {
   name := "",
   formAction :=
            "https://peilos.servebeer.com/Account/Login?ReturnUrl=%2f",
   kindMethod := "post",
   elements := { aems_userId_t, aems_normal_pwd_t, aems_normal_button_t}
}
```

Second, the specification of the illegitimate login form merely re-uses the template definition of the legitimate login form by modifying only the form elements of which only the password element is different since now it consists of the illegitimate string (e.g. SQL injection):

```
template BrowseFormType illegitimateLoginForm_t modifies
                                          legitimateLoginForm_t := {
   elements := { aems_userId_t, aems_sql_injection_pwd_t, aems_normal_button_t}
}
```

Since a TTCN-3 template can re-use other templates at any point, the above specified elements can be specified as separate templates themselves to allow maximal structuring. This is the result of the multiple functionalities of the template that acts both as a variable that can be assigned values or other templates and an implicit function (execution of matching mechanism). For example, the following three templates specify the content of the above form input for a normal legitimate login

```
template FormElementType aems_userId_t := {
        elementType := "text",
        name := "username",
        elementValue := "admin"
}

template FormElementType aems_normal_pwd_t := {
        elementType := "password",
        name := "password",
        elementValue := "123456"
}

template FormElementType aems_normal_button_t := {
        elementType := "submit",
        name := "",
        elementValue := "Log In"
}
```

Now, for an illegitimate login, all we need is to specify a different template for the password element that contains the illegitimate string, in this case, a typical SQL injection value:

```
template FormElementType aems_sql_injection_pwd_t := {
        elementType := "password",
        name := "password",
        elementValue := "' or 1=1 - "
}
```

Finally, the specification of a minimal test case using the above defined templates:

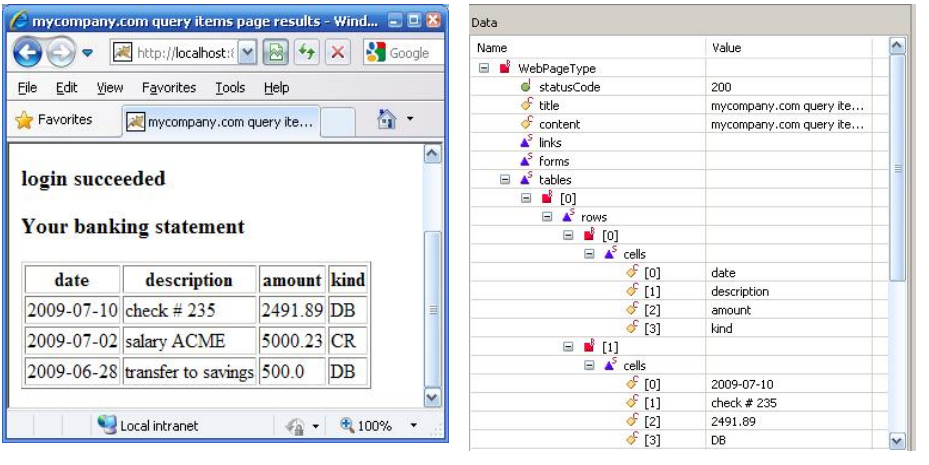
```
testcase AEMS_SQL_injection_attack_with_auto_verification()
                          runs on MTCType system SystemType {
   var template WebPageType legitimateResponse;

   … // open main page

   webPort.send(legitimateLoginForm_t);
   webPort.receive(WebResponseType:?)  -> value legitimateResponse

   … //return to main page

   webPort.send(illegitimateLoginForm_t);
   alt {
     [] webPort.receive(legitimateResponse) {
            setverdict(fail);
     }
     [] webPort.receive(WebResponse: ?) {
            setverdict(pass);
     }

   }

   …
}
```

In the above example, the request template *legitimateLoginForm_t* needs to be changed from application to application only. Each new web application has potentially different input field identifiers which need to be coded accordingly in the login template. The codec in the concrete layer will invoke the appropriate method providing the values of the parameters defined in the abstract template *legitimateLoginForm_t*. However, it is important to note that while the abstract layer will need to be modified from application to application, the concrete layer codec will not because it handles a generic representation of forms.

So far, this approach may appear to be complex when considering that the same principle could have been achieved with a simple string comparison between the two response contents using any GPL. The reality is not so simple. While in manual testing, a human tester can visually differentiate the content of two web pages on a browser because the rendering eliminates formatting information as shown on figure 2a, in automated testing when using string comparisons, test execution tools can only provide the HTML text in full as for example the following HTML code corresponding to figure 2a.

```
<HTML>
<HEAD>
<TITLE>mycompany.com query items page results</TITLE>
</HEAD>
<BODY >
<h3>login succeeded</h3>
<h3>Your banking statement</h3>
<table border="1" >
<tr> <th>date</th> <th> description </th> <th> amount </th> <th> kind </th></tr>
<tr><td>2009-07-10</td><td>check # 235</td><td>2491.89</td><td>DB</td></tr>
<tr><td>2009-07-02</td><td>salary ACME</td><td>5000.23</td><td>CR</td></tr>
<tr><td>2009-06-28</td><td>transfer to savings</td><td>500.0</td><td>DB</td></tr>
</table>
</BODY>
</HTML>
```

The above HTML code needs to be searched and finding the differences in potentially large HTML code is tedious mostly due to the presence of HTML tags. For example, we have observed that a sign in error web page for a real bank is composed of 258 lines of code and that the actual text indicating that the password entered is invalid is buried deeply at line 85. Instead, TTCN-3 decomposes web responses into structured content eliminating the HTML formatting information as shown on figure 2b in the process and then performs individual comparisons between these content's elements as a human would do, but with the additional benefit of automatically flagging the ones that did not match and thus making them easy to spot. This gives the tester an overview of test results and considerably increases the efficiency of debugging activities by reducing the number of test development iterations.



a) Legitimate web response page b) Abstract representation

Fig. 2. Abstract results inspection tools

4.2 Sharing Test Oracles among Actors

TTCN-3 is well known for its capability to compose complex test scenarios that include complex test configurations involving several test actors. In the case of XSS

vulnerabilities, we need to consider at least two actors, the penetration victim and the malicious attacker. Both of these users need to perform operations in a given order as shown on figure 3. Our example illustrates a classic persistent XSS attack that is achieved via a typical bulletin board application where users can post and read messages. A malicious user can post a message containing a script that in our case steals the bulletin board reader's cookies. The malicious user then uses the password stored in the cookie to illegally enter the victim's application. For this test to work, a precise choreography needs to be put in place that in TTCN-3 is implemented using coordination messages. Also, if we want to use the test oracles self-defining principle described in the previous section, we need to be able to make the content of the page reached through a normal login by the penetration victim available to the process that tests that the attacker can also reach that same page by exploiting the XSS vulnerability.

The TTCN-3 concept of parallel test component (PTC) is comparable to the object oriented concept of thread. It is an efficient way to separate the behavior of the two actors. One of the strong points of PTCs is that they also use TTCN-3 ports that can easily be connected at the abstract level without requiring any efforts about communication implementation. In our case, this language feature can be used for two purposes:

- Coordinate the actions of the two actors.
- Pass information from one actor to another.

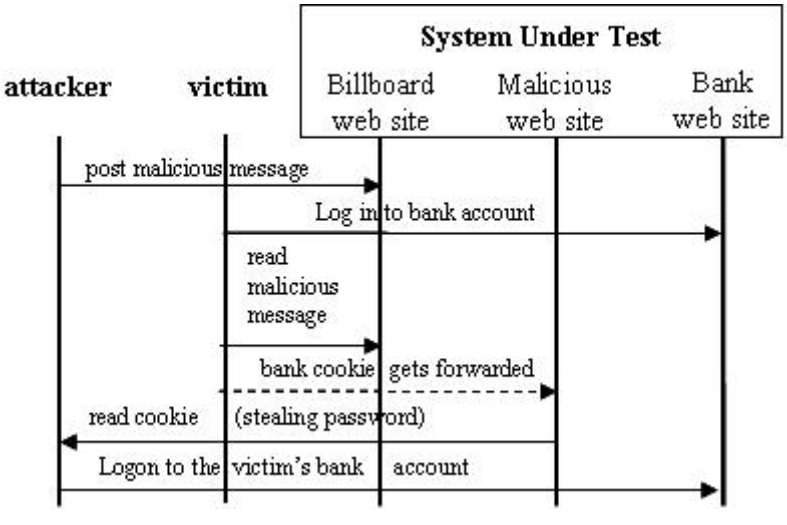


Fig. 3. persistent XSS attack sequence of events

The XSS vulnerability test case consists in three separate pieces of code. The first one describes the test case where two instances of PTCs are created to depict the attacker and victim. The coordination ports and the information passing ports are connected while the communication ports to the web application are mapped to concrete

connections. In our case the concrete connections consist of instances of *WebClient* classes that are part of the concrete layer. Since, the PTCs are independent threads, a number of coordination messages need to be exchanged between the master test component (MTC) and the PTCs as shown in the following test case:

```
testcase XSS_PersistentAttack() runs on MTCType system SystemType {
   var PTCType attacker := PTCType.create("attacker");
   var PTCType victim := PTCType.create("victim");

   connect(mtc.attackerCoordPort, attacker.coordPort);
   connect(mtc.victimCoordPort, victim.coordPort);
   connect(victim.infoPassingPort, attacker.infoPassingPort);

   map(attacker:webPort, system:system_webPort_attacker);
   map(victim:webPort, system:system_webPort_victim);

   attacker.start(attackerBehavior());
   victim.start(victimBehavior());

   attackerCoordPort.send("post message on bulletin board");
   attackerCoordPort.receive("post message done");
   victimCoordPort.send("login ");
   victimCoordPort.receive("login performed");
   victimCoordPort.send("read bulltein board");
   victimCoordPort.receive("read bulletin board done");
   attackerCoordPort.send("login");
   attackerCoordPort.receive("login performed");

   all component.done;
}
```

The discovered test oracle on the victim's PTC must now be passed to the Attacker PTC since they are running on different threads. This is achieved merely by the victim's PTC sending a message to the attacker's PTC using the tester defined *infoPassing* port in the respective behavior functions of the victim and the attacker that are executed in parallel as shown on figure 4.

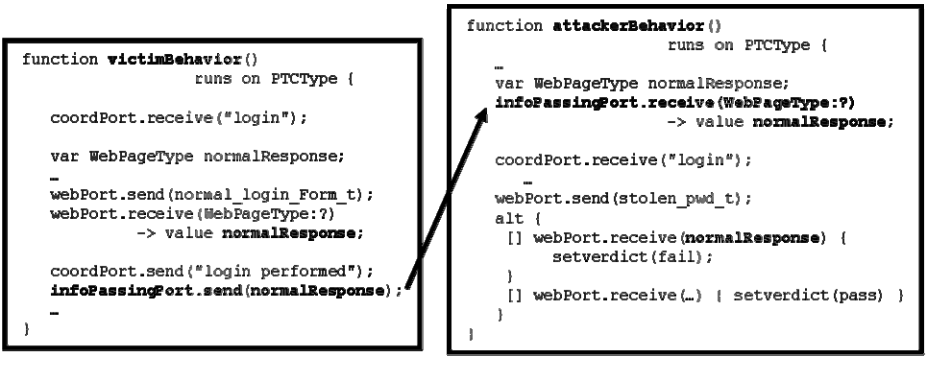


Fig. 4. Inter process transmission of test oracles

In the separate processes shown on figure 4, we use the blocking receive statement to control the execution of the attacker behavior rather than using a coordination message.

4.3 Re-usability of the Test Suite

Our test system is composed of a large portion of re-usable code at various levels. Besides the re-usability of the test oracle self-defining principle, the infrastructure on the SUT side is also re-usable for various different applications. For example, bulletin board message posting and cookie stealing mechanisms remains unchanged from application to application. Thus, the only portion that needs to be modified from application to application is confined to the login templates at the abstract layer and the parsing of cookies to actually steal passwords at the malicious attacker's cookie gathering application. It seems likely that penetration test patterns could be defined for these with TTCN-3 templates used as examples.

5 A Proposed Refinement to the TTCN-3 Model Architecture

So far we have discussed the advantages of the TTCN-3's separate abstract and concrete layers without providing too much detail on the concrete layer side. The concrete layer has two basic functionalities:

- A codec that translates the internal abstract representation of data to and from bytes.
- A test adapter that sends and receives data to and from the SUT as bytes.

The TTCN-3 standard part V [6] and part VI [7] clearly defines two interfaces for these two functionalities, namely the *TciCDProvided* interface and the *TriCommunicationSA* and TriPlatformPA interfaces respectively. Both need to be implemented by the test application developer to create a concrete codec and a concrete test adapter. Because TTCN was originally conceived for telecommunications applications that consist mostly in sending and receiving compacted information as bytes, the architecture of TTCN-3 execution tools consists of a strict sequence of invoking coding or decoding methods of the implemented codec class to obtain bytes and then invoking the *triSend* method of the implemented test adapter class in order to concretely transmit the bytes to the SUT over a communication media and vice versa as shown on figure 5a. This architecture is not usable when using object oriented frameworks such as *HtmlUnit* mostly because with these frameworks there is no such a clear distinction between data coding/decoding and data transmission as in the TTCN-3 model but also because of its original byte stream orientation it has not been designed for keeping states of objects as required by web testing frameworks. In short, the two TTCN-3 concrete layer classes can not be mapped directly to the single *HtmlUnit WebClient* class. For example, with *HtmlUnit* we have only one class, *WebClient*, instead of two to accomplish both codec and test adapter functionalities. The obvious solution to this mismatch of architectures would consist in bypassing the TTCN-3 codec class altogether and having *WebClient* object instances residing in the TTCN-3 test adapter. This solution requires a modification of the TTCN-3 execution tools

architecture. The TTCN-3 codec can not be eliminated because of the requirements of telecommunication applications but it can be adapted so as to act merely as a relay that passes abstract value objects down to the test adapter where frameworks can be used to both perform codec and data transmission functionalities as shown on figure 5b. Thus, we have created an *HtmlUnit WebClient* driven codec and test adapter. For this to work, the abstract values from the abstract layer must reach the test adapter without being converted to bytes by the traditional codec. This is actually easy to implement because even with the traditional architecture, the messages coming from the codec would not arrive directly as bytes but as objects instances of the class implementation of the *TriMessage* interface also defined in [7] that contain an attribute for the bytes. Thus, the solution would be to extend the *TriMessage* implementation class with an attribute containing the abstract values. The problem with this solution is that the standard specifies that the implementation of the *TriMessage* interface is the responsibility of the tool provider and usually this is proprietary. Thus, TTCN-3 execution tools are written so as to process *TriMessage* implementation objects but not any user defined extensions. Thus, this solution requires the collaboration of tool vendors which we obtained and it is to be noted that this solution does not require any changes to the TTCN-3 standard since *TriMessage* is an interface only. However, we do recommend that this feature be implemented in the TTCN-3 standard so as to promote this efficient solution to external frameworks integration.

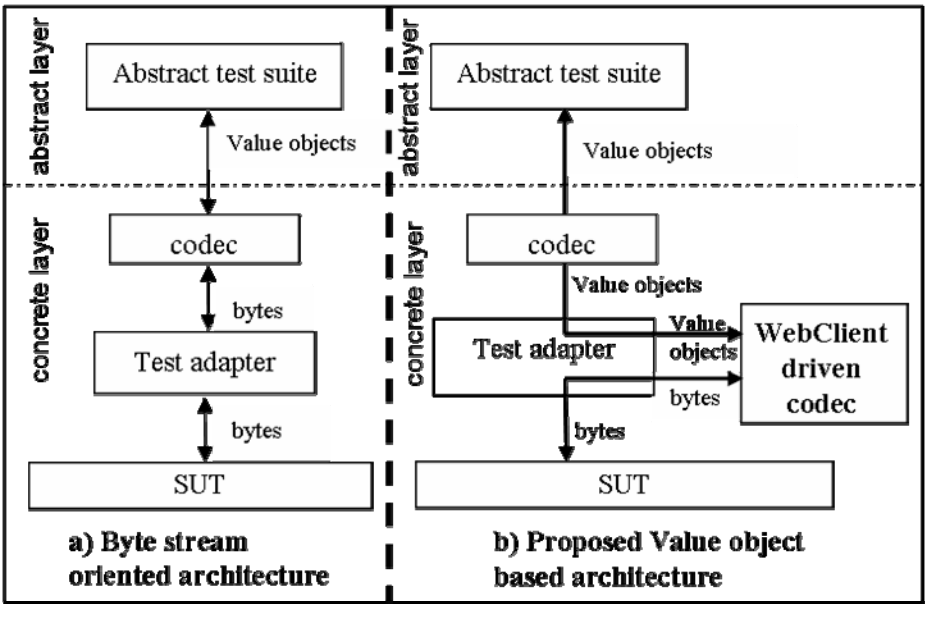


Fig. 5. TTCN-3 architectural models

This new architecture is a considerable improvement over previous work around such as making *WebClient* objects available to the codec class instance either through the user extended codec class constructor or via serialization which is not always

possible because objects need to implement the *Seriazible* class which is not even the case for the *WebClient* class. For example, a form submission would consist in invoking the *click()* method of the *Button* object of the login form. This clicking a button functionality makes no sense to be modeled using a byte stream. This requires populating the *Form* object instance with the parameter values that arrive as a TTCN-3 abstract *Value* object from the abstract layer. This *click()* method actually sends the request using the HTTP connection of the *WebClient* object. This architecture is even more important when there are client side javascript functions to be executed. In this case, even the concept of the SUT is no longer as clear as the TTCN-3 model defines it. Our approach proves to be particularly efficient in case of multiple user configurations using PTCs.

6 Conclusions

In this paper, we have shown how TTCN-3 can be effectively combined with frameworks like HttpUnit and HtmlUnit. Up until now, there has been little awareness in the literature that TTCN-3, as a telecom standard, was appropriate for application penetration testing, and certainly no discussion on how best to use TTCN-3. Of course, using a test specification language like TTCN-3 requires more sophistication and training of testers in order to use the tool effectively, but given the high cost of software development and the critical nature of web application security, we believe such an investment is more than worthwhile.

TTCN-3 provides a more systematic and effective approach to penetration testing by

- Separating test logic from data extraction logic.
- Leveraging test oracles and templates for more powerful assertion writing, support of parallel execution tests, and reuse.

These were illustrated with examples based on our experiences in actual projects in industry. However, more systematic case studies need to be done to validate and quantify the benefits that can be achieved by our approach.

It would also be beneficial to investigate the systematic create of test oracle templates, and penetration test patterns to address common situations that occur in web application penetration testing.

Finally, we have identified a key limitation in the current TTCN-3 model architecture as defined by the standard and proposed a simple refinement of the architecture to address it. A TTCN-3 tool vendor has already incorporated the change in their most recent tool offering.

Acknowledgements

The authors would like to thank Testing Technologies IST GmbH for providing us the necessary tool -- TTworkbench to carry out this research as well as NSERC for partially funding this work.

References

1. Andreu, A.: Professional Pen Testing for Web Applications. Wrox Press (2006)
2. Arkin, B., Stender, S., McGraw, G.: Software Penetration Testing. IEEE Security & Privacy 3(1), 84–87 (2005)
3. AttackAPI (2010), http://www.gnucitizen.org/blog/attackapi/ (retrieved November 2010)
4. Bishop, M.: About Penetration Testing. IEEE Security & Privacy 5(6), 84–87 (2007)
5. ETSI ES 201 873-1 (2008). The Testing and Test Control Notation version 3, Part 1: TTCN-3 Core notation, V3.4.1 (September 2008)
6. ETSI ES 201 873-5 (2008). The Testing and Test Control Notation version 3, Part 5: TTCN-3 Runtime Interface, V3.4.1 (September 2008)
7. ETSI ES 201 873-6 (2008). The Testing and Test Control Notation version 3, Part 6: TTCN-3 Control Interface (TCI), V3.4.1 (September 2008)
8. FireBug (2010), http://getfirebug.com/ (retrieved November 2010)
9. GreaseMonkey (2010), https://addons.mozilla.org/en-US/firefox/addon/748/ (retrieved November 2010)
10. HtmlUnit (2010), http://HtmlUnit.sourceforge.net/ (retrieved November 2010)
11. HttpUnit (2010), http://HttpUnit.sourceforge.net/ (retrieved November 2010)
12. JUnit (2010), http://www.junit.org/ (retrieved November 2010)
13. Brzezinski, K.M.: Intrusion Detection as Passive Testing: Linguistic Support with TTCN-3 (Extended Abstract). In: Hämmerli, B.M., Sommer, R. (eds.) DIMVA 2007. LNCS, vol. 4579, pp. 79–88. Springer, Heidelberg (2007)
14. Manzuik, S., Gold, A., Gatford, C.: Network Security Assessment: From Vulnerability to Patch. Syngress Publishing (2007)
15. Metasploit project (2010), http://www.metasploit.com/ (retrieved November 2010)
16. OWASP Testing Guide, OWASP Testing Guide (2008), https://www.owasp.org/images/8/89/OWASP_Testing_Guide_V3.pdf (retrieved November 2010)
17. OWASP TOP 10, OWASP TOP 10: The Ten Most Critical Web Application Security Vulnerabilities (2007), http://www.owasp.org/images/e/e8/OWASP_Top_10_2007.pdf (retrieved November 2010)
18. Palmer, S.: Web Application Vulnerabilities: Detect, Exploit, Prevent. Syngress Publishing (2007)
19. Potter, B., McGraw, G.: Software Security Testing. IEEE Computer Society Press 2(5), 81–85 (2004)
20. Prabhakar, T.V., Krishna, G., Garge, S.: Telecom equipment assurance testing, a T3UC India presentation (2010), http://www.ttcn3.org/TTCN3UC_INDIA2009/Presentation/1-ttcn3-user-conference-nov_updated_-2009.pdf (retrieved November 2010)
21. SANS TOP 20, 2010 TOP 20 Internet Security Problems, Threats and Risks, from The SANS (SysAdmin, Audit, Network, Security) Institute (2010), http://www.sans.org/top20/ (retrieved November 2010)

22. Splaine, S.: Testing Web Security: Assessing the Security of Web Sites and Applications. John Wiley & Sons, Chichester (2002)
23. Stepien, B., Peyton, L., Xiong, P.: Framework Testing of Web Applications using TTCN-3. International Journal on Software Tools for Technology Transfer 10(4), 371–381 (2008)
24. Thompson, H.: Application Penetration Testing. IEEE Computer Society Press 3(1), 66–69 (2005)
25. Xiong, P., Stepien, B., Peyton, L.: Model-based Penetration Test Framework for Web Applications Using TTCN-3. In: Proceedings Mce.Tech. 2009. Springer, Heidelberg (2009)

Towards Model-Based Support for Managing Organizational Transformation

Daniele Barone[1], Liam Peyton[2], Flavio Rizzolo[2], Daniel Amyot[2], and John Mylopoulos[3]

[1] Department of Computer Science, University of Toronto, Toronto (ON), Canada
barone@cs.toronto.edu
[2] SITE, University of Ottawa, Ottawa (ON), Canada
{lpeyton,frizzolo,damyot}@site.uottawa.ca
[3] DISI, University of Trento, Trento, Italy
jm@disi.unitn.it

Abstract. In an increasingly connected and dynamic world, most organizations are continuously evolving their business objectives, processes and operations through ongoing transformation and renewal, while their external environment is changing simultaneously. In such a setting, it is imperative for organizations to continuously monitor their performance and adjust when there is a need. The technology that delivers this monitoring capability is called Business Intelligence (BI), and over the years it has come to play a central role in business operations and governance. Unfortunately, there is a huge cognitive gap between the strategic business level view of goals, processes, and performance on one hand, and the technological/implementation view of databases, networks, and computational processing offered by BI tools on the other.

In this paper, we present a model-based framework for bridging this cognitive gap and demonstrate its usefulness through a case study involving organizational transformation. The business view is modeled in terms of the Business Intelligence Model (BIM), while the data collection and reporting infrastructure is expressed in terms of the Conceptual Integration Model (CIM). The case study involves a hospital implementing a strategic initiative to reduce antibiotic resistant infections.

Keywords: business intelligence, model-based, data integration, organizational transformation.

1 Introduction

In an increasingly connected and dynamic world, most organizations are continuously evolving their business objectives, processes and operations through an ongoing process of transformation and renewal. A variety of business methodologies or frameworks exist that are intended to guide an organization to improve its business processes in an incremental way [1]. Typically, strategic initiatives identify opportunities and enact change through a continuous process of monitoring and measurement to align operational performance with strategic targets. A

G. Babin, K. Stanoevska-Slabeva, P. Kropf (Eds.): MCETECH 2011, LNBIP 78, pp. 17–31, 2011.

tool-supported methodology that can integrate goals, processes and performance is essential to help management implement such initiatives by automating or semi-automating some of the implementation tasks [2]. In current practice, key performance indicators play a bridging role by integrating data from a variety of sources inside and outside an organization to measure how well strategic business targets are being met [3].

Unfortunately, there is a huge cognitive gap between the strategic business level view of goals, processes, and performance and the technological view of databases, networks, and computational processing needed to deliver an implementation of these concepts. The implementation is intended to offer a monitoring function for key performance indicators that determine how the organization is doing with respect to its strategic initiatives. A modeling approach is needed that can represent both the business and technological view of things, along with computational mappings that bridge the gap between the two levels and deliver ongoing monitoring and transformation.

In this paper, we present a model-based framework for organizational transformation that builds on concepts and technologies from computer science and management. The key elements of the framework are:

- A Business Intelligence Model (BIM) that represents strategic initiatives and their associated plans in terms of goals, processes, and indicators.
- A Conceptual Integration Model (CIM) that represents a conceptual view of organizational data integrated to create focused dashboards for reporting indicators used to monitor strategic initiatives.
- A mapping framework between BIM and CIM, along with corporate dashboards that link the two levels for purposes of monitoring and reporting.

A case study is used to demonstrate the workings of our proposed framework. It involves a hospital that decides to implement a strategic initiative to reduce antibiotic resistant infections.

Section 2 of this paper includes a brief introduction into BIM and CIM concepts, as well as an overview on managing organizational transformation and BI. Section 3 introduces the different phases to manage the lifecycle of an initiative. Section 4 presents a case study drawn from a strategic initiative currently underway at a large teaching hospital. Sections 5 and 6 illustrate, respectively, how BIM supports modeling activity and how CIM supports the mapping to interconnect BIM to data. Section 7 illustrates the related work, followed by Section 8 which provides an evaluation of our approach. Section 9 presents our conclusions and plans for future research.

2 Baseline

We give a brief overview of managing organizational transformation and Business Intelligence, as well as the foundations of our Business Intelligence Model (BIM) and Conceptual Integration Model (CIM).

Organizational Transformation and Business Intelligence. Organizational transformation [4,5] is a process through which low-performance organizations

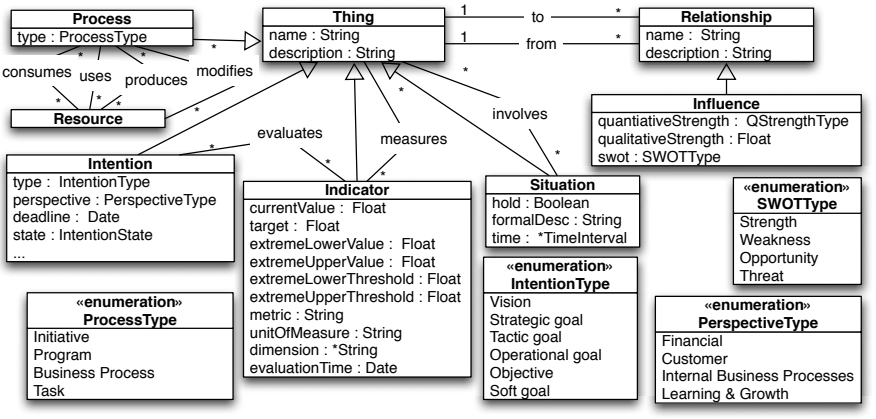


Fig. 1. A fragment of BIM

change state and become strategically healthy. As described in [6], "strategically healthy organizations respond efficiently to change, anticipate change in a beneficial way, and lead change within their industries". Business Intelligence [7] can be a powerful enabler for such a strategic *transformation* in order to produce a high-performance organization. In particular, BI systems combine operational data with analytical tools to present complex and competitive information to planners and decision makers. In fact, as described in [8], BI is a process that includes two primary activities: *getting data in* and *getting data out*. The former involves moving data from a set of sources into an integrated data warehouse; the latterconsists of business users (and applications) accessing data from the data warehouse to perform enterprise reporting, OLAP querying, and predictive analysis. BIM supports *getting data in* activities by defining clear requirements that make explicit the information needed to evaluate strategies; and *getting data out* activities by presenting to the user an abstract view of the their business (in terms of goals, processes, resources, and other concepts) and its performance. On the other hand, CIM collects and integrates organization's data sources (therefore, it supports *getting data in* activities) and makes them available to BIM, and in turn, to the business users.

The Business Intelligence Model. The Business Intelligence Model [9] allows business users to conceptualize their business operations and strategies using concepts that are familiar to them, including Actor, Directive, Intention, Event, Situation, Indicator, Influence, and Process. Figure 1 shows the fragment of BIM used in this paper (see [9] for details). BIM is drawn upon well-established concepts and practices in the business community, such as the Balanced Scorecard and Strategy Maps [10,11], as well as techniques from conceptual modeling and enterprise modeling, such as metamodeling and goal modeling techniques.

In particular, BIM can be used by business users to build a business schema of their strategies and operations and performance measures. Users can therefore query this business schema using familiar business concepts, to perform analysis

on enterprise data, to track decisions and their impacts, or to explore alternate strategies for addressing problems. The business queries are translated through schema mappings into queries defined over databases and data warehouses, and the answers are translated back into business-level concepts. BIM works together with CIM to address such an issue and, in this paper, we show how such a connection is performed (in particular) for indicators.

The Conceptual Integration Model. A data warehouse is a repository of data that has been materialized for statistical and analytical purposes. Data warehouses are organized in multidimensional fashion, i.e., the basic data stored in *fact tables* are linked to various views or *dimensions* that help analyze the data in multiple ways. As in the relational model [12], there is an impedance mismatch between (business intelligence) applications accessing the data and data's physical storage. The problem is exacerbated by the fact that the underlying multidimensional data is physically organized for data access performance rather than to reflect the conceptual and business models that the data and business analysts have in mind.

To raise the level of abstraction and bridge the ever increasing gap existing between physical data warehouse schemas and conceptual multidimensional models, the Conceptual Integration Modeling (CIM) Framework was proposed [13]. The CIM Framework offers both design time and run time environments based on a CIM Visual Model (CVM), which provides two different views of a data warehouse: a conceptual model of the data (called CVL — Conceptual Visual Language) and a physical model of the data (called SVL — Store Visual Language). In other words, the CVL provides an abstract, high-level view of the data stored in the physical tables of the SVL. The representational gap between the CVL (conceptual) and the SVL (physical) models is filled by the MVL (Mapping Visual Language) consisting of correspondences (with optional value conditions) between attributes of entities in the CVL and the SVL models. The CIM tool can then compile these simple correspondences into complex views over the physical model that can be used to efficiently evaluate queries posed on the conceptual model.

3 Managing the Strategic Initiative Lifecycle

Enacting organizational transformation through the implementation of strategic initiatives is a well understood process [4,10,11] that is taught in business schools. Changes to organizational intentions, processes, and resources are implemented and monitored in order to address a particular problem or opportunity. Figure 2 shows how the key elements of our framework integrate with and support the iterative lifecycle of a strategic initiative. The lifecycle is iterative because the organizational changes enacted by the initiative are refined and updated based on the feedback provided by indicators. The mapping framework from BIM to CIM and from CIM to Dashboard facilitates implementation and maintenance of the initiative throughout its lifecycle.

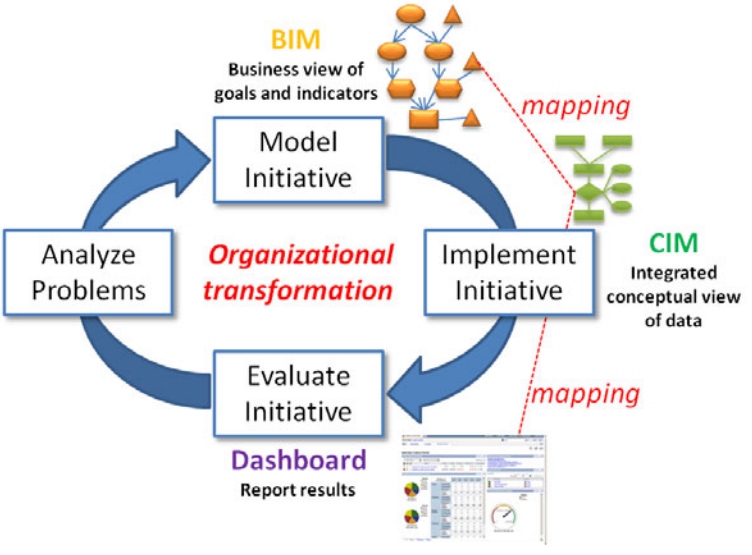


Fig. 2. Model-based management of strategic initiatives

Model Initiative: an initiative is modeled by the business analyst in all its aspects through the use of BIM. Strategic goals are defined and decomposed hierarchically until operational goals are reached. Business processes are described along with resources they use, consume and produce in order to achieve the hierarchy of goals. To evaluate the performance of the initiative, performance measures, i.e., key performance indicators, are created and associated to strategic goals, business processes, resources, actors, etc.

Implement Initiative: the initiative is implemented within the organization where business processes are executed and performed by employee, policies are enforced, resources are consumed and produced, and so on. In this phase, data are collected and integrated from a variety of applications, systems and documents into a data mart or a data warehouse. From such integrated view, CIM is used to obtain a corresponding conceptual representation which, in turn, is connected to BIM.

Evaluate Initiative: Performance measures are calculated from the collected data and are evaluated against the defined targets. Dashboards [14] are used to report such evaluations to the business users allowing for insight to reveal whether or not an actual value for a business's aspect deviates too far from a pre-defined target. Past trends and predictions can be also visualized.

Analyze Problems: Further analysis is performed on critical area identified in the previous phase to understand why an organization may or may not be on track to meet a specific target or objective. In this phase, operational information collected via monitoring is used to identify the causes of faults and errors as they occur, as well as to forecast performance levels and threats to

operational stability. Discoveries made during analysis should help the management in planning next steps, set new (or adjust existing) expectations, and predict what may happen based on organization's decisions.

The organizational transformation cycle allows for a continuous improvement process in which feedback from the measurement system provides managers with the necessary information to make change or adjust business activities. The details of the framework are explained and demonstrated through the use of a case study in the next sections.

4 Case Study: Reducing Antibiotic Resistant Infections

We will explain and illustrate our model-based framework for managing organizational change using examples drawn from a strategic initiative currently underway at a large teaching hospital to Reduce Antibiotic Resistant Infections (RARI) by changing the way antibiotics are used. Increasingly, hospitals have been plagued with outbreaks of micro-organisms that are resistant to antibiotics, including Clostridium difficile (C.difficile), methicillin-resistant staphylococcus aureus (MRSA), and vancomycin resistant enterococcus (VRE). One reason for these outbreaks is the overuse of antibiotics, which selectively allows these organisms to thrive in an environment [15]. Antibiotics are also very expensive. They account for about 30% of a typical hospital's pharmacy budget [16]. Thus, overuse of antibiotics leads to increased morbidity in patients and excess cost.

The ultimate goal of the RARI initiative is to reduce the number of incidents of antibiotic resistant infections, but the focus of the initiative is to limit the amount and number of prescriptions for antibiotics deemed to be high risk. An education campaign for physicians will be created to change the type, amount and number (or percentage) of antibiotic prescriptions. Correct medication guidelines will be defined for antibiotic usage and it will be monitored with monthly and annual reporting of prescription rates (percentage, number, total amount, and duration) by service, location, physician, and antibiotic type. It is expected that there will be cost savings to the hospital both from fewer antibiotics used, and through a lower rate of incidents.

Enacting such a strategic initiative is a complex task both from a business point of view and a technology point of view. In particular, it is important to precisely define the indicators that will be used to monitor whether the goals of the initiative are being met and map this definition accurately and efficiently to the collection and reporting of the data used to measure the indicators. The data needed for the indicators must be integrated from many different data sources including the pharmacy records, administrative records that indicate where patients were located (bed, unit, campus) when the prescription was made, and for what service the prescribing physician was working. As well, individual departments within the hospital each have their own clinical information systems to classify which antibiotics in what amounts are appropriate for what diagnoses.

An infection control dashboard was created for infection control analysts to monitor indicators and evaluate the effectiveness of the strategic initiative. It presented a dimensional view of indicators relevant to the initiative. The dimensional view allowed prescription usage to be broken down by time (hour, day, month, year), drug (drug category, drug type, drug brand), location (bed, nursing unit, campus) and by organization (physician, service, department). The indicators tracked were the **number of antibiotic prescriptions**, the **percentage of antibiotic prescriptions** (over all prescriptions), the **average duration of antibiotic prescription** (measured in hours), and the **average and total amount of antibiotic prescription(s)** (measured in milligrams for the entire duration of the prescription).

5 Modeling Strategic Initiatives with BIM

The set of primitives provided by BIM allows a business user to define a business schema representing the RARI initiative undertaken by the hospital. The initiative is modeled in terms of strategic goals, processes and resources, and is monitored by indicators to understand whether or not goals are met or to identify possible sources of problems. A complete description of how such schemas are built can be found in [17]; while, in the following sections, (part of) the business schema to define and monitor the RARI initiative is shown.

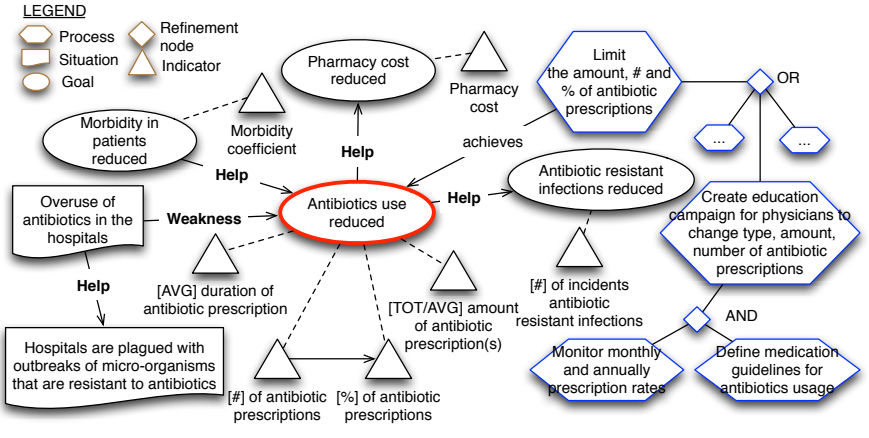


Fig. 3. Strategic goals

Strategic Goals Definition. Figure 3 illustrates the high level strategic goals for the hospital and the RARI initiative. The BIM Intention primitive is used to represent the hospital's strategic goals[1], while the Situation primitive is used

[1] The term Strategic goal is one of the values which can be assumed by the type attribute associated to the Intention primitive (see Figure 1).

to represent those partial states of the world which can positively or negatively influence such goals. For example, the Situation "Overuse of antibiotics in the hospital" undermines or weakens the hospital's goal to reduce the use of antibiotics. In the figure, the meaning of *weakness* and *threat* labels derives from SWOT analysis [18], in which the former represents an internal factor to the hospital that is harmful to achieving the goals while the latter is an external factor or condition which could do damage to the goals. To reason among goals and among situations, we use a qualitative (scale) contribution to characterize influence relationships among goals and among situations as is supported in GRL models [19]. For example, the "help" label in Figure 3 should be read as: *a reduced use of antibiotics "helps" to reduce pharmacy cost (see Section 4) while a situation in which antibiotics are overused can favor (help) outbreaks of micro-organisms that are resistant to antibiotics.*

Figure 3 also shows the RARI initiative and its decomposition. Due to space limitation, only one alternative is shown for its refinement, i.e., the education campaign creation, but more actions can be planned to reduce antibiotic resistant infections. The figure also shows a set of Indicators that are defined to monitor the impact of the initiative on the strategic goals. For simplicity, in the rest of the paper we will focus the analysis on the "Antibiotics use reduced" goal, but a similar analysis can be done for all the strategic goals in the figure, e.g, the "Antibiotic resistant infections reduced".

The Drug Treatment Process. Figure 4 describes the drug treatment process where antibiotics are prescribed. We can see that it is decomposed into the "Medication prescription" and "Medication administration" activities. In BIM, resources can be classified according to their nature; for example, we have

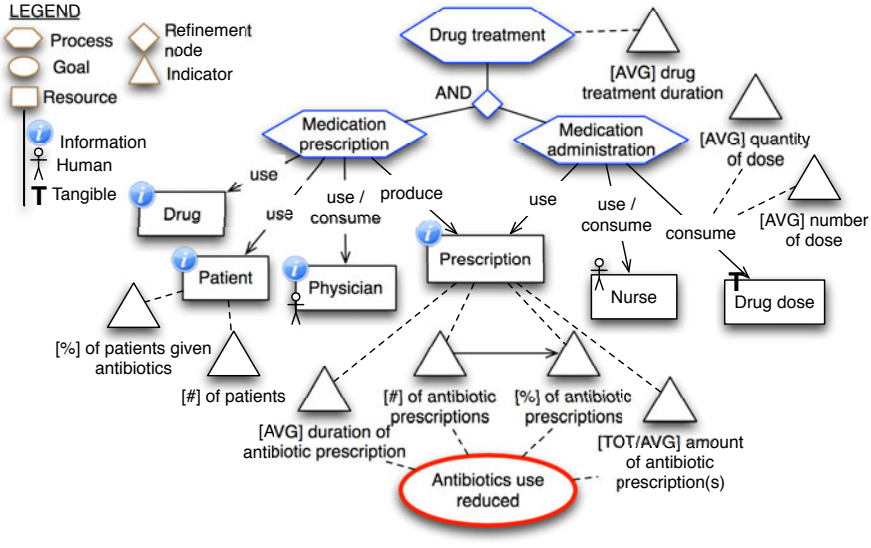


Fig. 4. The *Drug treatment* process

information resources, human resources, capability/skill resources, etc. Moreover, BIM provides four relationships among processes (or activities) and resources, namely *uses(p,r)*, *consumes(p,r)*, *modifies(p,r)*, and *produces(p,r)*. An in-depth description of resource classification and their relationships with processes can be found in [9,17]. For instance, in Figure 4, a prescription is produced by the "Medication process" by i) using information on patients and on the drugs available in the hospital, ii) using and consuming, respectively, skills and time of a doctor (this is the meaning of the the use/consume relationship associated to a human resource in the figure).

Notice how BIM allows to define indicators on processes and resources to monitor their performance with respect to intentions. Indeed, BIM helps to motivate why an indicator is needed (e.g, to evaluate antibiotics use reduction) and which aspect of business such an indicator must measure (e.g., the amount of antibiotic in a prescription). For example, in Figure 4, with respect to the RARI initiative, we need to concentrate on those indicators associated with the prescription resource since they monitor the doctor behavior[2] the initiative aims to modify.

In the following section, we show how BIM can be supported by CIM to feed indicators with data.

6 Data Mappings in CIM and BIM

Figure 5 shows a CIM model consisting of a CVL (on the left) and an SVL (on the right). Medication and Prescriber (shadowed rectangles) are CVL dimensions describing measures in the Prescription fact relationship (shadowed diamond). Non-shadowed rectangles (e.g., Drug, Physician) represent CVL levels in the dimensions. These levels are organized into hierarchies by parent-child relationships, which are drawn as edges between levels. For instance, the Prescriber hierarchy indicates that all physicians roll up to Unit, Campus and Service. The SVL is a UML-like representation of the relational data warehouse schema, containing relational table definitions, keys and referential integrity constraints. The left to right dashed arrows are part of the MVL and represent the correspondences between the models. For instance, the CVL Prescription fact relationship is physically stored in two different data warehouse tables: the SVL Prescription and Patient.

The CVL specification corresponds to what is increasingly called the semantic layer in industry. Such a layer liberates users from the low-level multidimensional intricacies and allows them to focus on a higher level of abstraction. For instance, the SVL model in the figure has normalized tables, which is not necessarily the best way to represent multidimensional entities in the conceptual view.

It is important to note that the only model that contains materialized data is the SVL; the CVL can access SVL data only through mappings. Interestingly, the user-defined correspondences that appear in the MVL are not sufficient for exchanging data from the SVL to the CVL – some data dependencies are lost by

[2] In fact, the term prescription is commonly used to mean an order (from a doctor to a patient) to take certain medications, while we use the term drug dose to identify the actual medication's dose a nurse administrates to a patient.

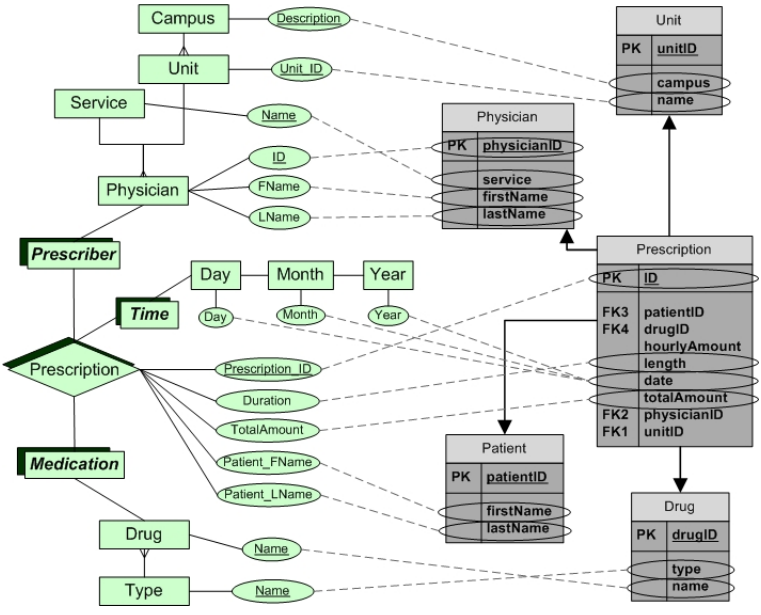


Fig. 5. CIM Visual Model for the medication prescription activity: CVL (left), SVL (right) and MVL (left-right dashed lines)

such simple attribute-to-attribute mappings. For data exchange, the CVL and SVL models are related by more complex mappings [20].

However, it is not practical for a high-level data analyst accustomed to deal only with the conceptual view of the data to come up with such complex view definitions in terms of the tables of the physical data warehouse. That is the reason why CIM requires from the user only very simple correspondences between attributes. Then, the CIM tool takes care of transparently compiling the user-defined correspondences into complex, fully-fledged, multidimensional mappings that can be used for query evaluation [21]. This is similar to the approach followed by EDM [12] for the relational setting.

A similar situation arises when trying to map a business model to an existing data warehouse with SVL, CVL and MVL already defined. In that situation, BIM entities can be related to a query expression (view) over the CVL, much in the same way CVL entities are mapped to views over the SVL. Such expressions can be complex multidimensional queries with aggregations and roll up functions. Again, writing these complex expressions is not practical for a business user. Instead, the business user draws simple correspondences between the models at hand, this time between BIM and CVL, and the BIM tool generates the multidimensional CVL views representing the user's data requirements expressed in the correspondences.

Consider Figure 6. The CVL model on the left-hand side is the one from Figure 5. The BIM model on the right-hand side corresponds to the Medication Prescription activity of Figure 4. BIM entities have attributes that are not represented

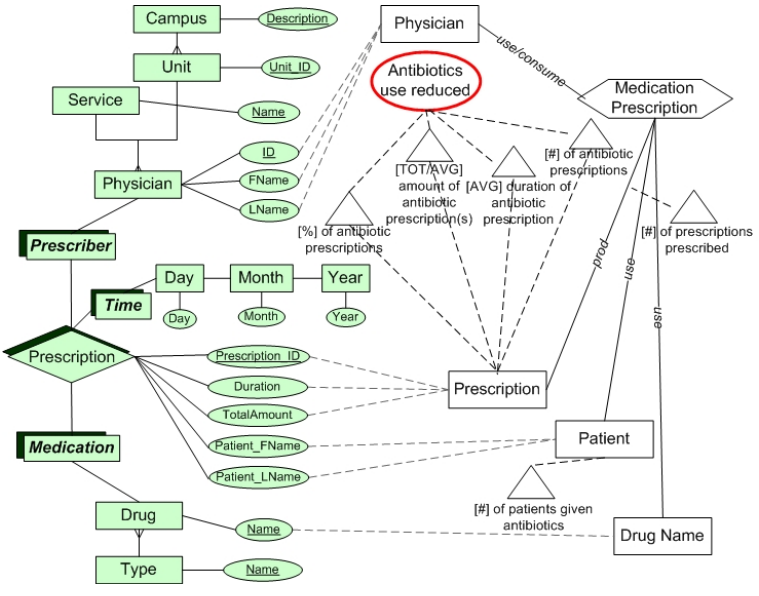


Fig. 6. BIM+CIM Visual Model for the medication prescription activity: CVL (left), BIM (right) and mappings (left-right dashed lines)

in the figure for simplicity – they happen to have the same names as the CVL attributes they are mapped to. For instance, there are four correspondences from the BIM Prescription resource to four attributes in the CVL Prescription fact relationship, i.e., Prescription_ID, Duration, Date and TotalAmount, which are also the names of the Prescription resource attributes. Moreover, for Indicators we have (hidden) information such as target, threshold, current value, etc., but also, more important for the mapping task, dimensions and levels to represent hierarchy[3].

The BIM mapping compilation takes these correspondences and creates views over the CVL. For instance, BIM Prescription is mapped to a CVL view defined as V1=Prescription(Prescription_ID, Duration, Date, TotalAmount), basically a SELECT query in SQL. Every time the Prescription resource needs to pull data from CIM, view V1 is used.

Some other views involve roll up queries with aggregations, i.e., the views to feed BIM indicators. For instance, for the following BIM indicators we have:

– [#] **of antibiotic prescriptions:** The actual value for the indicator is obtained by a query that aggregates the number of instances that appear in the CVL Prescription fact table which have a value equals to "antibiotic" for the Type.Name attribute in the Medication dimension.

[3] The possible dimensions (and levels) available for an indicator are elicited by the CVL fact table with which it is associated, e.g., the dimensions for [#] of antibiotic prescriptions are "Prescriber" and "Medication".

- [%] **of antibiotic prescriptions:** The actual value for the indicator is obtained by the value of [#] of antibiotic prescriptions divided by a query that aggregates the number of instances that appear in the CVL Prescription fact table, all multiplied by 100.
- **[TOT/AVG] amount of antibiotic prescription(s):** The actual value for the indicator is obtained by a query that aggregates the amounts that appear in the CVL attribute Prescription.TotalAmount for all those corresponding to Type.Name="antibiotic" on the Medication dimension. If the average is requested, the above value is divided by the [#] of antibiotic prescriptions.
- **[AVG] duration of antibiotic prescription:** The actual value for the indicator is obtained by a query that aggregates the durations that appear in the CVL attribute Prescription. Duration for all those corresponding to Type.Name="antibiotic" on the Medication dimension, and divides it by the [#] of antibiotic prescriptions.

As explained above, the current values for these indicators can be drilled-down using dimensions and levels defined in the BIM Indicator. Indeed, when a drill-down action is performed, a corresponding query is performed on the CVL. For example, a BIM user can desire to have [#] of antibiotic prescriptions prescribed by a Physician named "John Smith". In such a case the above query is reformulated considering the Prescriber dimension with Physician.FName="John" and Physician.LName="Smith".

7 Related Work

In the literature, different approaches from goal-oriented requirements engineering, e.g., [22,23], combine intentional and social concepts to model organization strategies and their elements (e.g., actors, resources, and processes). Other works have also extended *i** [22] and related frameworks (e.g., URN [23]) towards enterprise and business modeling, e.g., [24]. A recent extension of URN includes indicators [2], but does not address the question of how to link the business level view to technology. The BIM aims to unify various modeling concepts into a coherent framework with reasoning support and connection to enterprise data, built upon a firm conceptual modeling foundation. In particular, with respect to the above works, BIM includes (among others): the notion of *influence* which is adopted from influence diagrams [25], a well-known and accepted decision analysis technique; SWOT analysis concepts [18] (strengths, weaknesses, opportunities, and threats) and others which are adopted from OMG's Business Motivation Model standard [26]; and support for Balanced Scorecard and Strategy Maps [10,11].

Moreover, BIM's concepts are formalized through metamodeling in terms of abstract concepts such as Thing, Object, Proposition, Entity, and Relationship, taking inspiration from the DOLCE [27] ontology.

A number of conceptual multidimensional schemas for warehouse modeling have been proposed over the years (see [28] and references therein). Such

approaches are mainly proposals for modeling languages that are part of data warehouse design methodologies. By contrast, CIM [13] provides a run-time environment that allows a user to pose queries and do business analytics at conceptual and business levels.

On the industry side, two major vendors of business analytics solutions (namely SAP Business Objects and IBM Cognos) provide proprietary conceptual levels that they call "semantic layers". SAP Business Objects' semantic layer [29], called a *Universe*, is a business representation of an organization's data asset (i.e., data warehouse as well as transactional databases). IBM Cognos' semantic layer, Framework Manager [14], is similar to SAP's Universes and works according to the same principles.

In contrast to these approaches, the EDM Framework [30] provides a querying and programming platform that raises the level of abstraction from the logical relational data level to Peter Chen's Entiry-Relationship (ER) conceptual level [31]. EDM consists of a conceptual model, a relational database schema, mappings between them and a query language (Entity SQL – eSQL), over the conceptual model. A compiler generates the mapping information in the form of eSQL views that express ER constructs in terms of relational tables. Unlike CIM, which deals with the multidimensional data model, EDM deals with the classical relational data model.

8 Evaluation

The cyclic approach of Model, Implement, Evaluate, Analyze is not new. It is a classical approach to managing strategic initiatives that is taught in business schools, and is carried out by organizations around the world. However, the gap between the business view, the technical data view, and the results reported is quite large and is bridged largely in a manual, ad hoc fashion. Our approach leverages models to structure, systematize and automate the process, and provides analysts with novel tools that they do not currently have:

- A structured representation of the business view of a strategic initiative, which links goals to tasks that accomplish them and indicators that measure them.
- A conceptual view of data that collects the required data from disparate data sources across the organization in order to compute the indicators and report on them in a dashboard.
- A systematic approach to mapping from business view to conceptual view to dashboard.
- An opportunity for tool-based support for analysts to design, implement, and manage strategic initiatives.
- Formal mappings that ensure that changes to goals indicators, dashboards, and data sources can be flexibly accommodated, facilitating maintenance.
- Better support in the cognitive gaps between the business and technological points of view, which allows savings in terms of time, accuracy, and other qualities during the implementation phase.

9 Conclusions and Future Work

We have presented a model-based framework for bridging the business and technological levels within organizations. The workings of the framework are demonstrated through a case study that involves managing the strategic initiative lifecycle at a teaching hospital implementing an initiative intended to reduce antibiotic resistant infections. The case study demonstrates how the framework works, but also how it can help bridge cognitive gaps and reduce the need for manual processing.

Our plans for future work include fleshing out the framework and supporting it with tools that automate or semi-automate some of the implementation tasks. With this aim, the concepts of *flexibility* and *adaptability* defined in [32] will be investigated and applied to our approach to: i) satisfy the changing data analysis requirements of business users; and ii) cope with changes in local data sources. This will allow for the delivery of timely and accurate BI to business users.

Acknowledgments. This work has been supported by the Business Intelligence Network (BIN) and the Natural Sciences and Engineering Research Council of Canada. We are grateful to Eric Yu, Iluju Kiringa and many other colleagues for useful discussions and suggestions that helped shape this work.

References

1. Vonderheide-Liem, D.N., Pate, B.: Applying quality methodologies to improve healthcare: Six sigma, lean thinking, balanced scorecard, and more. HCPro, Inc. (2004)
2. Pourshahid, A., Amyot, D., Peyton, L., Ghanavati, S., Chen, P., Weiss, M., Forster, A.J.: Business process management with the User Requirements Notation. Electronic Commerce Research 9(4), 269–316 (2009)
3. Kronz, A.: Managing of process key performance indicators as part of the aris methodology. In: Corporate Performance Management, pp. 31–44. Springer, Heidelberg (2006)
4. Nadler, D.A., Shaw, R.B., Walton, A.E., Associates: Discontinuous change: Leading organizational transformation. JOSSEY-BASS, An Imprint of WILEY (1994)
5. Galliers, R.D., Baets, W.R.J. (eds.): Information technology and organizational transformation: Innovation for the 21st century organization. John Wiley Series in Information Systems (1998)
6. Burgin, A.L., Koss, E.: Transformation to High Performance. A journey in organizational learning. Report No. 823 (Summer 1993)
7. Negash, S.: Business intelligence. In: CAIS, vol. 13(15) (2004)
8. Watson, H.J., Wixom, B.H.: The current state of business intelligence. Computer 40, 96–99 (2007)
9. Barone, D., Mylopoulos, J., Jiang, L., Amyot, D.: Business Intelligence Model, version 1.0. Technical Report CSRG-607, University of Toronto (March 2010), ftp://ftp.cs.toronto.edu/csri-technical-reports/INDEX.html
10. Kaplan, R.S., Norton, D.P.: Balanced Scorecard: translating strategy into action. Harvard Business School Press, Boston (1996)
11. Kaplan, R.S., Norton, D.P.: Strategy maps: Converting intangible assets into tangible outcomes. Harvard Business School Press, Boston (2004)

12. Melnik, S., Adya, A., Bernstein, P.A.: Compiling mappings to bridge applications and databases. In: SIGMOD, pp. 461–472. ACM, New York (2007)
13. Rizzolo, F., Kiringa, I., Pottinger, R., Wong, K.: The conceptual integration modeling framework: Abstracting from the multidimensional, arXiv:1009.0255 (2010)
14. Volitich, D.: IBM Cognos 8 Business Intelligence: The Official Guide. McGraw-Hill, New York (2008)
15. Mazzeo, F., Capuano, A., Avolio, A., Filippelli, A., Rossi, F.: Hospital-based intensive monitoring of antibiotic-induced adverse events in a university hospital. Pharmacological Research 51(3), 269–274 (2005)
16. Salama, S., Rotstein, C., Mandell, L.: A multidisciplinary hospital-based antimicrobial use program: Impact on hospital pharmacy expenditures and drug use. Can. J. Infect. Dis. 7(2), 104–109 (1996)
17. Barone, D., Yu, E., Won, J., Jiang, L., Mylopoulos, J.: Enterprise modeling for business intelligence. PoEM (2010)
18. Dealtry, T.R.: Dynamic SWOT Analysis. Dynamic SWOT Associates (1994)
19. Amyot, D., Horkoff, J., Gross, D., Mussbacher, G.: A lightweight GRL profile for i* modeling. In: Heuser, C.A., Pernul, G. (eds.) ER 2009. LNCS, vol. 5833, pp. 254–264. Springer, Heidelberg (2009)
20. Lenzerini, M.: Data integration: a theoretical perspective. In: PODS, New York, NY, USA, pp. 233–246 (2002)
21. Nargesian, F., Rizzolo, F., Kiringa, I., Pottinger, R.: Bridging decision applications and multidimensional databases. SITE, University of Ottawa, University of Ottawa (2010)
22. Yu, E.: Towards modelling and reasoning support for early-phase requirements engineering. In: RE 1997, Washington, USA (1997)
23. International Telecommunication Union: Recommendation Z.151 (11/08): User Requirements Notation (URN) – Language definition, http://www.itu.int/rec/T-REC-Z.151/en
24. Andersson, B., Johannesson, P., Zdravkovic, J.: Aligning goals and services through goal and business modelling. Inf. Syst. E-Business Management 7(2) (2009)
25. Howard, R., Matheson, J.: Influence diagrams. Readings on the Principles and Applications of Decision Analysis II (1984)
26. Business Rules Group: The Business Motivation Model: Business Governance in a Volatile World. Ver. 1.3 (2007), http://www.businessrulesgroup.org/bmm.shtml
27. Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., Schneider, L.: Sweetening ontologies with DOLCE. In: Gómez-Pérez, A., Benjamins, V.R. (eds.) EKAW 2002. LNCS (LNAI), vol. 2473, p. 166. Springer, Heidelberg (2002)
28. Malinowski, E., Zimanyi, E.: Advanced Data Warehouse Design: From Coventional to Spatial and Temporal Applications. Springer, Berlin (2008)
29. Howson, C.: BusinessObjects XI (Release 2): The Complete Reference. McGraw-Hill, New York (2006)
30. Adya, A., Blakeley, J.A., Melnik, S., Muralidhar, S., the ADO.NET Team: Anatomy of the ADO.NET Entity Framework. SIGMOD (2007)
31. Chen, P.P.S.: The entity-relationship model—toward a unified view of data. ACM Trans. Database Syst. 1(1), 9–36 (1976)
32. Cheung, W., Babin, G.: A metadatabase-enabled executive information system (part A): a flexible and adaptable architecture. DSS 42, 1589–1598 (2006)

Cloud Computing Providers: Characteristics and Recommendations

Maciej Lecznar[1] and Susanne Patig[2]

[1] SBB AG, Lindenhofstrasse 1,
CH-3048 Worblaufen, Switzerland
maciej.lecznar@sbb.ch
[2] Institute of Information Systems, University of Bern, Engehaldenstrasse 8
CH-3012 Bern, Switzerland
susanne.patig@iwi.unibe.ch

Abstract. This paper deals with the current reality of cloud computing: First, based on a historical analysis, the core characteristics of cloud computing are summarized and joined together in a general definition; then, two classifications of cloud computing are presented. These results are used to assess the existing cloud computing providers. A sample of success stories is analyzed to find out which benefits were realized by current cloud computing consumers, and we contrast our findings with the ones of other empirical investigations. Finally, the risks of cloud computing are collected from the relevant literature to derive some recommendations.

Keywords: Cloud computing, cloud computing providers, benefits, risks.

1 Introduction

The nature of the term 'cloud computing' is still rather vague [8]: Some see cloud computing as the revolution of computer science [16], others as the evolution of service-oriented architecture [17] or grid computing [28] and the cynics as a new marketing label for well-known practices [9]. Nevertheless, cloud computing can help in reducing the costs for information technology (IT), and for that reason many companies intend to use it. However, these companies are confronted with the following difficulties: First, from a theoretical point of view, the definition and characteristics of cloud computing are not clear. Secondly, several cloud computing providers offer distinct cloud computing services at specific prices, which makes it difficult to compare them. Thirdly, cloud computing carries some risks that should be contrasted with its benefits before it becomes the basis of a company's everyday business.

Our paper addresses these difficulties: In Section 2 we derive a general definition of cloud computing and give two classifications of 'clouds'. The results of Section 2 are used to assess current cloud computing providers in Section 3.1. Based on an analysis of success stories, Section 3.2 shows the benefits realized by the current consumers of each cloud computing provider. Section 4 classifies the risk of cloud computing, and Section 5 gives some recommendations.

G. Babin, K. Stanoevska-Slabeva, P. Kropf (Eds.): MCETECH 2011, LNBIP 78, pp. 32–45, 2011.

2 Cloud Computing

2.1 Defining Cloud Computing

The idea of cloud computing is still evolving. Table 1 shows the most popular definitions[1]. All of them mention several *characteristics of cloud computing*, which have (in spite of the differing terms) the following generalized intensions:

1. *Shared IT resources*: Cloud computing pools IT resources to serve several consumers.
2. *Capabilities as IT services*: What is provided by the shared IT resources (qualitative point of view) meets the consumers' specific needs (see Section 2.2) and is available 'off-the-shelf'.
3. *Elastic*: The provided IT capabilities can be automatically adapted to the varying quantitative demand of the consumers at any time.
4. *Measured by use*: The (quantitative) usage of the IT capabilities is tracked by metrics to provide transparency for both the provider and consumer of cloud computing.
5. *Broad network/Internet access*: All IT capabilities are available over a network and accessed via web-based technologies

These generalized characteristics of cloud computing can be summarized in the following definition, which we use throughout this paper: *Cloud computing* provides elastic and metered IT capabilities, which are provided by shared IT resources and delivered on demand by web-based technologies.

Table 1. Definitions of cloud computing

		Forrester [27]	**Gartner [20]**	**U.S. NIST [18]**
Definition		Standardized IT capability (services, software, or infrastructure) delivered via Internet technologies in a pay-per-use, self-service way	Style of computing in which scalable and elastic IT-enabled capabilities are delivered as a service to external customers using Internet technologies	Model for enabling convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction
Characteristics	(2)	"IT capability (…) self-service way"	Service based	On-demand self-service
	(5)	"delivered via Internet technologies"	Uses Internet technologies	Broad network access
	(1)	-	Shared	Resource pooling
	(3)	-	Scalable and elastic	Rapid elasticity
	(4)	"pay-per-use"	Metered by use	Measured service

2.2 Classifications of Cloud Computing

In practice, distinct types of cloud computing exist, which differ in the provided IT capabilities and their accessibility. In the following, we present both classifications

[1] As we don't derive cloud computing from grid computing, the characteristics in Table 1 differ from the ones in [28].

[18], [21]. Concerning the (qualitative) *IT capabilities* that are provided and used, the following types of cloud computing can be identified [20]:

- *Software as a Service (SaaS)*: A (cloud computing[2]) consumer uses the (cloud computing[2]) providers' applications running on a cloud infrastructure without dealing with the applications' management or monitoring.
- *Platform as a Service (PaaS)*: A preconfigured infrastructure or application platform is provided where a consumer can deploy, test, monitor or develop his own applications, but does not control the underlying IT infrastructure such as storage or network devices.
- *Infrastructure as a Service (IaaS)*: A consumer uses IT resources (such as storage or network devices) and is allowed to deploy and run arbitrary software, which can include operating systems. The consumer controls operating systems, devices or even the selection of network components and, thus, the IT infrastructure.

These 'xyz as a service'-layers can be completed with 'Communication as a Service' and 'Monitoring as a Service' [21]. *Communication as a Service (CaaS)* means the management of hardware and software required to deliver voice over IP, instant messaging, and video conferencing. Though CaaS capabilities can replace infrastructure components such as phones or mobile devices, CaaS is based on communication applications (software solutions). Therefore, we include CaaS into the SaaS layer. *Monitoring as a Service (MaaS)* corresponds to outsourced IT security services because it comprises real-time, 24/7 monitoring and nearly immediate incident response across a security infrastructure [21]. As the security infrastructure consist of hardware such as routers, switches and servers, we subsume MaaS under IaaS.

According to the *accessibility* of a cloud's capabilities, the following types of cloud computing can be distinguished [18]:

- *Public cloud*: The cloud infrastructure is made available to the general public or a large industry group and is owned by an organization selling cloud computing services. The applications from several consumers coexist together on the cloud computing provider's infrastructure. Public clouds are most often hosted away from the consumers' premises, and, thus, reduce risk and cost by flexibly extending the consumers' IT infrastructure.
- *Community cloud*: The cloud computing infrastructure is shared by several organizations that have common concerns. It may be managed by these organizations or a third party and may exist on or off the consumers' premises. As an example, we refer to the SETI initiative [26], where each user adds a part of IT resources to the shared pool for long lasting calculations.
- *Private cloud*: The cloud infrastructure is operated solely for an organization and is built for the exclusive use by one consumer. Thus, this cloud computing type provides the utmost control over data, security, and quality of service. The consumer owns the cloud infrastructure and controls how applications are deployed on it. Private clouds can be built and managed by a company's own IT department or by a cloud computing provider.

[2] For brevity, we omit the terms 'cloud computing' in the following wherever appropriate.

- *Hybrid clouds* combine two or more clouds of distinct accessibility type (private, community, or public), which remain unique entities, but are bound together by standardized or proprietary technology to enable data and application portability. Hybrid clouds require determining how applications are distributed across public and private clouds. An issue that needs to be considered here is the relationship between processing and data: If the amount of data is small or the application is stateless, a hybrid cloud can be more successful than in a scenario where large amounts of data must be transferred from a private into a public cloud for a small amount of processing.

Table 2 compares the accessibility-driven types of cloud computing with the characteristics of 'clouds' in general (Section 2.1). To improve the separation among the types, we have introduced the attributes 'physical location of the cloud infrastructure', 'provider' and 'consumer'. We ignore hybrid clouds in Table 2 because they are defined as a composition of several accessibility types.

Private clouds are built within one enterprise; so, the shared IT resources don't serve several consumers, and the IT services cannot be considered as 'ready to use, direct from the shelf'. Moreover, in a private cloud the consumer owns the IT infrastructure and provides its capabilities within the organization. In that manner we doubt whether a private cloud still is cloud computing and suggest that it should be treated as an evolution of enterprise data centers, where the new focus is on delivering IT services instead of applications.

Table 2. Characterization of cloud computing by accessibility

	Cloud computing	**Public cloud**	**Community cloud**	**Private cloud**
Characteristics	(1) Shared IT resources	Yes	Yes	Not supported in full
	(2) Capabilities as services	Yes	Yes	Not supported in full
	(3) Elastic	Yes	Yes	Yes
	(4) Measured by use	Yes	Yes	Yes
	(5) Internet access	Yes	Yes	Yes
Attributes	Location of infrastructure	At provider's site.	In the community.	At consumer's site.
	Provider	External provider.	Community member or external provider.	Consumer's IT or external provider.
	Consumer	Enterprise, community members, Internet user.	Community members.	Part of or whole enterprise.

3 Market of Cloud Computing

3.1 Cloud Computing Providers

Because of the economies of scale in providing large amounts of IT resources, the market of cloud computing is dominated by the 'big players' in IT, namely Amazon.com, Google, Microsoft, Oracle and Saleforce.com. In this section we investigate and compare their solutions and pricing models. Table 3 characterizes the providers in terms of the clouds' general characteristics (Section 2.1), capabilities and

accessibility type (Section 2.2) as well as costs. We used the following *scenario* (see also [3] and [7]) to calculate and compare the costs of cloud computing for one month: (1) *storage* of 2 TB (Terabyte) of data, (2) *computation* requiring 50 hours of CPU/instance time per month and (3) *data transfer* of (3a) 2 TB (initial transfer for storage; only first month) or (3b) 500 GB (Gigabyte) per month for computation, respectively. The prices we state in the following sections were gathered from the Web sites of the providers and refer to May 2010. The distinction between the characteristics derived from the providers' Web sites (not shaded in Table 3) and perceived by the consumers (shaded in Table 3) is explained in Section 3.2. Risks of cloud computing and risk management are discussed in Section 4.

Table 3. Properties of cloud computing

Provider	Cloud computing			Pricing		Risk Management
	Type		Characteristics (as perceived)			
	Access	Capabilities		Model	♦Scenario values (month)	
Amazon	Public	IaaS, PaaS, SaaS (with software vendors)	(1) (2) (3) (4) (5)	Instances (number, usage hours), data volume (stored, transferred), PUT/ GET requests	(1) US$ 307 (2) US$ 9 to 144 (3) US$ 307/75	Security management and best practices are documented
Google	Public	PaaS, SaaS	(1) (2) (3) (4) (5)	CPU usage hours, stored data volume, number of applications; Free of charge	(1) US$ 307 (2) US$ 5 (3) –	Unknown
Microsoft	Public	IaaS, PaaS, SaaS	(1) (2) (3) (4) (5)	Data volume (stored, transferred), CPU usage hours; Subscription (with limitation)	(1) US$ 307 (2) US$ 6 (3) US$ 205/50	Unknown
Sales-force	Hybrid, Public, Private	PaaS, SaaS	(2) (3) (5) restricted (1) perceived (3)	Fee per user account (licence) without any resource limitation.	–	Hybrid clouds for one consumer
Oracle	Private	PaaS, SaaS	(2) (5) restricted (3) perceived (4)		–	Private clouds for one consumer

♦ *Scenario values*: (1) Data storage: 2 TB; (2) Time for computation: 50h/month; (3) Data transfer: 2 TB non-recurring/500 GB/month.

3.1.1 Amazon.com

In early 2006 Amazon Web Services (AWS) started to provide companies of all sizes with IT services out of 'a cloud'. Currently the backbone of the Amazon cloud includes the following solutions [1]:

- *Amazon Elastic Compute Cloud* (Amazon EC2) is a Web service that provides resizable computing capacity with virtual machines as individual instances and a

configurable infrastructure. The pricing is per consumed instance-hour for each type of the virtual machine.

- *Amazon Simple Storage Service* (Amazon S3) is a simple Web service interface that can be used to store and retrieve large amounts of data. Charges are based on data storage volume.
- *Amazon CloudFront* is a content delivery Web service that allows the distribution of content to end users. The fees are calculated from the amount of transferred outgoing data.
- *Amazon SimpleDB* stores and executes queries in real time. The pricing model uses the Amazon SimpleDB machine utilisation hours and data transfer.
- *Amazon Simple Queue Service* (Amazon SQS) is a hosted queue for storing messages. It enables moving data between distributed components of applications without losing messages.

We associate the Amazon.com solutions with PaaS (EC2) and IaaS (S3, CloudFront, SimpleDB, SQS) as they are related to platform and infrastructure, respectively. Apart from order fulfillment (Amazon Fulfillment Web Service FWS) and payment (Amazon Flexible Payments Service FPS), there is a gap in the SaaS layer, which is filled by alliances with software vendors: For instance, Amazon.com and SAP provide a scalable environment with SAP ERP, CRM and BI products [25].

The Amazon.com cloud has all cloud characteristics of Section 2.1. The pooled IT resources provide services for Internet users, i.e., it is a public cloud. The pricing model agrees with 'measured by use': The fee for Amazon EC2, SimpleDB and the SAP products is based on usage hours of instances. According to the distinct specifications of the instances, the prices range from US$ 0.18 to 2.88 per hour [1]. In case of Amazon Cloud Front and SQS, a user pays for the volume of data transfer: US$ 0.15/GB for the first 10 TB to US$ 0.03/GB for over 1,000 TB. Due to fee progression, e.g., the cost for stored data in Amazon S3 decreases from US$ 0.15/GB for the first 50 TB to, finally, US$ 0.055/GB for more than 5,000 TB [1]. Consequently, consumers are encouraged to utilize Amazon.com solutions on a large scale.

3.1.2 Google

Currently Google acts in the cloud computing market with the following solutions [11]:

- *Google App Engine* provides the capability of running the consumers' Web applications on Google's preconfigured infrastructure.
- *Google Docs* offers tools for the creation and storage of office documents, e.g., spreadsheets, presentations, calendars, as well as a collaboration environment.

Both solutions are distributed to consumers via the Internet and satisfy the properties of PaaS (App Engine) or SaaS (Docs). As the infrastructure is available to the general public, we categorize the Google cloud as a public one. The billing policy is founded on the 'free of charge' model, where the consumers need to create a Google account, or based on the used resources, e.g., the CPU time (US$ 0.10 per hour), the volume of stored data (US$ 0.15 per GB per month) or the number of deployed applications ($8 per user per month each).

3.1.3 Microsoft

The core of Microsoft's cloud is the Windows Azure platform with the following components [5]:

- *Windows Azure* provides a Windows-based environment (platform) for running applications and storing data on servers in Microsoft data centers.
- *SQL Azure* provides data services in the cloud based on SQL Server.
- *Windows Azure platform AppFabric* provides infrastructure services to connect applications running in 'the cloud' or locally.

Altogether, Windows Azure supports a consumer by platform, software and infrastructure services, therefore covers all three levels of cloud capabilities (PaaS, SaaS and IaaS). The pricing model is 'pay-per-use'; usage is calculated based on CPU time (US$ 0.12 per hour), storage space (US$ 0.15 per GB per month) or data transfer (US$ 0.10 to 0.45 per GB). Alternatively users can subscribe, and the monthly base fee includes defined amounts of allowed resource usage. Microsoft cloud computing solutions are available to anyone, and therefore the cloud is public [30].

3.1.4 Salesforce.com

Salesforce.com released the first version of its online products in 2001 [23]. Currently, the company offers three cloud products [22]:

- *Salesforce CRM* offers CRM and ERP applications based on the provider's or the consumer's IT infrastructure.
- *Force.com Platform* is the platform for business applications.
- *Chatter* enables collaborating with people at work.

Thus, the cloud computing solution portfolio of Salesforce.com covers SaaS and PaaS; Chatter corresponds to a CaaS solution, which we classified as SaaS in Section 2.2. Because of the fact that the infrastructure resides either on the provider's or consumer's location, hybrid clouds can be built.

The Salesforce.com cloud does not cover all characteristics of Section 2.1: Firstly, the charges rely on the users' licenses, which are valid for a user and an application without any limitation of time or resources and, thus, contradict our cloud characteristic 'measured by use' (4). Secondly, Salesforce CRM can be placed in a private cloud, which violates our characteristic (1), i.e., 'shared IT resources'.

3.1.5 Oracle

Oracle offers two types of cloud capabilities [19]:

- *Oracle on Demand* is a SaaS cloud as it provides CRM and ERP applications for enterprises.
- *Oracle Platform for SaaS* is the underlying middleware platform and, thus, PaaS.

Both products are delivered by private clouds, where a consumer pays the licenses per user account. Moreover, Oracle's cloud services work on the IT infrastructure provided by the consumer. Due to these facts the coverage of the cloud characteristics is limited: The characteristics 'shared IT resources' (1) and 'measured by use' (4) are not fully supported; support for 'elastic' (3) - via virtual machines - can be assumed.

3.2 Cloud Computing Consumers

Every cloud computing provider has its consumers (see Table 4), whose decisions on whether or not to use cloud computing usually depend on an analysis of benefits and risks. Here, we concentrate on the benefits. Based on the success stories published at the cloud computing providers' Web sites in May 2010, Table 4 summarizes why the consumers decided to use cloud computing. We have used the English Web sites; success stories in the form of video or audio streams were ignored. The total number of analyzed consumers amounts to 35 - which is limited, but gives first indications for typical benefits. Companies of any size are represented.

The column 'reasons' in Table 4 shows the wording from the success stories, which we translated in the characteristics of cloud computing (see Table 2). This translation is based on the assumption that the reasons to use cloud computing (*benefits*) are grounded in the characteristics of cloud computing, and it enables a comparison and quantitative assessment of the importance of particular benefits (see Table 5).

Almost half of the consumers of Table 4 named more than one reason to use cloud computing. In the following, we give the absolute frequency of each reason and the percentage of consumers that have mentioned it. Because of multiple answers per consumer, the sum of percentages exceeds 100%.

Scalability turned out to be the most important benefit (25/71%) of cloud computing. The second most important benefit (16/46%) is cost saving due to 'measured by use', followed by 'access from anywhere' (6/17%), 'capabilities as services' (5/14%) and, finally, 'shared IT resources' (4/11%).

Table 5 compares our results with two other studies: The first study was conducted in June and July 2009 by IBM [14] in a group of 1090 IT decision makers around the world. The study focused on the potential drivers behind cloud computing adoption: 77% of the respondents chose cost saving as a key driver. The faster 'time-to-value', including being able to scale IT resources to meet needs ('elasticity'), was mentioned by 72 % of the respondents. The fact that cloud capabilities are provided as IT services off-the-shelf allows companies to focus on their core competences, which was an important benefit for 38% of the IT decision makers. Finally, 37% of the respondents believe in a reduction of in-house IT infrastructure because of sharing IT resources.

The European Network and Information Security Agency (ENISA) launched the online survey "An SME perspective on cloud computing" on the 16th of April 2009, where the reasons behind a possible engagement in cloud computing were gathered. Till the 1st of November 2009, 74 answers were collected [6]. This study shows that 'avoiding capital expenditure' was the most important benefit of cloud computing for 68% of the respondents. The second most important benefits (63,9%) were 'flexibility' and 'scalability' of IT resources, followed by increased computing capacity and business performance as a result of the cloud's IT services, i.e., the cloud's capabilities (36%). 29% of the respondents found 'access from anywhere' important. Finally, the optimization of the shared IT infrastructure showed up as an important factor for one fourth (25%) of the study's participants.

Table 4. Consumers' reasons to choose cloud computing solutions, ordered by provider

Reasons	Character-istics	Consumers	Providers
"scalable"	(3)	Ipswitch	Amazon [2]
"elastic", "cost-effective", "easily accessible"	(3) (4) (5)	Moonwalk	
"granular pricing", "elastic scale"	(3) (4)	Sonian	
"increase system robustness without increasing costs", "multiple location availability", "scalability"	(3) (4) (5) (R)	HiperStratus	
"the speed and easy of deploying new instances"	(2)	July Systems	
"avoiding buying and managing computers"	(1)	Cycle Computing	
"stable, robust, flexible, and low cost"	(3) (4) (R)	Harvard Medical School	
"needed access to more computing capacity than we could possibly maintain internally "	(1) (3)	Pathwork Diagnostics	
"speed of system deployment and roll-out"	(2)	Hitachi Systems	
"scalability, reliability and the detailed reporting"	(3) (4) (R)	Tubaah	
"perfect match for the pay as you go model"	(4)	SearchBlox	
"cost structure and scalability"	(3) (4)	Digitaria	
"scale the hardware very, very quickly"	(3)	Virgin Atlantic Airlines	
"scalability, reliability, and utility pricing"	(3) (4) (R)	DreamFactory	
"maximal utilization of resources", "pay for actual usage", "have a scalable solution"	(1) (3) (4)	MarketSimplified	
"reduce the friction of scaling"	(3)	Morph AppSpace	
"scalable experience that has the minimum footprint on local network"	(3) (5)	Napera Networks	
"to scale capacity on demand", "per-hour pricing"	(3) (4)	PostRank	
"take advantage of the $2B+ that Amazon has invested"	(1)	StarPound	
"easily-scaled services "	(3)	CapGemini	Google [10]
"flexibility, price, reliability, and security"	(3) (4) (R)	San Francisco Consulting	
"track site performance on a granular lever "	(4)	The Huffing Post	
"to create a highly scalable, global solution"	(3) (5)	Quark	Microsoft [29]
"to provide scalability"	(3)	Sitemasher	
"highly elastic needs"	(3)	TicketDirect	
"scale up and down ", "provisioning, billing, and metering capabilities"	(3) (4)	Wipro	
"to provide applications to customers everywhere, without deploying data centres around the world"	(5)	3M	
"the pricing model is easy to calculate"	(4)	City of Miami	
"we can respond to large-scale peaks in demand"	(3)	European Environment Agency	
"our challenge is scaling"	(3)	Origin Digital/Accenture	
"Having the whole system implemented in-house had made it expensive to distribute software", "implement rapidly and that would give us more agility", "flexibility and the cost factor were key drivers"	(2) (3) (4) (5)	Siemens	
"a system that could be developed and implemented quickly"	(2)	Konica Minolta	Salesforce [24]
"we needed simplicity"	(2)	Polycom	
"lever on cost, up or down, depending on which way we go with our business"	(3) (4)	Agilent	Oracle [20]
"the scalability and flexibility"	(3)	Exterran	

Abbreviation: (R): Reliability.

Both the IBM and the ENISA study asked *potential consumers* about the *expected benefits* that may prompt them to use cloud computing. In contrast, our results are based on benefits that are *realized* by *current* cloud computing consumers. The most important *expected* benefit is cost saving, while the most important *realized* benefit is scalability (see Table 5). Altogether, our analysis of success stories confirms both benefits as the two main ones of cloud computing. Moreover, in our analysis one *unanticipated benefit* emerged: reliability, which was mentioned by 5 of the current consumers (14%) in Table 4. Finally, the current consumers appreciate the access to their cloud solutions from anywhere; in contrast to the characteristic (5) from Section 2.1, 'Internet access', at least literally, is not required.

Table 5. Comparison of the reasons to choose cloud computing

Cloud computing benefits	Cloud computing characteristics	Success stories	IBM study [14]	ENISA study [6]
Scalability	(3)	71%	72%	63,9%
Cost saving	(4)	46%	77%	68%
Access from anywhere	(5)	17%	–	29%
Capabilities as services	(2)	14%	38%	36%
Reliability	–	14%	–	–
Resource pooling	(1)	11%	37%	25%

The deviating perception of cloud computing characteristics (see Table 3) can be explained as follows: First, the marketing departments of the cloud computing providers edit the success stories and control which characteristics are emphasized. Secondly, the characteristics (1) and (2), 'shared IT resources' and 'capabilities as services', are closely intertwined with particular cloud computing solutions and, thus, often 'overlooked' by the consumers. Moreover, these characteristics are not only applicable to cloud computing, but also to 'traditional' outsourcing. Thirdly, probably because of the private clouds provided, the characteristic 'Internet access' is not perceived for Oracle; for the clouds of Google this characteristic is not valued as a benefit as 'Internet access' is the key business of this company. Whether or not the Salesforce solutions are scalable was probably not known for the two consumers analyzed; however, it can be assumed according to the information at the provider's Web site.

4 Cloud Computing Risks

Because of the benefits listed in Section 3.2, consumers are keen on using cloud computing, but they should be aware of the potential risks. Cloud computing risks are rarely reported in success stories. Therefore we base our discussion on the existing surveys and literature [3], [4], [6], [12], [13]. Table 6 summarizes our classification of risks, which is explained in the following.

Table 6. Cloud computing risks, their relevance and management measures

Organizational Risks			**Technical Risks**			**Legal Risks**		
Type	*To*	*Action*	*Type*	*To*	*Action*	*Type*	*To*	*Action*
(1) Loss of governance (1a) Cloud provider acquisition	++ S, + C G	P (P)	(5) Security (5a) Malicious insider (5b) Data leakage in transit	 ++ S, +C G	P(RM) P Y, C	(9) Compliance, auditability (9a) Change of jurisdiction	G ++ S, +C	P –
(2) Co-tenant vulnerability	G	P	(6) Isolation failure	++ S	P	(10) Privacy	++ S, +C	P(RM)
(3) Cloud service termination	G	P	(7) Unavailability	G	P	(11) Subpoena, e-discovery	++ S, +C	–
(4) Lock-in	G	–	(8) Data transfer bottleneck	G	Y	(12) Licensing	++ C, +S	–

Abbreviations: G: General, S: Storage, C: Computation, ++/+: High/moderate risk
P: Provider selection, Y: Physical data transfer, C: Secure channel;
RM: Risk management.

Basically, three types of risks can be identified [7] that are, however, interrelated. Related risks are arranged in the same row in Table 6. *Organizational risks* stem from relying on an additional business partner (i.e., the cloud computing provider); *technical risks* arise from the use of an external, distributed IT infrastructure, and *legal risks* refer to regulations relevant for IT.

Loss of governance means that the cloud computing consumer (necessarily) cedes control over data to the cloud computing provider (or its buyer in case of an *acquisition*). This raises both the technical issue of *data security* (i.e., data must be kept safe from lost or corruption and under controlled access) and the legal problems of *privacy* (i.e., personally identifiable information must be protected) and *compliance* (i.e., difficulties with audits or certification processes; untransparent storage of data in multiple jurisdictions, which may, moreover, *change*). Typical technical security risks of cloud computing are *malicious insider*, i.e., an abuse of high privilege roles in the cloud provider's organization, and *data leakage in transit*, i.e., by sniffing, spoofing, man-in–the-middle attacks etc. Some of these risks are especially high (++) when data is *stored* at a cloud computing provider's infrastructure (e.g., malicious insider, privacy) and a little smaller (+) when cloud computing is only used for *computations*; see Table 6. Since both usage types (storage/computation) require data transfer from the consumer to the provider, data leakage in transit is a general risk of cloud computing.

Because of pooling resources, several cloud computing consumers usually share IT infrastructure. One consumer's bad behavior can affect the reputation of all other consumers (*co-tenant vulnerability*). If the provider's hardware is confiscated (as a result of *subpoena*), stored data of several consumers is at risk of disclosure. Moreover, the technical risk of *isolation failures* exists, i.e., the failure of mechanisms to separate storage or memory between tenants of the shared IT infrastructure.

Cloud computing consumers are prone to the technical risk of short-time *unavailability* of a service and the organizational risk that a cloud computing provider

changes its business and eventually *terminates* some service. In that case, the lack of standardization among cloud computing providers makes it extremely difficult to move data or applications from one site to another (*lock-in*). To aggravate this problem, current software *licenses* often restrict the IT infrastructure on which software can run, or they are not applicable to the cloud computing situation (e.g., per seats agreements, online license checks).

Either cloud computing usage type requires at least an initial *data transfer* (from the cloud computing consumer to the provider), which can become a *bottleneck* – as the following example illustrates (see also [5]): For the Amazon S3 service, a write bandwidth between 5 and 18 was measured [11]. If we apply a bandwidth of 20 Mbit/s to our scenario, the following (tremendous) upload times result:

- (**2TB** * 1024GB * 1024MB * 8Bit)/20 MBit/s = 16,777,216MBit/ 20 MBit/s = 838,860s ≈ **9.7 days**
- (**500GB** * 1024MB * 8Bit)/20 MBit/s = 4,096,000MBit/ 20 MBit/s = 204,800s = **56.8h** ≈ **2.4 days**

We have derived the cloud computing risks (1) to (12) in Table 6 from the relevant literature. Empirically, the top five concerns named by potential cloud computing consumers in the ENISA study [10] were confidentiality (94% of the respondents[3]), privacy (89%) as well as integrity of data (85%), availability (82%), and loss of control (74%).

The identification of risks is a prerequisite to risk assessment and minimization. Currently not enough empirical data exists to assess all risks of cloud computing, but at least the risk of unavailability is de facto disproved by the benefit of 'reliability' we observed (see Table 5). Measures to reduce the other risks are sketched in Section 5.

5 Conclusions

Companies that aim at scalability should consider cloud computing because scalability is the most important realized benefit. According to the ISACA, the risks of cloud computing still outweigh its benefits for 45% of the US business and IT professionals [15]. Thus, as a next step in deciding about cloud computing, companies should assess the relevant risks (see Section 4) and take measures to reduce them. The following *measures* can be recommended (see Table 6):

- Carefully decide about the *usage type,* as storing 'in the cloud' is more risky than 'computing' (see Table 6).
- *Ship data physically* on hard disks to avoid data transfer bottlenecks (recommended by Amazon.com for large data quantities [11]) and to reduce data transfer risks.
- *Secure channels* to reduce data leakage in transit.
- Do not only use cost criteria to *select* a cloud computing *provider,* but also consider its reputation and risk management strategies.

[3] The percentages are calculated by dividing the sum of counts for the ratings 'very important' and 'showstopper' by the response count.

Table 3 shows the risk management strategies of the analyzed cloud computing providers (as far as they are communicated): For the Amazon Services, security management descriptions and best practices [1] are given. Moreover, the private or hybrid clouds of Salesforce.com and Oracle, which are designed for one consumer, increase privacy. In case of Microsoft and Google we were not able to find information about risk management.

Altogether (see Table 3), Amazon.com shows all cloud computing characteristics, has not only the largest number of perceived benefits, but also the most detailed strategies for risk management; its pricing is comparable and its reputation sound. Thus, our results suggest relying on Amazon.com as a cloud computing provider.

The validity of our quantitative results is limited by the following facts: First, the success stories were published by the cloud computing providers and, thus, are probably biased. Secondly, we did not consider all success stories of a particular provider, but only those with explicit, literal statements on the reasons to use cloud computing. Third, the sample period was May 2010, and the number of cloud computing consumers has been constantly increasing since then. The first limitation must be accepted, the second and third one will be removed in future research where we use a coding schema to gather the benefits of cloud computing from free texts. Based on the increased amount of data that will result from the coding, we can do a more sophisticated statistical analysis of the benefits of cloud computing.

Since decisions about cloud computing have to balance benefits and risks, our next research task is a detailed empirical investigation of cloud computing risks and the measures used to reduce them.

References

1. Amazon: Amazon Web Services, http://aws.amazon.com
2. Amazon: Case Studies, http://aws.amazon.com/solutions/case-studies/
3. Armbrust, M., et al.: Above the Clouds: A Berkeley View of Cloud Computing. Technical Report No. UCB/EECS-2009-28. UC Berkeley Reliable Adaptive Distributed Systems Laboratory (2009), http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.pdf
4. Catteddu, D., Hogben, G.: Cloud Computing. Benefits, risks and recommendation for information security. Technical Report. ENISA (2009), http://www.enisa.europa.eu/act/rm/files/deliverables/cloud-computing-risk-assessment/at_download/fullReport
5. Chappell, D.: Introducing the Windows Azure Platform. Technical Report. David Chappell & Associate (2009), http://www.davidchappell.com/writing/white_papers/Windows_Azure_Platform_v1.3-Chappell.pdf
6. European Network and Information Security Agency (ENISA): An SME perspective on Cloud Computing. Survey (2010), http://www.coe.int/t/dghl/cooperation/economiccrime/cybercrime/cy-activity-interface-2010/presentations/Outlook/SURVEY_An_SME_perspective_on_Cloud_Computing.pdf
7. Garfinkel, S.: An Evaluation of Amazon's Grid Computing Services: EC2, S3 and SQS. Technical Report TR-08-07, Harvard University (2007)
8. Geelan, J.: Twenty-One Experts Define Cloud Computing. Cloud Computing Journal (2009), http://cloudcomputing.sys-con.com/node/612375

9. Gillet, A.: Cloud Computing - Interview at the MIT campus. Beet.tv (2008), http://www.beet.tv/2008/09/cloud-computing.html
10. Google: Google Apps Case Studies, http://www.google.com/apps/intl/en/business/customer_story.html
11. Google: Google App Engine – Google Code, http://code.google.com/appengine/
12. Gregg, M.: 10 Security Concerns for Cloud Computing. White Paper, Global Knowledge Training LLC (2009), http://www.globalknowledgesa.com/pdf/WP_VI_10SecurityConcernsCloudComputing.pdf
13. Heiser, J., Nicolett, M.: Assessing the Security Risks of Cloud Computing. Gartner, Document ID 685308 (2008)
14. IBM: Dispelling the vapour around cloud computing. Drivers, barriers and considerations for public and private cloud adoption. IBM White Paper (2010), ftp://ftp.software.ibm.com/common/ssi/sa/wh/n/ciw03062usen/CIW03062USEN.PDF
15. ISACA: IT Risk/Reward Barometer – US Edition. ISACA (2009), http://www.isaca.org/AMTemplate.cfm?Section=20102&Template=/ContentManagement/ContentDisplay.cfm&ContentID=56656
16. Jennings, C.: The cloud computing revolution. ComputerWeekly.com, (December 22, 2008), http://www.computerweekly.com/Articles/2008/12/22/234026/the-cloud-computing-revolution.htm
17. Jetter, M.: Cloud Computing: Evolution in der Technik, Revolution im Geschäft. BITKOM (2009)
18. Mell, P., Grance, T.: The NIST Definition of Cloud Computing. National Institute of Standard and Technology, NIST (July 10, 2009), http://csrc.nist.gov/groups/SNS/cloud-computing/cloud-def-v15.doc
19. Oracle: CRM On Demand | Oracle On Demand | Oracle, http://www.oracle.com/us/products/ondemand/index.html
20. Plummer, D.C., et al.: Cloud Computing: Defining and Describing an Emerging Phenomenon. Gartner Research, Document ID 697413 (2008)
21. Rittinghouse, J.W., Ransome, J.F.: Cloud Computing. Implementation, Management and Security (2010)
22. Salesforce: Application Development with the Force.com Cloud Computing Platform, http://www.salesforce.com/platform/
23. Salesforce: CRM Press Releases, http://www. alesforce.com/company/news-press/press-releases/
24. Salesforce: CRM Snapshots and Case Studies & Force.com Testimonials, http://www.salesforce.com/customers/
25. SAP: Virtualization at SAP, http://www.sdn.sap.com/irj/sdn/virtualization
26. SETI: Institute Homepage, http://www.seti.org
27. Staten, J.: Cloud Computing for the Enterprise. Forrester Research (2009), http://www.forrester.com/imagesV2/uplmisc/CloudComputingWebinarSlideDeck.pdf
28. Weinhardt, C., et al.: Cloud-Computing – A Classification, Business Models and Research Direction. Business & Information Systems Engineering 51, 453–462 (2009)
29. Windows: Windows Azure Case Studies | Microsoft Cloud Evidence | Windows Azure Platform, http://www.microsoft.com/windowsazure/evidence/
30. Windows: Windows Azure Platform | Cloud Computing | Online Service | Data Storage, http://www.microsoft.com/windowsazure/

Evolution of Goal-Driven Pattern Families for Business Process Modeling

Saeed Ahmadi Behnam and Daniel Amyot

School of Information Technology and Engineering, University of Ottawa, Canada
sahma088@uottawa.ca, damyot@site.uottawa.ca

Abstract. Using patterns is a well-known approach for increasing reusability. A pattern-based framework that lays down a foundation for capturing knowledge about business goals and processes and customizing it for specific organizations in a given domain is hence valuable. However, the problems and solutions within a domain are always changing. Consequently, such framework can be useful only if it can evolve over time. In this paper, we propose and formalize a mechanism for evolving a pattern-based framework for goal-driven business process models. We demonstrate its feasibility with an example from the patient safety domain that illustrates how to evolve a pattern family with our extension algorithm.

Keywords: business process models, framework, goal models, healthcare, patient safety, pattern, User Requirements Notation.

1 Introduction

The value of software applications to an organization is based on how well business goals are satisfied through their use. When developing such applications, organizations often have difficulties in properly identifying and documenting their goals, their business processes, and the links between these two views [1]. In addition, although the gap between business processes and software development is generally understood, the gap between business goals and business processes has received far less attention. Many software development projects yield disappointing results or are simply canceled because software applications and business processes are not aligned properly with business goals. For instance, in healthcare software applications, there are more failure stories than success stories as a result of the above difficulties [2].

Modeling business goals and processes separately is not sufficient to bridge this gap, and hence traceability must also be taken into account. In addition, as defining such models from scratch is challenging, it is becoming increasingly difficult to ignore the benefits of knowledge reusability [3]. Reusing domain knowledge captured in the form of *patterns* can often help address this issue. For instance, design patterns have been quite successful in the construction of software applications [4]. However, patterns that span business goals and processes are far

G. Babin, K. Stanoevska-Slabeva, P. Kropf (Eds.): MCETECH 2011, LNBIP 78, pp. 46–61, 2011.

less common, and reusing existing knowledge in this context remains an open problem [5].

Motivated by the above challenges, we have introduced a pattern-based framework based on the User Requirements Notation (URN, [6]) for reusing domain knowledge in the form of goal models and business process models [7]. In this framework, *pattern families*, i.e., collections of related patterns, are used for capturing and reusing domain knowledge. Patterns capture business goals and business processes along with links that define the realization relationships between them. These links indicate which business processes alternatively realize particular business goals. Capturing such links in models facilitates finding known solutions in the context of conditions and requirements of an organization. Hence, such links help bridging the gap between requirements of a particular organization and corresponding (existing) solutions.

However, domain knowledge about business goals and processes is changing over time at a more rapid pace than for design patterns. Therefore, pattern families can be useful only if they can adapt to the changes that happen in the domain and reflect the solutions for current recurring problems. Introducing mechanisms that systematically help maintain and evolve a pattern family is hence a necessity. Evolution is a gradual process where a pattern family changes into a different and better form. Adding new patterns, removing obsolete ones, modifying patterns, and combing pattern families are core aspects of family evolution. It is important that evolving a pattern family be not limited to evolving individual patterns but that it also involves the family itself, to preserve its integrity. In this paper, we propose the *Goal-oriented Pattern Family Framework* (GoPF) that enhances our previous pattern-based framework by improving its metamodel and adding an evolution mechanism for extending pattern families.

The paper is organized as follows. Section 2 presents background information on URN and patterns. Then, section 3 gives an overview of the framework and how it is formalized with the help of URN. Section 4 provides the extension algorithm and describes its application, which is then illustrated in Section 5 in the domain of patient safety. Sections 6 and 7 follow with a discussion, our conclusions, and future work.

2 Background

2.1 User Requirements Notation (URN)

The User Requirements Notation, a standard of the International Telecommunication Union (ITU-T Z.150 Series) [6,8], is intended for the elicitation, analysis, specification, and validation of requirements. URN is also suitable for the modeling and analysis of business goals and processes [9]. This standard contains two complementary modeling languages: the Goal-oriented Requirement Language (GRL) for goals and Use Case Maps (UCM) for scenarios and processes. GRL supports the graphical modeling of business goals, non-functional requirements (NFRs), alternatives, and rationales. Modeling goals and NFRs of stakeholders with GRL makes it possible to understand the problem that ought to be solved.

GRL enables business analysts and IT architects to model strategic goals and concerns using various types of intentional elements and relationships, as well as their stakeholders called *actors* (⬭). Core intentional elements include *goals* (⬭) for functional requirements, *softgoals* (⬭) for qualities and non-functional requirements, and *tasks* (⬭) for activities and alternative solutions. Intentional elements can also be linked by AND/OR decomposition and by contributions. Quantitative or qualitative contributions scale may also be used to present the effects of contributions. Fig. 4 (left) presents a GRL goal model that shows how the softgoal Take Action can be achieved by accomplishing sub-goals. Other goal models may further refine each of the sub-goals.

The Use Case Map (UCM) notation is a visual process modeling language for specifying causal scenarios and optionally binding their activities to an underlying structure of components. UCMs are used to model scenarios and processes in the form of causal relationships, linking together *responsibilities* (×) which may be assigned to *components* (▭). Responsibilities represent activities performed in a process whereas components represent actors, systems, and system parts. UCMs support most of the concepts used in common workflow modeling notations including *start points* (●), *end points* (▮) as well as alternative and concurrent flows. *Stubs* (◇) are containers for sub-maps and can be used to organize a complex model in a hierarchical structure. Furthermore, URN allows typed links to be established between modeling elements (e.g., goal and scenario model elements). These links are called *URN links*. Fig. 4 (right) illustrates two UCM diagrams that depict alternative processes that lead to Take Action. Finally, jUCMNav is an open source Eclipse plug-in used for creating, analyzing, and managing URN models [10]. It also supports the definition of URN profiles, which tailor URN to specific domains [11].

2.2 Patterns

Patterns are rules that express a relation between a problem, a solution, and a certain context [12]. They have been proposed to capture and categorize knowledge of recurring problems and give advice on possible solutions to those problems [4,12]. A pattern can be thought of as a reusable model that describes a need and solves a problem which may occur at different levels of abstraction (e.g., in the area of software engineering with design patterns [13], in conceptual modeling with analysis patterns [14], and in information system architectures with architecture patterns [15]).

Patterns also provide a description of the forces at play. Forces typically discuss the reasons for using the suggested solution for the given problem in the given context. The most significant contribution of patterns is perhaps describing forces and clarifying the trade-off among them. Because patterns modularize problems and solutions, pattern-based software applications are often robust when changes happen.

Reusable knowledge in patterns enables efficient transfer of skills and expertise in the domain. However, many pattern descriptions tend to focus on the solution

to a problem and not so much on the problem and forces that are involved [16]. In addition, traditional pattern descriptions are mostly expressed textually.

In this paper, patterns are described with URN [7] by formalizing a) the description of the problem and the forces that are involved with GRL's intentional elements as well as contribution and correlation links, b) the solution with UCM models that provide a more detailed description of the behavior and structure of the solution, c) the links between the problem depicted with GRL and solutions represented with UCM diagrams, and d) the links between individual patterns when one refines the other. Furthermore, we also formalize the pattern with URN in a way such that one pattern groups similar solutions that address a recurring problem, with their tradeoffs.

3 Goal-Oriented Pattern Family Framework (GoPF)

In previous work [7], we introduced a pattern-based framework and described how patterns can be useful for capturing and reusing the knowledge of the business domain. In this paper, we propose an enhanced framework, named *Goal-oriented Pattern Family* (GoPF) framework, to facilitate finding, documenting, and reusing problems and solutions that recur in a domain. Fig. 1 represents GoPF's architecture.

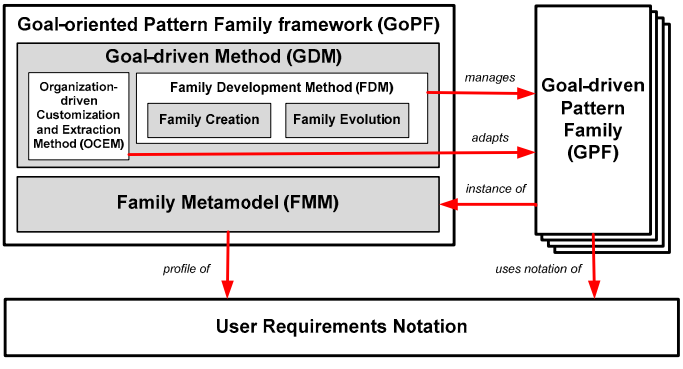


Fig. 1. Architecture of GoPF

GoPF is composed of a Family Metamodel (FMM) and a Goal-driven Method (GDM). FMM is a metamodel that lays down a structure for Goal-driven Pattern Families (GPF). A GPF captures the knowledge about a particular domain with patterns formalized with goals, business processes, and links between them. It specifies typical refinements of goals in terms of processes for a particular domain (e.g., patient safety or software development). A GPF is the key enabler for reusing knowledge.

The Goal-driven Method is composed of two major parts: (i) a Family Development Method (FDM), and (ii) the Organization-driven Customization and Extraction Method (OCEM). FDM provides algorithms for creating a GPF and evolving it over time. Our focus in this paper is on introducing the extension algorithm that evolves GPF, which leads to accuracy and overall quality. OCEM, described in [7], includes algorithms that help adapting instances of solutions that are most appropriate for particular organizations within the domain. OCEM uses a GPF as an input and assesses the impact of alternative solutions for achieving the high-level goals of a given organization in a step-by-step, top-down approach. Another input is an incomplete business goal model where only some of the high-level goals of an organization need to be identified. OCEM's main output is a more complete goal model combined with business processes aligned with the identified goals, as well as additional traceability links between the two views.

The FMM formalizes the concepts of GPF as a profile of URN, as shown in Fig. 2. The names between guillemets refer to corresponding metaclasses from the URN metamodel [2]. In URN, a *concern* is a model element that groups other model elements, including other concerns. URN metadata is used to associate stereotypes to model elements in a URN model that are part of this framework, as specified in Fig. 2 (e.g., a concern may be stereotyped as a «pattern»).

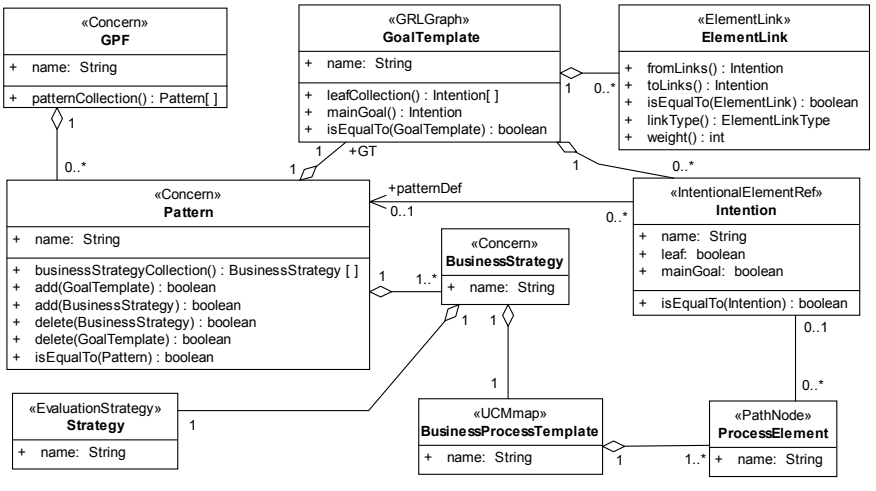


Fig. 2. Family Metamodel (FMM)

A GPF contains *patterns*, each of which includes one *goal template* (that formulates a problem and elements of its solutions) and at least one *business strategy* (that captures the arrangement of a solution along with its effect on the problem). Each goal template is essentially a GRL graph, and hence includes *intentions* (e.g., goals, tasks, and softgoals), and *element links* between them. The goal intentions contributing to the main goal of such a template can

themselves be refined by other patterns, through the *patternDef* relationship. Business strategies contain two main parts: a *strategy* (i.e., a regular GRL evaluation strategy, used for the evaluation of a goal model) and a corresponding *business process template* (i.e., a UCM map describing the process that specifies among other things the ordering of the goals selected by the strategy). In addition, goals and tasks in the goal template can be realized by *process elements* (e.g., stubs and responsibilities) in the business process template. Such *realization links* are supported with URN links. Further realization links between goals and business process templates are derived from existing associations (from Intention to BusinessProcessTemplate via patternDef). The decomposition of patterns into goal template and corresponding strategies enables organizing the problem and its solutions in a reusable manner. Depending on the complexity of the system, decompositions can recur to form a hierarchy of problems and solutions in a particular GPF. FMM, together with additional UML Object Constraint Language (OCL) constraints, not discussed in this paper, enforce consistency of pattern families.

4 Evolution of Goal-Driven Pattern Family

Evolution mechanisms keep the patterns in a GPF up-to-date so they can address current problems and solutions that stakeholders within the domain are facing. *Extension*, *modification*, *elimination*, and *combination* are four types of evolution mechanisms that can maintain the usefulness of a GPF by increasing the quality and accuracy of its patterns and their interrelationships. These mechanisms respectively evolve a GPF by (i) adding a new pattern, (ii) modifying a current pattern, (iii) eliminating an obsolete pattern, and (iv) combining two GPFs that represent problems of the same domain. Each mechanism must keep the integrity of GPF in addition to evolving individual patterns. In this paper, we only discuss the extension mechanism (Algorithm 1), supported by a short description of the modification mechanism (Appendix A). The extension algorithm adds a new pattern to the GPF and integrates it with existing patterns within the family. An extension is composed of three major steps. First, it modifies those patterns that are affected by the new pattern, then it adds the new pattern to the GPF, and, finally, it connects the new pattern to related patterns.

A *GPF analyst* is a modeler who observes the goal models and business processes used in organizations and locates the recurrent problems and solutions. When the observed recurrences highlight the need for adding a new pattern, this analyst prepares the inputs of the extension algorithm before its application.

This algorithm takes three inputs: pf is an initial GPF, xp is a pattern used to extend the initial GPF, and the modifications set (Table 1) highlights the effects that extending pf with xp has on other patterns of the family. The GPF analyst prepares the second and third inputs based on recurrences.

As different types of modifications may be necessary, the second, forth, and fifth elements may be *null*. The toLinks side of the link must always point to the main goal of rp. If the action is *Add* then the fromLinks side must point

Inputs of the modification algorithm
I1. *pf:GPF* /* initial pattern family */
I2. *modifications:* **set of** *(p: Pattern, link:ElementLink, action* ∈ *{**Add,Delete**}, bst:BusinessStrategy , oldbst:BusinessStrategy)* **where** *link.toLinks.isEqualTo(p.mainGoal())*

Output of the modification algorithm
O1. **modified** *pf:GPF* /* the modified pattern family */

Steps of the modification algorithm
for each *m* **in** *modifications*:
S1. *mp:Pattern = a pattern* ∈ *pf.patternCollection()* **where** *pattern.isEqualTo (m.p)*
/**m.p* is the first element of the *m */*
S2. **if** *(m.action == **Delete**)* **then**
S3. *mp.GT.delete(m.link)*
S4. **elseif** *(m.action == **Add**)* **then**
S5. *mp.GT.add(m.link)*
S6. **endif** /* if no action is provided then the GoalTemplate is unchanged */
S7. **if** *(m.oldbst* ≠ **null***)* **then**
S8. *pbst:BusinessStrategy = a businessStrategy* ∈ *mp.businessStrategyCollection()* **where** *businessStrategy.isEqualTo(m.oldbst)*
S9. *mp.delete(pbst)*
S10. **endif**
S11. **If** *(m.bst* ≠ **null***)* **then**
S12. *mp.add(m.bst)*
S13. **endif**

Algorithm 1. Extension of GPF

Table 1. Elements of the modifications set

Element	Description
rp	A related pattern that is affected and must be modified
link	A link between two intentions that highlights the part of the goal template that must be modified
action	An indicator of what must be done
bst	A new business strategy that represents a new solution
oldbst	An old business strategy that must be eliminated from rp

to an intention which is refined by xp. These two preconditions prevent using the extension algorithm for merely changing an unrelated pattern in pf. Isolated modifications of patterns must use the modification algorithm, which is described in Appendix A.

Step S1 initializes mg with the main goal of xp's goal template. Steps S2 to S4 take the modifications set and invokes the modification algorithm (Algorithm 2) to modify related patterns in pf. This captures the possible effects of adding xp on other patterns in GPF. Next, in step S5 the new pattern, xp, is added to pf. Although this is conceptually a simple insertion of a pattern, in reality a deep copy is taking place in which every element of xp is copied into pf. In step S6 a set of leaf intentions (without any incoming links) of other patterns in pf that are equal to the main goal of xp is assigned to relatedIntentions. This is the set of intentions that are refined by xp. In S7, the set of leaf intentions of xp's goal template is assigned to leafIntentions, whose elements may be refined by other patterns in pf. In step S8, the related intentions are linked to xp by assigning xp

to patternDef of intentions in relatedIntentions. Finally, in step S9, intentions of xp's goal template that can be refined with other patterns in pf are linked to the appropriate pattern by setting their patternDef accordingly.

5 Patient Safety Example

5.1 Family Creation (GDM-FDM)

Healthcare institutes strive to improve the safety of their patients. Yet, every year, thousands of patients suffer from adverse events, which are defined as undesirable outcomes caused by healthcare business processes. Decreasing adverse events by improving these processes forms our patient safety domain. In [17], we showed that goal and business process modeling with URN can be used effectively in this domain in order to capture problems and their solutions. We also discussed how these models were used to create an adverse event management system. While modeling problems and solutions in different departments of a teaching hospital (in Ontario, Canada) for improving patient safety, we observed that some problems and solutions were recurring over the span of the domain. We hence built a pattern-based framework (a GPF with 32 patterns) targeting the documentation and reuse of knowledge about problems and solutions in this particular aspect of patient safety [7].

For example, Increase Patient Safety is an abstract, recurring requirement in different hospital departments and other healthcare organizations. The top of Fig. 3 represents the goal template and two business process templates of this pattern. The goal template (in GRL) shows the contributions of Collect Data, Generate Informative Outcome Information, Make Safety Decision, and Adopt Decision to the realization of Increase Patient Safety altogether with side-effects (e.g., on quality of care) and dependencies (e.g., on required infrastructures). Two strategies have been defined for this pattern. The first one (A) includes only the sub-goals Collect Data and Generate Informative Outcome Information. The second strategy (B) includes also the two other sub-goals, Make Safety Decision and Adopt Decision, and adds corresponding activities to its business process template. UCM models represent these two strategies as business process templates, which describe the ordering of the activities (that are further refined in other patterns of the Goal-driven Pattern Family). All these models together constitute the Increase Patient Safety (p1) pattern. The area within the solid border in the UML object diagram of Fig. 5 represents this pattern in terms of instances of the Family Metamodel described in Fig. 2. Note that this object diagram uses acronyms as identifiers instead of the real object names in order to save space.

5.2 Customization (GDM-OCEM)

The OCEM algorithm [7] uses an organizational goal model for adapting the appropriate goal and business process models of a given Goal-driven Pattern Family to the organization where this GPF is applied. The organizational goal

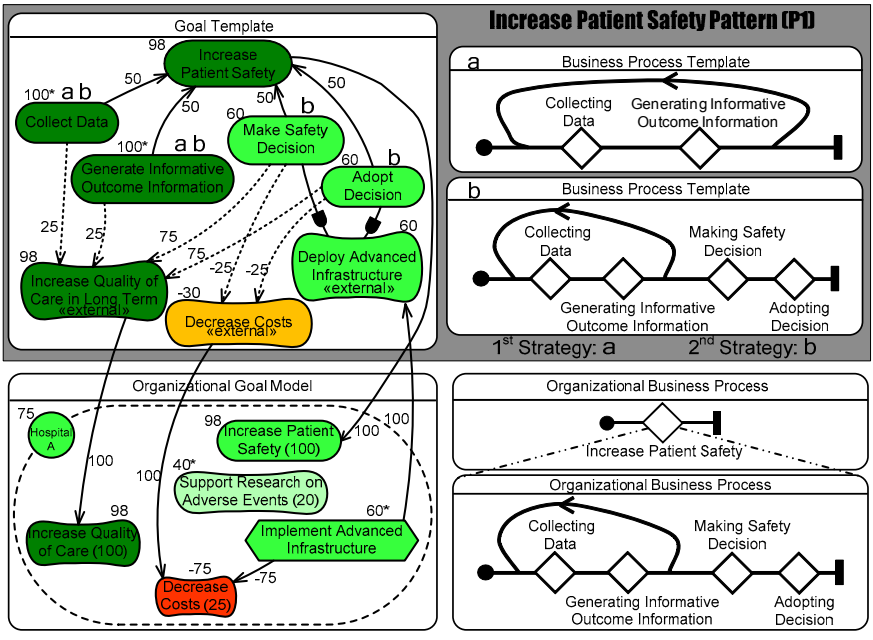


Fig. 3. Applying GoPF: and Overview

model of Hospital A, shown in the bottom half of Fig. 3, identifies the main goal (Increase Patient Safety) and three high-level softgoals related to quality, cost, and research concerns. Importance values are added to some of these intentional elements in the organizational goal model (e.g., the importance of Increase Patient Safety to its containing actor is deemed to be 100 while the importance of Decrease Cost is 25, which means that decreasing cost is less of an issue to Hospital A than increasing care and safety). Furthermore, Fig. 3 depicts the current evaluation strategy of the organization, describing the as-is situation. Initial satisfaction levels, shown by the presence of a star (*), are provided: 60 to the task, indicating that although some advanced infrastructure is available, there is still room for improvement, and 40 to describe the current level of support to research on adverse events at that hospital. Color feedback is provided to show the satisfaction level of the intentions (the greener, the better).

With OCEM, we establish links between the initial organizational goal model and the goal template of the pattern: a contribution with weight 100 is added from the Increase Patient Safety goal in the pattern to the Increase Patient Safety of the organization (showing their equivalence), two contributions with weight 100 are added from the quality/cost softgoals in the pattern to the quality/cost softgoals of the organizational model, and finally a contribution with weight 100 is added from the task in the organizational model to the softgoal with the dependencies in the pattern. Not all intentional elements from the organization goal model need to be linked to an element of the pattern (e.g., Support Research on Adverse Events is not addressed by the current pattern).

Next, all alternative strategies are compared automatically for finding the best solution. Fig. 3 shows the result of the second strategy (B) because it yields the better result than the first strategy (A), given that the organizational goal model places more value on quality than on cost and already has some advanced infrastructure available. A different healthcare institute with more focus on cost than quality and no advanced infrastructure available would see the first strategy (A) win over the second strategy (B). This evaluation is automated with OCEM, as it builds on GRL's quantitative evaluation algorithm, which propagates the known satisfaction levels to other intentional elements in the GRL models through their links.

The goals in the pattern and the links are then added to the organizational goal model, while the business process template related to the chosen strategy is added as a sub-model to the Increase Patient Safety stub (indicated by the long-dash-dot-dotted line). The figure represents the output of application of the OCEM method: a refined GRL model of the organization with a model of the chosen business process options, together with the rationale for their selection. Applying the patterns in the GPF can further refine the four sub-goals and linked stubs of Increase Patient Safety pattern. In order to support satisfactory level of details, the patterns of our patient safety family include ten layers of decomposition (of which only the top one is discussed here).

5.3 Family Evolution (GDM-FDM)

As illustrated in our example, the knowledge in the GPF can be reused for particular organizations interested in patient safety. These patterns focus on the problem of improving safety by highlighting processes that can be improved. However, the patterns in our family do not use the collected data for taking immediate actions to prevent an adverse event for a particular patient. Over time, we observed that some hospitals use such collected data not only for *a posteriori* analysis but also for *preventing* the potential adverse events that may happen. Consequently, a new pattern (xp) is created that captures the problem of taking action to prevent adverse events and its alternative solutions. The recurring excerpt of the observed goal models that formulates Take Action forms the goal template for xp (left side of Fig. 4). In this pattern, we have chosen not to show the side effects to simplify the example. It is composed of the main

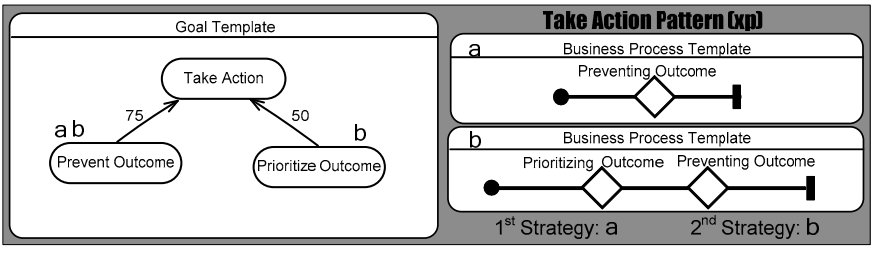


Fig. 4. Take Action Pattern

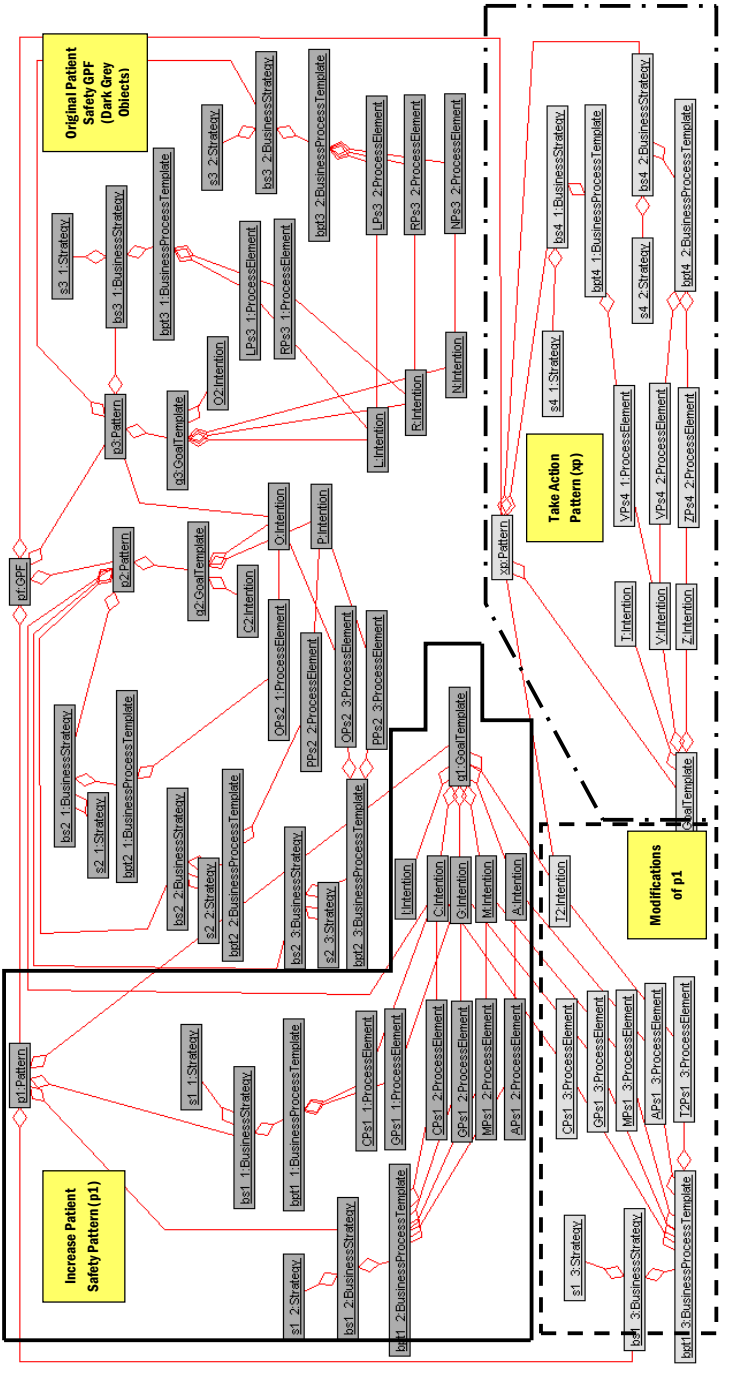


Fig. 5. UML object diagram of the GPF evolved to includes xp

high-level goal, i.e., Take Action, and elements of solution, i.e., Prioritize Outcome and Prevent Outcome. The alternative solutions are captured in the form of business strategies composed of business process templates and strategies. The two business process templates of the new pattern are shown in Fig. 4 (right).

The extension algorithm (Algorithm 1) evolves the patient safety GPF to include this new pattern. In order to evolve the GPF, the analyst prepares and provides inputs: pf as the initial GPF that must be extended (I1), xp as the new pattern (I2), and modifications as a set that represents the needed modifications on other patterns in pf (I3).

The dark grey objects in Fig. 5 illustrate the initial pf as a FMM-based, UML object diagram (containing only 3 out of the 32 patterns, for simplicity). In this example, p1 is the only pattern affected by the extension because Take Action positively contributes to Increases Patient Safety. Therefore modifications is set to {(p1, link_I_T, *Add*, bs1_3, *null*)}, indicating the new goal that must be added to p1's business goal template. It also indicates the new business strategy that represents an alternative solution that must be added to p1.

5.4 Applying the Extension Algorithm

Step S1 initializes mg with T (Take action), which is the main goal of xp. The next three steps (S2, S3, and S4) invoke the modification method with pf and the modifications set as its inputs. The modification algorithm (Appendix A) applies these changes on the related patterns, i.e., p1, as it is the only related pattern in pf. The part of pf in Fig. 5 encapsulated within a dashed box represents the modifications. Step S5 adds xp to pf. This is a deep copy of all elements of xp into pf. After this step, xp (see dash dotted area in Fig. 5) becomes a part of pf but it is still an isolated pattern as the links between xp and related patterns in pf are not yet established. Step S6 finds those intentions that are (i) leaves of other patterns in pf and (ii) equal to the main goal of xp. In this example, T2 (Take Action) is a leaf intention in p1 and is equal to mg. Therefore, relatedIntentions is set to {T2}. Step S7 assigns the leaves of xp ({V, Z}) to leafIntentions. Steps S8 and S8.1 set the patternDef of T2, which is the only member of relatedIntentions, to xp. This captures the fact that xp is refining the T2 intention in the goal template of p1. Steps S9, S9.1 are applied for each element of leafIntentions.

However because no pattern in pf refines either V or Z, no action takes place in 9.1.1 , 9.1.2 , or 9.1.3. The whole Fig. 5 shows the output of our method (O1). It represents the extended GPF where the new pattern xp was added and integrated with the patterns in the initial GPF (pf).

6 Related Work

Iida described an early attempt at capturing process elements with patterns, for the software development domain [18]. Through transformations applied to a primitive process, customization to a particular organization becomes possible, which is similar in spirit to our OCEM method. However, their approach considers only roles, products and activities (process definitions at the level of UCM),

and not goals of the patterns or of the organization. The selection of patterns to apply during transformations is hence done in an ad hoc way. In addition, there is no mechanism in place to evolve the patterns themselves. In his thesis [19], Tran reviewed many pattern-based process modeling approaches that suffer from the same weaknesses. His approach however formalizes the process patterns with a metamodel and provides algorithms to use them successively on processes for their evolution. The technique shares some common objectives with our own, except that we focus on the evolution of patterns and we also consider goals.

Lapouchnian *et al.* [20] present a requirement-driven approach to address the problems caused by the gap between goals and business processes. The authors provide annotations that enrich goal models to capture information about business processes and ordering. By preserving the variability in the executable business processes that are generated from enriched goal models, stakeholders can change the behavior of the application through their high-level goals. However, combining goal models and business processes in such a way makes the models specific to one organization, which hurts reusability. Furthermore, this approach does not provide a formal mechanism for accommodating the changes that happen in organizations. In our approach, we provide a framework that keeps goal and process models separate while traceability links between these models are created and preserved in the patterns. The patterns make it possible to reuse solutions in contexts where goals and the forces at play are compatible. Keeping goal and business process templates separate also enables the use of formal evolution mechanisms that accommodate changes occurring in the domain.

Zhao *et al.* suggested an approach for the evolution of pattern-based designs [21] and of design patterns [22]. They propose a graph transformation method at the pattern level for evolving and validating patterns and pattern-based designs. Likewise, Dong *et al.* [23] proposed a transformation based on two levels (primitive and pattern) to formulate the evolution process of design patterns. However, these approaches are limited because (i) they are focused on design patterns and are mostly fine tuned toward evolving UML class diagrams, and (ii) evolution is limited to variations of the initial pattern, i.e., the evolved pattern must be reducible to the initial graph. For instance, an *abstract factory* pattern can evolve to a new variation of the abstract factory pattern, which must be based on the principles of abstract factory patterns. In this paper, we consider evolutions of the patterns at a more abstract level that captures the knowledge about the goals and requirements of stakeholders. Furthermore, patterns can evolve beyond their initial versions, without a need for some equivalence.

7 Conclusions and Future Work

Changes in stakeholder requirements are unavoidable. One benefit of patterns is that they encapsulate recurring problems and solutions into modules. This is valuable because patterns are less subject to changes than software in general, and hence organizations that use the proposed framework become more robust to changes. However, rapid pace of changes in technology, business environments,

and concerns of stakeholder has highlighted the need for evolving pattern-based frameworks in order to accurately reflect current domain knowledge.

In this paper, we presented GoPF, a goal-oriented pattern-based framework that formalizes families of patterns (GPFs) and provides mechanisms for evolving them to reflect the changes in a particular domain. The Framework Metamodel (FMM) lays down the foundation of GPFs and formalizes the patterns that capture the knowledge about business goals and processes. FMM is formalized as a profile of the standard URN modeling notation, which combines goals, processes, and links between them. Patterns are captured as goal templates and business strategies, which enable the selection of appropriate solutions in the context of a particular organization (e.g., with the OCEM method). We equipped GoPF with a new evolution mechanism that extends GPFs. Our extension algorithm provides the steps for integrating a new pattern with a given GPF. The result is a GPF that includes the new pattern that provides solutions to a particular problem, refines related patterns, and may be refined by related patterns already in the GPF. We illustrated this algorithm with the help of a real-life example, i.e., the evolution of a patient safety GPF with a new pattern. To our knowledge, this is the first attempt at describing and formalizing an evolution method for patterns that integrate business goals and processes.

For future work, we need to take better advantage of the commonalities found among our extension, modification, elimination, and combination algorithms. We already invoke *modification* from within *extension*, but we also suspect that the Family Creation method is in fact a special case of our extension algorithm (where we start from an empty GPF). These algorithms can also benefit from automation (e.g., in our jUCMNav tool). One challenge here will be to make the creation of the modifications set (required by Algorithm 1) simple for modelers. Also, since *key performance indicators* were added to URN by Pourshahid *et al.* [24] as another type of intentional element, they could be included in GPF descriptions as common indicators used to measure the goals described in the patterns. This could improve reuse in the context of performance management of business processes such as the palliative care processes recently documented by Kuziemsky *et al.* [25], which include indicators. Finally, we plan to research opportunities, on the pattern selection side (i.e., OCEM), of using a constraint-oriented solving approach that will find a "globally optimal" process in the context of the organization based on the various strategies.

Acknowledgments. This research was supported by the NSERC/CIHR Collaborative Health Research Program (Canada).

References

1. Alencar, F., Marín, B., Giachetti, G., Pastor, O., Castro, J., Pimentel, J.H.: From i* Requirements Models to Conceptual Models of a Model Driven Development Process. In: Persson, A., Stirna, J. (eds.) PoEM 2009. LNBIP, vol. 39, pp. 99–114. Springer, Heidelberg (2009)

2. Berg, M.: Implementing information systems in health care organizations: myths and challenges. Int. J. of Med Info 64, 143–156 (2001)
3. Ostadzadeh, S.S., Aliee, F.S., Ostadzadeh, S.A.: An MDA-Based Generic Framework to Address Various Aspects of Enterprise Architecture. In: Sobh, T. (ed.) Advances in Computer and Information Sciences and Eng., pp. 455–460 (2003)
4. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, Reading (1995)
5. Čiukšys, D., Čaplinskas, A.: Ontology-based approach to reuse of business process knowledge. Informacijos Mokslai 42–43, 168–174 (2007)
6. URN Virtual Library, http://www.usecasemaps.org/urn/ (2010)
7. Behnam, S.A., Amyot, D., Mussbacher, G.: Towards a Pattern-Based Framework for Goal-Driven Business Process Modeling. In: 8th Int. Conf. on Software Eng. Research, Management and Applications (SERA 2010), pp. 137–145. IEEE CS, Los Alamitos (2010)
8. ITU-T – International Telecommunications Union: Recommendation Z.151 (11/08) User Requirements Notation (URN) – Language definition, Switzerland (2008)
9. Weiss, M., Amyot, D.: Business process modeling with URN. Int. J. of E-Business Research 1, 63–90 (2005)
10. jUCMNav, v. 4.3.0 (2010), http://jucmnav.softwareengineering.ca/jucmnav/
11. Amyot, D., Horkoff, J., Gross, D., Mussbacher, G.: A Lightweight GRL Profile for i* Modeling. In: Heuser, C.A., Pernul, G. (eds.) ER 2009. LNCS, vol. 5833, pp. 254–264. Springer, Heidelberg (2009)
12. Alexander, C., Ishikawa, S., Silverstein, M.: A Pattern Language: Towns, Buildings, Construction. Oxford University Press, US (1977)
13. Hoffman, T.: Study: 85% of IT departments fail to meet biz needs. Computer World 11 (October 1999)
14. Fowler, M.: Analysis Patterns: reusable object models. Addison-Wesley, Reading (2000)
15. Buschmann, F.: Pattern-Oriented Software Architecture: A System of Patterns. Wiley, Chichester (2002)
16. Mussbacher, G., Amyot, D., Weiss, M.: Formalizing Patterns with the User Requirements Notation. Design Pattern Formalization Techniques, IGI Global, 302–322 (2007)
17. Behnam, S.A., Amyot, D., Forster, A.J., Peyton, L., Shamsaei, A.: Goal-Driven Development of a Patient Surveillance Application for Improving Patient Safety. In: Babin, G., Kropf, P., Weiss, M. (eds.) MCETECH 2009. LNBIP, vol. 26, pp. 65–76. Springer, Heidelberg (2009)
18. Iida, H.: Pattern-Oriented Approach to Software Process Evolution. In: Int. Workshop on the Principles of Software Evolution, pp. 55–59 (1999)
19. Tran, H.: Modélisation de Procédés Logiciels à Base de Patrons Réutilisables. Thése de doctorat, Université de Toulouse-le-Mirail, France (November 2007)
20. Lapouchnian, A., Yu, Y., Mylopoulos, J.: Requirements-Driven Design and Configuration Management of Business Processes. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714, pp. 246–261. Springer, Heidelberg (2007)
21. Zhao, C., Kong, J., Dong, J., Zhang, K.: Pattern-based design evolution using graph transformation. J. of Visual Languages and Computing 18, 378–398 (2007)
22. Zhao, C., Kong, J., Zhang, K.: Design pattern evolution and verification using graph transformation. In: 40th Annual HICSS, p. 290a. IEEE CS, Los Alamitos (2007)
23. Dong, J., Zhao, Y., Sun, Y.: Design pattern evolutions in QVT. Software Quality Journal 18, 269–297 (2010)

24. Pourshahid, A., Chen, P., Amyot, D., Forster, A.J., Ghanavati, S., Peyton, L., Weiss, M.: Business Process Management with the User Requirements Notation. Electronic Commerce Research 9(4), 269–316 (2009)
25. Kuziemsky, C., Liu, X., Peyton, L.: Leveraging Goal Models and Performance Indicators to Assess Health Care Information Systems. In: QUATIC 2010, pp. 222–227. IEEE CS, Los Alamitos (2010)

Appendix A: Modification Algorithm

This appendix describes the algorithm for evolving a GPF by *modifying* its patterns. Such evolution is needed when i) changes in a domain indicate that the goal template or business strategy of a pattern must be updated, ii) the GPF is being extended with a new pattern that affects a particular pattern (i.e., the way it is used within Algorithm 1), and iii) another pattern in the GPF is eliminated and a particular pattern is affected.

A modification is composed of three main steps (see Algorithm 2): first, it modifies the goal template of a pattern, then it removes the old business strategy (if available), and, finally, a new business strategy is added (if available).

```
Inputs of the modification algorithm
I1.  pf:GPF     /* initial pattern family */
I2.  modifications: set of (p: Pattern, link:ElementLink, action ∈ {Add,Delete},
                            bst:BusinessStrategy , oldbst:BusinessStrategy)
                    where link.toLinks.isEqualTo(p.mainGoal())

Output of the modification algorithm
O1.  modified pf:GPF      /* the modified pattern family */

Steps of the modification algorithm
for each m in modifications:
S1.  mp:Pattern = a pattern ∈ pf.patternCollection() where pattern.isEqualTo (m.p)
          /*m.p is the first element of the m */
S2.  if (m.action == Delete) then
S3.       mp.GT.delete(m.link)
S4.  elseif (m.action == Add) then
S5.       mp.GT.add(m.link)
S6.  endif   /* if no action is provided then the GoalTemplate is unchanged */
S7.  if (m.oldbst ≠ null) then
S8.       pbst:BusinessStrategy = a businessStrategy ∈ mp.businessStrategyCollection()
                                  where businessStrategy.isEqualTo(m.oldbst)
S9.       mp.delete(pbst)
S10. endif
S11. If (m.bst ≠ null) then
S12.      mp.add(m.bst)
S13. endif
```

Algorithm 2. Modification of GPF

All the steps of this modification algorithm are taken for every element of the modifications set. Step S1 initializes mp with the pattern of the active member of the modifications set (m). Next, in steps S2 to S6, depending on the type of action, the link will be either added to or deleted from mp's goal template. Then, if m.oldbst is not null, it will be added to mp in steps S7 to S10. Finally, if m.bst contains a business strategy, then it is added to mp (steps S11 to S13).

Searching, Translating and Classifying Information in Cyberspace

Jacques Savoy, Ljiljana Dolamic, and Olena Zubaryeva

Computer Science Department, University of Neuchatel,
Rue Emile Argand 11, 2000 Neuchâtel, Switzerland
{Jacques.Savoy,Ljiljana.Dolamic,Olena.Zubaryeva}@unine.ch

Abstract. In this paper we describe current search technologies available on the web, explain underlying difficulties and show their limits, related to either current technologies or to the intrinsic properties of all natural languages. We then analyze the effectiveness of freely available machine translation services and demonstrate that under certain conditions these translation systems can operate at the same performance levels as manual translators. Searching for factual information with commercial search engines also allows the retrieval of facts, user comments and opinions on target items. In the third part we explain how the principle machine learning strategies are able to classify short passages of text extracted from the blogosphere as factual or opinionated and then classify their polarity (positive, negative or mixed).

Keywords: Search technology, web, machine translation, automatic text classification, machine learning, natural language processing (NLP).

1 Introduction

We have witnessed the apparition of the web over the previous decade, but during the next we will most certainly be able to confirm the maturity, diversity and necessity of the web as a communication and processing medium. In this paper, we briefly describe and evaluate the quality of three important services being applied on the web: search technologies [1], [2], machine translation [3] and automatic classification systems [4]. Given their affiliation with natural language processing (NLP) [5], they play a key role in the web's current development and in the near future they will lead to the development of new web-based services.

First, it is through the help of search engines that the web has grown to its current size. Without these systems it would be nearly impossible to search for specific information (e.g, who was the first man in space), retrieve named pages or services (e.g., passport renewal), and find the wished web sites designed for buying goods or services (e.g., hotel Brussels). When asked for their opinion, users seem relatively satisfied with commercial search engines, but the questions we would like to answer in this domain is whether or not current search technologies are still progressing and what are the foreseen limits. Section 2 will provide some answers to these points.

G. Babin, K. Stanoevska-Slabeva, P. Kropf (Eds.): MCETECH 2011, LNBIP 78, pp. 62–75, 2011.

Second, although in its early years the web was dominated by one language, this monolingual aspect is no more the norm. Many users would now like to communicate in a variety of languages (e.g., in multilingual countries such as Canada, Switzerland, or international companies, etc.) and to do so they need to overcome certain language barriers. With current search engines for example, they may wish write a request in one language and retrieve documents written in others. Conversely, while some users can read documents in another language, they would not be able to formulate a query in that language or provide the reliable search terms needed to retrieve the documents being searched (or for retrieving images, music, video, or statistical tables for which language is less important). In other circumstances, monolingual users may want to retrieve documents in another language and then automatically translate the retrieved texts into their own language (before manually translating the most important information or confirming their own understanding of a given web page). What about the current quality of these freely available translation services? What important aspects still need some improvement? These questions will be addressed in Section 3.

Third, when handling and processing huge amounts of information, we need efficient methods for classifying it into predefined categories. Simply searching for factual (or objective) information on a given item or issue is not always the final objective, and web surfers may want to know more about the opinions, feelings or comments other users might have. Given the current growth in activities in the search community, especially in adding content to blogs, online forums, Internet platforms, etc., the task of detecting and classifying information has become increasingly popular. It is consequently more important that a system be developed to process the multitude languages in use and also detect any subjective expressions [6], [7]. The current technology and underlying difficulties in this domain will be presented in Section 4.

Separately, the search engine technology, the machine translation service and the opinion detection and classification system are useful *per se*. In conjunction they allow the user to discover additional services or products. Moreover, the user has now the opportunity to know previous experiences done by other people on the target service or product. Imagine the following scenario.

You travel to Berlin and you need to book a hotel. Using a commercial search engine, you can find various hotels in Berlin, and several of them seem to fit your criteria (location, price, comfort, etc). However you don't have a clear indication about the noise in the room because you really want a quiet room. It is not clear if the location itself is noisy or not. Exploring the blogs without a search engine is not possible, and many comments are written in German, French, Italian, Dutch or Spanish languages that you cannot understand. Using a new search engine combining these three technologies, you will be able to discover comments written about the selected hotel in Berlin. To achieve this, you write a request and if needed, your query will be translated into other languages to obtain a broader coverage. This new search system will also classify the retrieved comments according to their polarity, namely positive, negative or mixed. Since

many of them are written in another languages than English, you can just click on them to obtain an English translation.

2 Search Technologies

During the last decade, the information retrieval (IR) domain [1], [2], has been confronted with larger volumes of information from which pertinent items (text, web pages, images, video, music) must be retrieved, when responding to user requests (*ad hoc tasks*). Given that web users tend to submit short requests composed of only one or two words [8], effective IR models are being proposed to meet this challenge. Moreover, users expect to find one or more appropriate answers at the very top of the retrieved items listed, and within minimal waiting periods (less than 0.2 sec).

For this reason commercial search engines (mainly Google, Yahoo!, and Bing) have based their technology on matching keywords (the same or very similar terms must appear in the query and in the retrieved web pages). In addition to this initial evidence on their relevance, search engines must also consider information on the links pointing to these pages. Current practices therefore accounting for the number of incoming links to a page, thus helping to define the page's merit (or usefulness) or the probability of finding the desired information during a random walk (e.g., PageRank algorithm [9]), or other variants on such link-based popularity measures [10]. The page's hyperlink structure is also useful to obtain short descriptions of the target page by inspecting the anchor texts. For example, these sequences may include " Micro$oft " or " the Big Empire ", as well as descriptors "Micro$oft" and "the Big Empire" linking to the target page *Microsoft.com*. This short text-based evidence helps in describing pages containing various formulations other than those provided by the target page's author. Moreover, these anchor texts are very useful in obtaining textual descriptions of various other media types including images, graphics, photos, video, music, etc.

Current search engines also take user information into account by providing them with more personalized answers, adapted to their personal interests or other personal data. They may for example, combine the search request "Movie" and the user's IP number, thus allowing the search engine to provide a list of films showing the same day in theatres located within the user's geographical proximity. Final search results could also be influenced by a user's previous queries (or previous search sessions) processed by the search engine (thematic context). Moreover, in addition to considering user location or country (the US, for example), the search engine could crosscheck user location with data available in a national or regional census database, and then detect useful demographic information related to the searcher, such as age, race, revenue, etc. Finally, through considering click-through rates, search engines may be able to adapt their answers to this and other forms of user feedback. When users tend to select the same retrieved item for the same query for example, this target page will obtain better rankings in the future.

In order to develop an overview of the effectiveness of current search technologies, we reviewed various papers published in three well-known evaluation campaigns, including TREC [11] (`trec.nist.gov`), NTCIR (`ntcir.nii.ac.jp`) or CLEF (`www.clef-campaign.org`). Based on their analysis of the current state of the art in IR technology, Armstrong *et al.* [12] demonstrate that the retrieval quality of the new search strategies presented in international conferences (ACM-SIGIR or ACM-CIKM) has not significantly improved since 1998. This research seems in fact to have reached a plateau, thus limiting expectations of better search quality in the near future. In another study evaluating IR systems, Hawking & Robertson [13] demonstrate that the greater volume of information available will render the search task easier, at least in their ability to retrieve one or only a few pertinent documents. Current commercial search engines are working on this context, with the goal being to provide the typical user with a single appropriate answer. So when searching in larger volume, the search task will be simpler. Future improvements may in fact concern the processing of higher volumes, not really the quality of the search *per se*. Although at present users are generally satisfied with commercial search engines, they do complain about the poor quality in terms of interfaces and results when using dedicated search engine working within a single web site [14].

The question that then arises is whether or not current search technology is perfect, or to understand when it fails to provide the correct result. To come up with an answer we analyzed answers provided by the best IR systems, based on a set of 160 topics submitted during three CLEF evaluation campaigns. We thus found three main categories of failure: 1) IR process flaws, 2) natural language defects, and 3) incomplete and imprecise user topic formulations.

Among the processing faults found in certain search engines, we discovered problems related to stopword removal and letter normalization (conversion of uppercase letters to lowercase). Stopword lists are used to eliminate frequently occurring common words (*the*, *of*, *a*, *is*, *us*, or *it*) having no specific meaning and possibly generating noise during the IR process (e.g., pages are retrieved because they share the terms *the* and *are* with the query). On the other hand when processing requests including phrases such as *IT engineer*, *vitamin A* or *US citizen*, the systems should not remove the forms *it*, *a* or *us*.

To increase the likelihood of obtaining a match between query terms and web pages, search systems apply a stemming procedure to conflate word variants into a common stem. For example, when a topic includes the word *computer*, it seems reasonable to also retrieve documents containing the word *computers*, as well as morphologically related terms such as *computational* and *computing*. Included in our set of requests for example is the topic "Elections parlementaires européennes" (European Parliament Elections) for which relevant documents found had *élections* and *européennes* or *Europe* in common with the query. Those documents have the noun *parlement* instead of the adjective form *parlementaire* as expressed in the topic. Although the search system was able to conflate the form *europe* and *européennes* under the same form, it was not able to establish a link between the terms *parlement* and *parlementaire*, and thus missed many relevant pages.

As with all natural language processing systems, users expect a certain degree of robustness. Thus, when a query contains a spelling error, the search engine should either suggest a corrected version or try to provide a reasonable answer. In the request "Innondationeurs en Hollande et en Allemagne" (Flooding in Holland and Germany) for example, the system should preferably suggest the term *Innondations* for the incorrect spelling *Innondationeurs*, rather than limiting the query to its second part (Holland and Germany). Processing this type of situation becomes less clear when handling spelling variants (color vs. colour), particularly proper names (Gorbachev vs. Gorbachov, Oscar vs. Oskar), and when both variants are present in many pages.

The second main category of search engine failures involves different problems related to natural language expressions. Among these are similar expressions conveying the same idea or concept using different words (synonyms) or formulations. In our topics, we found this problem with the nouns *film* and *movie*, or *car* and *automobile*. In such cases when the query uses one form and the document the other, there would not be a match and the corresponding document is not retreived. Different regions or countries may employ different formulations to refer to the same object. An example would be the topic "risques du téléphone portable" (risks with mobile phones), in which the relevant documents contain country dependant synonyms. In Switzerland for example, a portable phone is usually a *natel*, in Belgium *téléphone mobile, portable* in France and *cellulaire* in Québec. Among the top ten documents retrieved by the IR system, one can find documents written in France (by using the formulation *téléphone portable*) and some documents about the risk of being in the mountains. Other retrieved articles may simply concern certain aspects of mobile phones (new joint ventures, new products, etc.).

The second main problem related to natural languages is polysemy when certain words or expressions may have several interpretations or meanings. The noun *bank* for example may be related to a financial institution or a river. Thus when one form appears in a user request and the other in relevant documents, there would not be a proper match, and not all pertinent information would be retrieved. Other examples include the name *Jaguar*, which refers to an animal, a car or a software package; while *Java* refers to an island, a coffee, a dance or a language. The use of short acronyms is also the source of many incorrect interpretations (e.g., BSE could be Bombay, Beirut, Bahrain, ... Stock Exchange, Bovine Spongiform Encephalopathy, Basic Set Element, Breast Self-Exam, etc.).

A third main source of search system mismatch occurs when user requests contain expressions that are too broad or too imprecise. To illustrate this problem we analyzed the requests for "Trade Unions in Europe," "Edouard Balladur" or "World Soccer Championship". The top ranking documents retrieved by the IR system in all cases had not one but at least two or three terms in common with the query. Are those perfect matches? The users judged these pages to be irrelevant because the real intent behind the topics was "the role and importance of trade unions between European countries," "the economics policies of E. Balladur" or "the result of the final". Another case involving less evident examples

was the query "AI in Latin America", in which the IR process must have inferred that the acronym AI represents *Amnesty International* and not *Artificial Intelligence* (a polysemy problem), and in a second step that *Latin America* refers to a region containing several countries. Relevant documents would cite a country (e.g., Mexico or Colombia) without explicitly linking the country name to the continent or region name. The request "Wonders of Ancient World" caused the same problem in that relevant pages did not include the expression "Wonders of Ancient World" when describing the pyramids in Egypt.

3 Machine Translation

Retrieving the correct information is the first step, and then we need to provide it into the appropriate language. In fact we must recognize that we are living in a multilingual world [15], and given recent developments on the web, language (and script) diversity is steadily increasing. Based on freely available translation services, we may hope to cross easily these various language barriers, facilitating communication among people speaking different languages. In Europe and India for example, and also in large international organizations (e.g., WTO, Nestlé), multilingual communication is a real concern. In the European Union for example there are 23 official languages that requires 23 x 22 = 506 translation pairs. While English is not the language spoken by the majority of people around the world, it certainly plays a central role as an *interlingua* medium in transmitting knowledge or expressing opinions. The CNN success story is just one example of the increasing importance of this language, yet the Al Jazeera network certainly confirms that other languages will certainly be of greater importance in a near future. The fact remains however that English is often the first foreign language learned in Europe, India and in Far-East countries, and thus it is still very important to provide adequate resources for translating from other languages to English and vice-versa.

The major commercial search engines have certainly not ignored this demand for translation resources to and from English, and Google is certainly a case in point, given its efforts in improve searching in pages available in English as well as other languages on the web. Regardless of language in which queries are written, Google has launched a translation service providing two-way online translation services, mainly between English and more than 55 other languages (`translate.google.com`). Over the last few years other free Internet translation services have also been made available, including Yahoo! (`babelfish.yahoo.com`) and Reverso (`www.reverso.net`). Behind these web services is a translation strategy based on statistical machine translation approaches [3], [5], where the most probable translation is selected according to occurrence frequencies in parallel corpora, and able to make adjustments based on user feedback.

While translations from/to English are freely available, the resulting translated document may not be of the highest quality. We know in fact that both automatic and human translations can sometimes be very poor (see examples in [16]). Quality translation may depend on the relationship between the source

and target languages, particularly those having a close relationship with English (e.g., French, German) as opposed to more distant languages such as Japanese. Although we do not intend to evaluate translations *per se*, we will analyze various IR and translation systems in terms of their ability to retrieve items written in English, based on the automatic translation of queries written in German, French, Spanish, Chinese and Japanese languages. A previous partial study can be found in [17].

Our evaluation is based mainly on 310 CLEF topics reflecting typical web search requests, and consisting of two or three words in mean. These topics cover various subjects such as "El Niño and the Weather," "Chinese Currency Devaluation," "Eurofighter," "Victories of Alberto Tomba," "Computer Animation," "Films Set in Scotland," "Oil Prices," or "Sex in Advertisements." This topic set is available in English, German (DE), French (FR), Spanish (SP), Chinese (ZH) and Japanese (JA).

To evaluate the quality of Google and Yahoo! translation services, we performed a search using query formulations written in English to define the baseline (monolingual search performance at 100%). Then using the topics available in the various other languages, we first automatically translated them using either Google or Yahoo!, and carried out the search with the translated topics (see Table 1 for performance differences). As expected, lower performance levels were obtained for all query translations other than English. The best level was for the Spanish formulation translated by Google, in which case the decrease was around 1.4%, relative to the baseline. Compared to the monolingual search, this level was considered non-significant by a statistical test (paired t-test, significant level at 5%), while all others were viewed as significant.

Table 1 clearly indicates that Google's translations were better than those achieved by Yahoo! (at least with our search process). Upon inspecting the results for the different languages, the translation process seems easier from French, German or Spanish languages than from Chinese, a more remote language, while in certain cases (Japanese), the differences were smaller than expected. Finally, based on the absolute values, we may conclude that automatic query translation is a feasible alternative. The performance decreases due to translation are not that large, around 6% for German and French, 8% for Japanese and a little more for Chinese (14%). How can we obtain better automatic translation and where are the real difficulties?

The first source of translation difficulties was the presence of proper names in the topic. Although in certain cases names did not change from one language to

Table 1. Comparative performances of queries translated into English using machine translation systems and manual methods (in % compared to monolingual search)

	From DE	From FR	From SP	From ZH	From JA
Using Google	93.6%	93.1%	**98.6%**	85.2%	91.7%
Using Yahoo!	75.3%	83.4%	73.7%	62.7%	72.8%

English (e.g., *France* or *Haiti*), some modifications usually had to be made (e.g., *London* is written *Londres* in French). The same problem seemed to appear for other topics, such as in "Return of Solzhenitsyn" which was written as "le retour de Soljénitsyne" in French, "Retorno de Solzhenitsin" in Spanish, or "Rückkehr Solschenizyns" in German. In this case, when French or German was the query language, Yahoo!'s translation system was not able to provide the correct English spelling for this name.

The correct translation of a proper name was found to be more difficult when it had a specific meaning in the source language. In the query "Schneider Bankruptcy" for example, because the name *Schneider* also means *cutter* in German, and this meaning was selected by Yahoo!'s translation system, with the phrase being translated as "Cutter bankruptcy." Another and more difficult example occurred with the topic "El Niño and the Weather," where the weather phenomenon in Spanish was designated as the noun *the boy*, and thus the Yahoo! translation returned "the boy and the time," ignoring the fact that the topic contained a specific noun. When Chinese was the query language, both Google and Yahoo! were not able to translate the proper name, leaving the Chinese word untouched or providing a weird expression (e.g., "Schneider Bankruptcy" became "Shi Tejia goes bankrupt" with Yahoo!).

A second main source of translation error was semantics, especially polysemy, meaning that a source language term can be translated into several other terms in the target language. More precisely, in order to find the appropriate word (or expression) in English, the translation system had to take the context into account. As shown previously for example in the Spanish topic "El Niño y el tiempo", the word "tiempo" could be translated as "weather" or "time", the Yahoo! system selected the latter. For the query "Theft of 'The Scream'" written in French as "Vol" du 'Cri'", the French word vol could be translated by *flight* or *theft*, and the translation produced by Google was "The Flight of the 'Scream'" and by Yahoo! was "Flight of the 'Cry'."

This latter translation demonstrates another problem related to synonymy, wherein the translation system had to handle various meanings for a given translation, such as the French word *cri* could be translated as either *scream* or *cry*. This synonymy aspect was also found in various requests involving the related terms *car* and *automobile*. In the original English version the topics, the translation service provided the term *car* more frequently (five times to be precise) while it never returned the term *automobile*. Moreover, the semantic relationship between two (or more) alternatives is not always that close, as illustrated by the query "Ship Collisions", which for the Spanish, Yahoo! returned the translation "Naval collisions."

We found a third translation difference within the English topics where various forms of a given root (*merger*, *merge* or *merging*) representing morphological and grammatical categories. For the original request "Merger of Japanese Banks", for example, the system ranked the first relevant item in the top position, yet with the translation of "Merging of Japanese Banks" the first relevant article appeared in the 6th position. The same problem occurred again in the query

"Golden Globes 1994" where the retrieval system returned a relevant document in the first position, while for the translated query "Gold Globes 1994" the first relevant item only appeared in the 6th position. In this case, with the form *golden*, the IR system was able to rank a relevant item in the first position, but not with *gold*. In our previous example, the classical English stemmer was not able to conflate the forms *merger* and *merging* into the same root (*merging* is transformed into the stem *merg* while *merger* is left untouched).

A fourth main source of translation problems was that compound constructions, such as those occurring frequently in the German language, were not always translated into English, and thus the system simply returned the German term. With the topic "Shark Attacks" for example Google returned the German term *Haifischangriffe* while for the query term "Bronchial asthma" Yahoo! returned *Bronchialasthma*. In both cases, the translated query performs very bad.

4 Automatic Classification in the Blogosphere

As described in Section 2, search technologies are not always perfect, although with commercial search engines users can normally find the information they are seeking. As explained in the previous section, if needed they can resort to machine translation to find the desired information in a given language (although the best performance is obtained when translating from/to English). The role played by web users has changed during the last years however; to the point they are no longer simple information consumers. They may in fact produce information, share their knowledge within Wikipedia-like services, comment on the news, write their own blogs or even their own web sites to express opinions, feelings or impressions on the latest events (politics, exhibits, sports, etc.), or even products. Now, given the expanding number of contact opportunities available in cyberspace through several social networking sites, these exchanges are becoming very common.

With blogs and social networking sites in particular, the nature of information exchanged is becoming *personal*, *subjective* and *opinionated* rather than factual, objective and neutral. Thus when we are looking for a product or service, commercial search engines provide factual information (price, technical data, sale conditions, schedule, etc.), but the value of comments and experiences obtained from previous customers is sometimes considered more valuable than objective information. Opinionated comments are not only related to physical products (books, computers, mobile phones, etc.) but may also include services (hotel, air companies, movies, etc.) and well-known personalities (actors, politicians [18], etc.). Given this variety of information available, retrieving of opinionated web pages or passages is far more complex.

Not only do search engines have to retrieve pertinent information items, they also have to filter out any underlying garbage. Within the blogosphere in particular, we need to assume that data might contain spam or other junk material, including lies or even propaganda written by malicious people or by persons paid to include positive (as well as negative) comments on a target topic. Second, the

system must identify the most pertinent passages, especially those containing comments about the targeted product, while discarding any irrelevant segments (within a blog post, or a web page). Third, the system must detect opinionated passages (binary classification between factual or opinionated category) and even classify the opinion types found within them. Detecting opinionated sentence is usually the first step however as they then have to be classified according to their polarity (positive, negative or mixed) [6], [7].

Gathering information on the users' opinions on a given topic is of interest to individuals (market benchmarking, information search, curiosity), but even more to corporations (marketing, advertisement, market analysis, competitor surveys, detecting new opportunities, etc.) and governments (monitoring, intelligence gathering, protection of citizens). These classification tasks are more complex than they appear, for several reasons, and correct attribution cannot always be done with absolute certainty. In this case, we want to known or indentify the real author. Is this a real customer? Do customer really have any experience with the target items or are they simply reporting known rumor?

The filtering of pertinent opinions can be rendered even more difficult by noisy data (e.g., spelling errors, incomplete sentences) or divergence from standard writing. Latter, different internet-based writing situations (e.g., e-mail, chat groups, instant messaging, blogs, virtual worlds, etc.) may generate their own literary register due to the specific graphical, orthographical, lexical, grammatical and structural features they employ [19]. Examples of these might include the presence of smileys (e.g., `:-)`), the use of commonly occurring abbreviations such as those used in text messaging (e.g., *irl* for *in real life*), as well as certain graphical conventions (e.g., the use of uppercase letters, spaces or stars such as "I SAID NO!"for highlighting reasons), together with the various colors and animations used for display purposes in documents. Finally, search system accuracy could be lower than expected due to a small number of examples from which the system is able to learn.

The previously described problems can be illustrated by an example. To find customer reactions and comments on a given product, we need to discriminate between factual information ("the price of the new iPhone is $800") and opinionated passages (e.g. "the screen design of the iPhone is terrific, not the microphone"). An automatic classification system would have to determine whether this last sentence expressed an opinion concerning a given component (the *screen* and the *microphone*) of a given product (*new iPhone*). And then we might be asked to determine the opinion's polarity (positive for the screen, negative for the micro). Moreover, the intensity and polarity may change radically when another element is added (e.g. "the iPhone is beautiful," "the iPhone is very beautiful,"and "it's *not* very beautiful"). To be useful, an automated system must learn the target of a given negative clause, sometimes by resolving anaphora (e.g., what object is replaced by the pronoun *it*). The target item could of course be more difficult to identify in other cases ("the iPhone clone is really good").

To evaluate the current technologies available in this domain, various international evaluation campaigns have been conducted (see the blog track at TREC or

the MOAT track at NTCIR [20]). In text categorization tasks [4], the various text forms (e.g., clause, sentence, passage, paragraph, document) must be represented by a numerical vector comprising useful and relevant features. Throughout this process we would also need to extract and select those features most useful in identifying the style differences found within the categories. In a second stage, we would weight them according to their importance in the underlying textual representations and also their discriminative power. Finally, through applying classification rules or a learning scheme, the system has to decide whether or not to assign the new input text to a given category.

From among all these possible features, the objective is to select those that are topic-independent but would closely reflect the style of the corresponding text-genre. To achieve this goal, three main sources can be identified. First, at the lexical level we could consider word occurrence frequency (or character n-grams), *hapax legomena* (word occurring once), vocabulary richness [21], total number of tokens, average word length, number of characters, letter occurrence frequency, along with other symbols, etc. Special attention should also be paid to the use of function words (e.g., determinants, prepositions, conjunctions, pronouns such as *an*, *the*, *in*, *or*, *we*, etc.) together with certain verbal forms (*is*, *has*, *will*). Although the precise composition of these word lists is questionable, different authors have suggested a wide variety of lists [22], [23].

Secondly we can also take account for syntactic information as, for example, the part-of-speech (POS) tags by measuring distribution, frequency, patterns or various combinations of these items. Thirdly, some authors [24], [25] have also suggested analyzing structural and layout features, including the total number of lines, number of lines per paragraph, number of tokens per paragraph, presence of greetings or particular signature formats, as well as features derived from html tags. More generally, when classifying passages according to predefined categories, a fourth feature should be considered, namely the occurrence frequency information obtained from certain content-specific keywords (e.g., *said, good, bad, impression,* etc.).

Of the four sources mentioned, those features extracted directly from words tend to reflect the semantic aspect of the underlying text. While they may correspond directly to surface forms, very frequent forms are usually ignored (stopword removal) and remaining words are stemmed, and features having low occurrence frequencies (e.g., less than four) are usually ignored. Finally, text representation could be limited to the presence and absence of words or stems (set-of-words) or each feature could be weighted (bag-of-words). More sophisticated representations could generate by applying morphological analyzers returning the lemma and the part-of-speech (POS) for each surface word (e.g., *kings* produces *king*/*noun*). When accounting for POS information, we try to reflect intrinsic stylistic patterns and those aspects more orthogonal to the topics, while the bag-of-words feature tends to provide a better reflection of underlying semantics.

Although individual words tend to perform better than the POS-based features, a combination of both should tend to improve the quality of the learning scheme. As a third common representation, we could mention the selection of

a predefined set of function words (between 50 to 120) (e.g., *because*, *did*, *each*, *large*, *us*, etc.), together with certain punctuation symbols (e.g., $. ! %, etc.). Finally, a final text representation may simply account for the occurrence frequency of these terms.

After selecting the most appropriate features, we would then need a classification scheme capable of distinguishing between the various possible categories found in the input passage. Two possible learning approaches can be applied to achieve this objective: symbolic or machine learning. The symbolic (or knowledge engineering) school suggests using different knowledge representation paradigms (ontology, frames, rules) and inference mechanisms (first order logic, proposition calculus) to determine the category of a new text. Relying on the services of experts to build these representations would be difficult and expensive, and as such represents a real knowledge acquisition bottleneck. Tuning and verifying the final accuracy of a system like this would take time, and in the end updating the underlying categories or input information would usually require restarting the process at the beginning.

The machine-learning paradigm clearly dominates the field at present, and consists of different approaches (e.g., decision tree, neural network, naïve Bayes model, nearest neighbour, support vector machines (SVM)). These models are data-driven or based on an inductive process. To set them up, a set of positive and negative instances (training set) are provided, thus allowing the classifier to observe and determine the intrinsic characteristics of both positive and negative examples. These features are then searched within an unseen text and used to classify it. During this general scheme known as supervised learning, the system knows the exact category of each item belonging to the training set. When the appropriate learning scheme is selected, no further manual adjustments are needed, at least in principle, especially when a relatively large number of instances forms the training set. Once this resource is available, the machine-learning scheme can be applied. Otherwise, a training set must be built, a task that is however usually easier than creating a rule-based system. This process could be simply matter of adding a label to existing data. The training set must however reflect real and future items to be classified, and without this data, no classifier can be generated. Recently, some attempts have been made to develop mixed learning strategies, using both a machine learning model and some form of knowledge representation (be it a general thesaurus such as WordNet, or some manually written classification rules).

An analysis of the results of opinion detection on sentences [20] during the last ntcir evaluation campaign shows that the success rate (F-measure) was around 40% for the best system in English language, 64% for Japanese and from 55% to 68% for (simplified or traditional) Chinese sentences. In a similar evaluation, Boiy & Moens [7] reported an accuracy level of about 80% for the English language and 66% for the corresponding French part. From our point of view, these performance levels are encouraging but are not sufficient to allow current commercial applications. Given how important this topic is to individuals, firms and governments, we expect real improvements will be made in the near future.

In this vein, hybrid learning strategies may fill the gap between current performance levels and a level needed to obtain a successful commercial product.

5 Conclusion

Large amounts of data are freely available on the Internet, to individual users as well as companies and they all need to extract pertinent information items from this huge volume. For queries currently being submitted, search technologies are able to provide at least a few good responses, which in the best cases comprise numerous pertinent web pages. Current IR systems are mainly based on keyword matching, but they could still improve the quality of documents returned through making better matches between surface words appearing in the query formulation and in the web pages. This expected progress is somewhat limited however by the underlying features of all natural languages, particularly those related to the synonymy and polysemy linked to all natural languages on the one hand, and on the other, to the imprecise formulation of user's information need (see Section 2).

Machine translation services, the most effective of which are based on statistical models, allow searchers to retrieve the information they desire in the English language, even when the returned items are written in another language (German, Russian, Chinese, Japanese, etc.). The translations resulting from this process are not perfect and various problems still remain to be solved. The correct translation of names or selecting correct meanings from among two or more available translations as explained in Section 3 are just two examples.

The advantages of statistical approaches applied to natural language processing are that they can handle large amounts of data. Furthermore there are also many interesting and pertinent perspectives possible through the ability to detect opinionated sentences within blogospheres, etc. And then in an additional step classification could be done according to their polarity (positive, negative or mixed). Knowing the reactions of customers, the feelings of individuals or people general satisfaction will clearly have a real impact in marketing studies, market surveys, and also on public opinion follow-up. The web is here to stay and it will most certainly change the way we interact with each other. On the other hand, we really need to explore and develop techniques able to store, manage, search, translate and process large amount of textual information that can be found on the web.

Acknowledgments. This work was supported in part by the Swiss NSF, under Grant #200021-124389.

References

1. Manning, C.D., Raghavan, P., Shütze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge (2008)
2. Boughanem, M., Savoy, J. (eds.): Recherche d'information: Etat des lieux et perspectives. Lavoisier, Paris (2009)

3. Koehn, P.: Statistical Machine Translation. Cambridge University Press, Cambridge (2010)
4. Sebastiani, F.: Machine Learning in Automatic Text Categorization. ACM Computing Survey 14, 1–27 (2002)
5. Indurkhya, N., Damereau, F.J. (eds.): Handbook of Natural Languages Processing, 2nd edn. Chapman & Hall/CRC, Boca Raton (2010)
6. Abassi, A., Chen, H., Salem, A.: Sentiment Analysis in Multiple Languages: Feature Selection for Opinion Classification in Web Forums. ACM-Transactions on Information Systems 26 (2008)
7. Boiy, E., Moens, M.F.: A Machine Learning Approach to Sentiment Analysis in Multilingual Web Texts. Information Retrieval 12, 526–558 (2009)
8. Spink, A., Wolfram, D., Jansen, M.B.J., Saracevic, T.: Searching the Web: The Public and their Queries. Journal of the American Society for Information Science and Technology 52, 226–234 (2001)
9. Brin, S., Page, L.: The Anatomy of a Large-Scale Hypertextual Web Search Engine. In: Proceedings of the WWW'7, pp. 107–117 (1998)
10. Borodin, A., Roberts, G.O., Rosenthal, J.S., Tsaparas, P.: Finding Authorities and Hubs from Link Structures on the World Wide Web. In: Proceedings of the WWW'10, pp. 415–429 (2001)
11. Voorhees, E.M., Harman, D.K.: TREC Experiment and Evaluation in Information Retrieval. The MIT Press, Cambridge (2005)
12. Armstrong, T.G., Moffat, A., Webber, W., Zobel, J.: Improvements that Don't Add Up: Ad Hoc Retrieval Results since 1998. In: Proceedings ACM-CIKM, pp. 601–609. The ACM Press, New York (2009)
13. Hawking, D., Robertson, S.: On Collection Size and Retrieval Effectiveness. Information Retrieval Journal 6, 99–105 (2003)
14. Nielsen, J., Loranger, H.: Prioritizing Web Usability. New Riders, Berkeley (2006)
15. Danet, B., Herring, S.C. (eds.): The Multilingual Internet. Language, Culture, and Communication Online. Oxford University Press, Oxford (2007)
16. Crocker, C.: Lost in Translation. Misadventures of English Abroad. Michael O'Mara Books Ltd., London (2006)
17. Savoy, J., Dolamic, L.: How Effective is Google's Translation Service in Search? Communications of the ACM 52, 139–143 (2009)
18. Véronis, E., Véronis, J., Voisin, N.: Les politiques mis au net. Max Milo Ed., Paris (2007)
19. Crystal, D.: Language and Internet. Cambridge University Press, Cambridge (2006)
20. Seki, Y., Ku, L.W., Sun, L., Chen, H.H., Kando, N.: Overview of Multilingual Opinion Analysis Task at NTCIR-8. In: Proceedings NTCIR-8, pp. 209–220. NII publication, Tokyo (2010)
21. Hoover, D.L.: Another Perspective on Vocabulary Richness. Computers & the Humanities 37, 151–178 (2003)
22. Burrows, J.F.: Delta: A Measure of Stylistic Difference and a Guide to Likely Authorship. Literary & Linguistic Computing 17, 267–287 (2002)
23. Zhao, Y., Zobel, J.: Effective and Scalable Authorship Attribution using Function Words. In: Proceedings of the Second AIRS Asian Information Retrieval Symposium, pp. 174–189 (2005)
24. Stamatatos, E., Fakotakis, N., Kokkinakis, G.: Automatic Text Categorization in Terms of Genre and Author. Computational Linguistics 26, 471–495 (2001)
25. Zheng, R., Li, J., Chen, H., Huang, Z.: A Framework for Authorship Identification of Online Messages: Writing-Style Features and Classification Techniques. Journal of the American Society for Information Science & Technology 57, 378–393 (2006)

E-Tourism Portal: A Case Study in Ontology-Driven Development

Hafedh Mili, Petko Valtchev, Yasmine Charif, Laszlo Szathmary, Nidhal Daghrir, Marjolaine Béland, Anis Boubaker, Louis Martin, François Bédard, Sabeh Caid-Essebsi, and Abdel Leshob

LATECE Laboratory, Université du Québec à Montréal
http://www.latece.uqam.ca

Abstract. Software development is a fairly complex activity, that is both labour-intensive and knowledge-rich, and systematically delivering high-quality software that addresses the users' needs, on-time, and within budget, remains an elusive goal. This is even more true for internet applications presents additional challenges, including, 1) a predominance of the highly volatile interaction logic, and 2) stronger time-to-market pressures. Model-driven development purports to alleviate the problem by slicing the development process into a sequence of semantics-preserving transformations that start with a computation-independent model, through to an architecture-neutral platform independent model (PIM), all the way to platform-specific model or code at the other end. That is the idea(l). In general, however, the semantic gap between the CIM and PIM is such that the transition between them is hard to formalize. In this paper, we present a case study where we used an ontology to drive the development of an e-tourism portal. Our project showed that it is possible to drive the development of an internet application from a semantic description of the business entities, and illustrated the effectiveness of this approach during maintenance. It also highlighted the kinds of trade-offs we needed to make to reconcile somewhat lofty design principles with the imperative of producing a product with reasonable quality.

Keywords: model-driven development (MDD), ontology-driven development, computation-independent model (CIM), platform-independent model (PIM), platform-specific model (PSM).

1 Introduction

Software development, in general, is a fairly complex activity, and systematically delivering high-quality software that addresses the users' needs, on-time, and within budget, remains an elusive goal. There are many reasons for this, including, 1) incomplete or vague user requirements when projects start, 2) important conceptual gaps between the different representations of software throughout the lifecycle, and 3) the complexity and variety of knowledge domains that are brought to bear during development. All of these mean that automation is anywhere from hard to achieve to unattainable. The development

G. Babin, K. Stanoevska-Slabeva, P. Kropf (Eds.): MCETECH 2011, LNBIP 78, pp. 76–99, 2011.

of internet applications presents additional challenges, including, 1) a predominance of interaction/user-interface logic, which is typically more volatile than core business functions, and 2) more stringent time-to-market requirements, which put additional scheduling constraints on development. At the same time, internet applications present some characteristics which could make them more amenable to automation. First, notwithstanding the great variety of semantic content, the mechanics of the interaction logic are fairly predictable, leading to well-documented design patterns (e.g. MVC) and fairly powerful frameworks (e.g. Spring, Java Faces, etc.). Second, the underlying processing logic (the model component of the MVC), itself, tends to follow a relatively small number of process patterns. Both suggest that a transformational approach might be feasible for such applications [3]. Further, such an approach would help address the volatility and the time-to-market issue. In this paper, we explore the issue of using a transformational approach within the context of the development of an e-tourism portal.

Model-driven development (MDD) views software development as a sequence of transformations that start with a model of the domain entities and processes (computation independent model or CIM), producing along the way an architecture-neutral model of the software entities and processes (platform independent model, or PIM), all the way to platform-specific software models (PSMs) and code. The transformations embodied in the transitions from CIM to PIM and from PIM to PSM are inherently labour intensive and knowledge intensive. Modeling standards (e.g. UML, MOF, CWI, XMI) and a renewed interest in transformational approaches, have largely addressed the labour-intensive part of MDD. The codification of solution patterns (e.g. platform specific profiles) has embodied some but not all of the knowledge-intensive aspects of MDD transformations. A number of challenges remain, including:

1. Handling the CIM (i.e. business/domain model) to PIM (platform-neutral software model) transformation. Indeed, most MDD research focussed on the PIM to PSM transformation; we believe that the CIM to PIM transformation is at least as challenging [17].
2. Deciding which transformation (i.e. solution pattern/profile) to apply to a particular model, as in selecting a technology profile (EJB, web services, SCA, etc) to get a PSM from a PIM (see e.g. [2], [7], [9], [18]).
3. Having identified the transformation, marking appropriate entities of the input model so that they are transformed (see e.g. [15], [5], [4], [10]).

In other work, we have addressed the issue of choosing a transformation, within the context of detailed design [18] and marking input models to apply such transformations [10]. In this paper, we focus on the challenges of starting with a business model.

The application discussed in this paper is an *e-tourism portal* (ETP) that we are developing to help tourism organizations in so-called *least developed countries* to manage, promote, and sell their tourism services to—mostly international—tourists. ETP corresponds to what is commonly referred to as a *destination management system* [23]. ETP is part of the *United Nations Conference on*

Trade and Development e-tourism initiative (www.etourism.unctad.org). ETP is to be developed in increments, with the first iteration focussing on the collection, management, and publication of tourism information, leaving transactional and marketing capabilities to later versions [26]. From a high-level functional point of view, the first release of ETP corresponds to little more than a *content management system*, with functionalities for creating document templates to represent the different tourism services, creating descriptions (documents) of tourism services from those templates, searching them, managing their access, their lifecycle, and publishing them on the web. Given the great variety and volatility of the set of tourism services, but the uniformity of the underlying processes (creating, storing, versioning, searching, publishing, etc.), we opted for a generative approach. The question then is one of figuring out which representation is most appropriate for representing the semantics of the domain entities to support the various processes. We quickly realized that UML does not have the required expressive power or flexibility, and chose *ontologies* to represent domain entities, and more specifically, PROTEGE. In this paper, we explain how the resulting ontology was used to drive the development of ETP, and discuss the advantages and challenges of *ontology-driven development*, within the context of model-driven software engineering.

Section 2 presents UNCTAD's e-tourism initiative, and discusses the business and technical requirements of ETP. Section 3 presents the architectural solution framework. Section 4 provides a high-level description of the ontology (organization, rationale, etc.). Section 5 presents the different ways that the ontology was used in the development of the various components of ETP. Lessons learned and related work are discussed in section 6. We conclude in section 7.

2 UNCTAD's E-Tourism Initiative

According to the UN World Tourism Organization (WTO, www.world-tourism.org), tourism is the biggest industry in the world, accounting for nearly 10% of employment worldwide, and a significant part of the GDP, with international tourism generating close to 900 billion $ US in 2009, for an estimated 900 million arrivals. For many of the so-called *least developed countries (LDCs)*, tourism is *the* main—and for some, the *only*—export industry. And yet, it fails to generate the anticipated benefits for those countries. Notwithstanding problems of subcapacity, which prevents such countries from exploiting the full potential of their touristic assets, their reliance on first-world (foreign) tour operators means that only a small fraction (20%) of the economic benefits remain in the country. This is due to several reasons. First, without the means to promote their own tourism products independently, local providers are at the mercy of foreign tour operators who leave them with very small profit margins. Second, the hard currency earned from tourism sales is used, for the most part, to import supplies for tourism facilities (furniture, electrical and electronic equipment, food products, liquor, etc.), with little impact on the local economy, and leaving little hard currency on hand. Finally, a number of tourism facilities operate in an

unsustainable, environmentally 'unfriendly' fashion, adversely impacting other economic activities[1], and degrading the tourism capacity of the host country in the long run.

The *United Nations Conference on Trade And Development* (UNCTAD, www.unctad.org) has set-up the *e-tourism initiative* to help such countries increase their *autonomy* to develop and promote their tourism products directly to the target clientele. UNCTAD is based on the premise that developing countries can *develop* (or develop faster), if given the opportunity and means to *trade* in the open world market. The e-tourism initiative recognizes *tourism* as an economically important and widely shared *product* (service), and *e-technologies* as a promising technology that enables developing countries to make a low-cost entry in the world market. The e-tourism initiative thus aims at providing target countries (LDCs) with three components: 1) a *tool* to help them "identify, standardize, coordinate, and propose tourism services" to potential customers wherever they are, 2) a *method* for "collecting the relevant information, standardizing it, and disseminating it on the Internet", and 3) a *partnership building approach* to help actors in the public and private sectors, alike, to work together to share infrastructure, and offer more complex, value-added products. UNCTAD has partnered with our lab to develop the *tool* component, dubbed *e-tourism platform*.

The main objective of the E-Tourism Platform (ETP) is to enable *destination management organizations* in developing, least developed, and island countries to identify, classify, organize, market, and sell tourism services online to national and international tourists. In tourism parlance, a *destination management organization* (DMO) is an organization responsible for managing and promoting a tourism 'destination', and coordinating its actors. The destination could be a region (e.g. swiss alps), a country (e.g. Italy) or even a set of neighbouring countries. Typical DMOs include regional tourism offices, tourism ministries, or supra-national organizations. Systems for managing destinations are called *destination management systems* (DMS). Pollock defined *destination management systems* as "the IT infrastructure used by a destination organization for the collection, storage, manipulation and distribution of information in all its forms, and for the transaction of reservations and other commercial activities" [23]. The ETP platform should include core functionalities for *publishing* a tourism portal, but also for *building* and *maintaining* such portals. In software engineering terms, ETP is meant as a *framework* or *program family* from which individual platforms or portals can be instantiated, as opposed to a single portal. In addition to the typical functionalities of a *destination management system*, ETP has to satisfy the following main business requirements:

1. ETP has to support the business processes that underlie the business relationships that can exist between the various actors of the tourism industry. In doing so, ETP has to support a wide variety of processes that need to take into account the different levels of sophistication, both technical and

[1] For example, tourism facilities are notorious for being voracious water consumers, often competing with general water use and agriculture.

organizational, of the actors. Worse yet, within the same process, ETP needs to be able to integrate players with different levels of sophistication (say a western tour operator with a local bed & breakfast).
2. ETP has to support DMOs that are themselves loose federations of other destination management organizations. Such federated DMOs may emerge out of organizations such as the Economic Community of West Africa (www.ecowas.int), The Common Market for Eastern and Southern African (www.comesa.int), the Union du Maghreb Arabe (www.maghrebarabe.org), or from regional trading blocs in Southeast Asia. This has a number of implications on the business models, functionalities, and architecture of ETP.
3. ETP has to be delivered in increments that build on top of each other from both a functional and architectural point of view. The level of complexity and functionality of the increments should evolve on par with the needs and technical sophistication of the host DMOs.

From a development point of view, ETP has to be developed according to open standards, using open source software, and be, itself, open source so that individual DMOs are able to customize it for their own needs. Further, it has to be developed in technologies for which the hardware, software, and skills are readily available in the host countries. Indeed, UNCTADs initial financial and technical assistance notwithstanding, the host DMOs should be able to quickly appropriate the technology.

Figure 1 shows a simplified evolution scenario for ETPs usage context. It is expected that a first version of ETP will support basic functionalities for travelers and tourism suppliers. A subsequent version of ETP should enable instances of

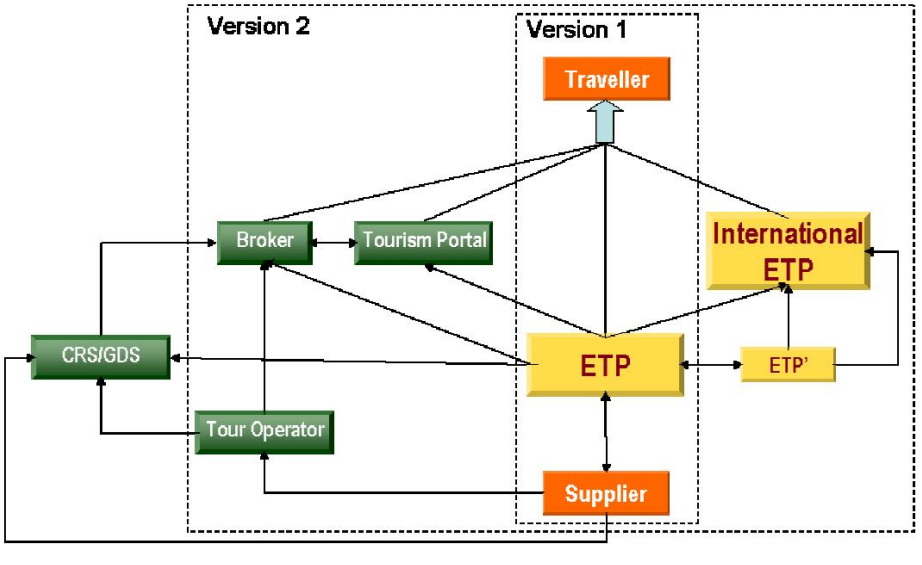


Fig. 1. Evolution scenario diagram for ETP. From [26].

ETP to interact with, a) other instances of ETP, including international cross-country or cross DMO ETPs, and b) other tourism players such as external tourism portals (e.g. expedia.com, travelocity.com), brokers, tour operators, or computerized reservation systems/global distribution systems.

3 ETP: The High-Level Architecture

As mentioned earlier, the first release of ETP needed to focus on the content management functionality, leaving transactional and marketing capabilities for later versions. It was tempting at first to think of ETP as little more than a content management system (CMS), and to adopt and adapt an open source CMS. Indeed, a number of open source CMSs, such as Joomla or OpenCms, support data collection, access control, and some more or less sophisticated publication functionality. However, such a solution suffers from many problems:

1. According to the business model proposed by the e-tourism initiative, data collection is to be performed in a decentralized, and off-line fashion, by unsophisticated users. This largely defeats the benefits of using the (thin) web client web server paradigm adopted by most CMSs.
2. Existing open source CMSs are not very good at supporting a clean (and enforceable) separation between data definition and data entry, nor at enforcing semantic data control.
3. The publication functionality of CMSs—open source or otherwise—is typically tightly coupled into the content management functionality, making it difficult to publish the content on a different platform.
4. Typical CMSs cannot support the evolution scenario described in Figure 1 (scalability, federated user profiles, etc.) or easily accommodate transactional and marketing functionalities planned for future releases.

That being said, ETP will certainly need a CMS component that acts as a central repository for tourism information, that manages the collected tourism data, and serves that data to a publication platform. The need to separate the various functions, and to support the evolution scenario shown in Figure 1 led us to adopt a loosely structured component- or service-oriented architecture for ETP that enables ETP components to talk to each other regardless of locality and jurisdiction. Figure 2 provides a fairly high-level view of the main components of ETP, and highlights some of the technical choices that we need to make. The plain components (LDAP directory, web portal and/or web server, and CMS) represent open source software components that we can probably reuse off the shelf as is (e.g. web server and LDAP directory) or through minor adaptations (web portal and CMS). The shaded components represent tools that will either be developed from scratch (e.g. recommender functionality), or through an adaptation of existing tools. Note also that we make no assumption about the location of the various components. It is expected that Data Collector instances be installed on portable computers which operate mostly off-line. The CMS, web

server/portal, and the LDAP directory need not reside in the same server[2]. In a federated system, a single LDAP directory would service multiple ETP instances (portals), and a single portal may be serviced by many CMS instances.

One of the first tasks of the project was to evaluate existing open-source frameworks/applications to implement the various components of the architecture, and to develop an *architectural proof of concept* with selected technologies to assess its feasibility.

4 The E-Tourism Ontology

4.1 Why a Tourism Ontology

The first version of ETP is about "the collection, storage, manipulation and distribution of [tourism] information in all its forms". For a given destination (country, region), ETP will publish information about accommodation facilities (hotels, lodges, youth hostels), historical landmarks, natural reserves, parks, transportation services, travel agencies, and the like. The first order of the day is to identify the kind of information that we need to describe about such tourism products, services, service providers, and the like, i.e. some sort of a conceptual data model of tourism-related information that reflects or embodies industry standards, i.e. a *tourism ontology*, according to Gruber's definition [12][3]. Thus, we knew we needed such a conceptual data model, and we knew that, at the very least, it would inspire the underlying database / document model of the CMS.

Further, the data collection component—called *Data Collector*—is, on some level, a data entry application for tourism information. Thus, a tourism ontology is needed for both the internal data model of *Data Collector* and for its GUI. Indeed, users of Data Collector will essentially be filling out forms that capture tourism-related information. For a full-service hotel, we need to enter a name, an address, a class (number of stars), contact and reservation/booking information, the different *types* of rooms, and the corresponding set of amenities and rates, common services (restaurants, bars, gift shops, fitness facilities, foreign exchange counters, etc.), neighbouring attractions, directions to the hotel, etc. Thus, we knew we needed to identify what that information should be, and to make sure that our data entry forms accounted for it.

When we contemplated the variety of tourism information that we needed to capture (over *fifty* categories) and the likely *frequent* evolution of that information, we quickly resolved to find ways to *generate the data entry forms from a program-manipulable representation of the ontology*, as opposed to building/coding the forms manually.

Section 4.2 will briefly describe the ontology design process. In particular, we will discuss the various sources of information we considered for building our

[2] For some target countries, it is conceivable that none of the server-side components will run in the country, due to the poor reliability of the basic electrical and telecom infrastructure, and to concerns over data security for transactional functionality.

[3] Gruber defines an ontology as a "formal explicit specification of a shared conceptualization".

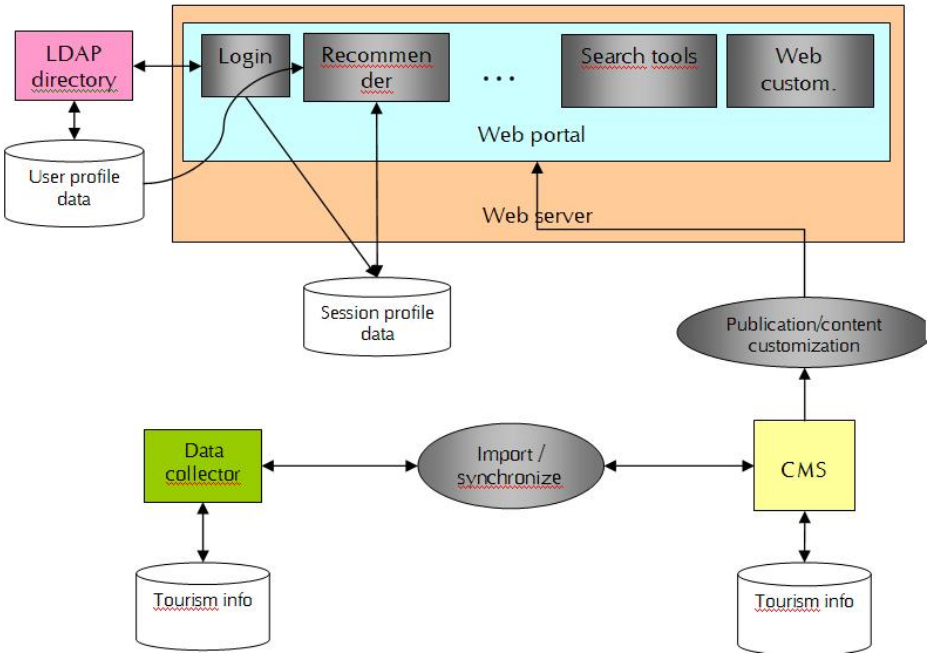


Fig. 2. A high-level architectural view of ETP

ontology, and how we used them. Section 4.3 presents the actual ontology where we present the high-level structure, and discuss the organizing principles.

4.2 The ETP Ontology Design Process

The "least-developed countries' " tourist sector is a subject that has been largely ignored in the mainstream e-tourism literature where developed world-grade infrastructure (e.g. golf fields, conference halls, credit card payments) is assumed [22]. In contrast, the tourism sector in the countries targeted by the ETP design covers a less diverse set of services while addressing specific concerns like health and security hazards, insufficient communication means, poor or non existing infrastructure, etc. This basically meant that there were virtually no available sources and even fewer experts in this narrow field. Therefore, the ontology design unfolded differently from the mainstream guidelines: rather than by a domain expert it was driven by our gradual grasp of the complex reality at hand. Hence it followed the iterative process of knowledge gathering by our team.

Ontology sources. Currently, there are few standardization efforts made by various professional organizations, tourism consortiums and university teams. Large multilingual specification corpuses have been created or are under way. The existing/emerging sources present highly diverging degrees of formality and structuring:

- Standardized *terminologies* are simple collections of normalized concept names.
- *Thesauri*: terminology hierarchically organized with "more-general-term-than" links.
- *Ontologies*: the concepts from a thesaurus represented in terms of valued properties.
- *Data-centred* specifications, e.g., database schemas, emphasize the data formats.

Open Travel Alliance (OTA) [22] has published a complete set of electronic messages to embody the information exchange pertaining to business activities in the tourism sector. These cover tasks like availability checking, booking, rental, reservation, reservation cancelling and modifying, etc. The main drawback of the implicit conceptual model these messages represent is their focus on travel instead of destinations. The ***World Tourism Organization*** (WTO) [28] is a United Nations agency that serves as a global forum for spreading the latest developments in practical tourism know-how. WTO publishes the ***Thesaurus on Tourism and Leisure Activities*** which is a complete multilingual collection of tourism terminology in use (currently, ca. 2000 concepts). It is intended for the standardization and normalization of a common language to be used in tasks related to indexing and search. Main shortage thereof is the lack of structure in term representations. The ***e-Tourism Working Group*** [11] at Digital Enterprise Research Institute (DERI) is behind the development of an advanced e-Tourism Semantic Web portal powered by a collection of travel-related ontologies, in particular for Accommodations and for Attractions plus Infrastructure. The ***Harmonisation Network for the Exchange of Travel and Tourism*** (HarmoNET) [14], is a network of 25+ members, including WTO which, "[...] provides and maintains a tourism specific ontology as a common definition of concepts and terms, their meaning and relationships between them. This ontology serves as a common agreement for the HarmoNET mediation service as well as a reference model for building specific data models or tourism information systems" [14]. Although the ontology design project is now over, the ontology was not yet freely available at the moment of the eTP inception.

Building strategy. The ontology design was initiated within the *Protégé* editor using the OWL language. These choices, that reflected our portability concerns, were later on reversed due to the need for the ontology concepts to embed information about data formats and entry form layout. While OWL provides a limited support for such external information (mainly as annotations), the Frames/Slots/Facets model native in *Protégé* (until the v. 4.0) happened to cover a large portion thereof, e.g., value ranges for numerical values, lists of admissible values for enumerations, etc. Hence we decided to trump the portability of OWL for the flexibility of *Protégé* Frames. Besides, this choice might be revisited once the latest development in the ontology field, OWL 2.0, gets more widely accepted.

The design process itself unfolded along the lines of the very general ontology development methodology in [21]: concept identification, concept hierarchy design, property identification, assignment of concepts to property domains and to property ranges. These steps were iterated on while incorporating new knowledge about tourism objects and feedback from the UNCTAD partners.

From an information processing point of view, our primary source has been the WTO thesaurus. Its content has been carefully scrutinized for terms pertaining to destinations. The relevant concepts have been introduced into the ontology draft and then inserted into its evolving hierarchical structure. For each of the newly identified domain concepts, three types of descriptors had to be elaborated:

1. Concept names were borrowed, whenever possible, from the WTO thesaurus. This insured available standard terminology was used to the highest possible degree.
2. Textual definitions to support concept comprehension were provided in the cases where the concept names admitted alternative interpretations.
3. At a later stage, the list of properties for each concept were identified. They comprised both properties to become concept attributes and hence to generate form fields and properties to hold references to (instances of) other concepts.

4.3 The eTP Ontology

The ontology was developed first in French as the initial customers of the eTP tools, in particular, *Data Collector* were French-speaking countries such as Mauritania and Togo. The ontology has undergone half a dozen major revisions, mostly bringing its content in line with its twofold role as both machine-readable domain representation and data schema for the *eTP* data. At its highest level of complexity, the ontology comprised 150 classes (frames) and 140 properties (slots). Currently, these figures are decreasing.

The following is the list of the high-level categories, i.e., the immediate descendants of the universal concept (Thing). They are split into main ones and auxiliary ones. In the list, classes are cited with their official names (in French, English equivalents given in brackets) and informal descriptions. **Main entity classes** are as follows :

- ***Activité*** (Activity). Comprises sport, leisure, cultural, well-being, etc. activities.
- ***AttractionTouristique*** (TouristAttraction). A tourist attraction is located at a site, e.g., the Sacré-Coeur cathedral is located at the Montmartre hill site.
- ***Destination*** (Destination). Roughly, this is a "place" in a geographic sense. Could be a whole country, a region (state, province), a natural park, a town/city, etc. Geographically smaller places are qualified as sites.
- ***FournisseurService*** (ServiceProvider). Sub-categorized with respect to the provided services, e.g., tourist offices may provide translation and information services while a restaurant may provide "à la carte" and fixed menu services.

- ***Manifestation*** (Event). Typically, a cultural event (festival, exposition, open-air performance, etc.). Opposed to *Activity* which is recurrent in the short term.
- ***Service*** (Service). Can be food-and-drink, lodging, transportation, etc. Price and availability in time is typically indicated. Category—specific information is also provided, e.g., departure and arrival times for transportation services.
- ***Site*** (Site). A destination may comprise a set of sites, e.g., the city of Paris comprises sites such as the Montmartre hill and the Père-Lachaise cemetery.

Auxiliary classes of the *eTP* ontology comprise : ***Adresse*** (Address), ***Contact*** (ContactPerson), ***HeuresOuvertureParJour*** (DailyWorkingHours), and ***Tarification*** (Fares).

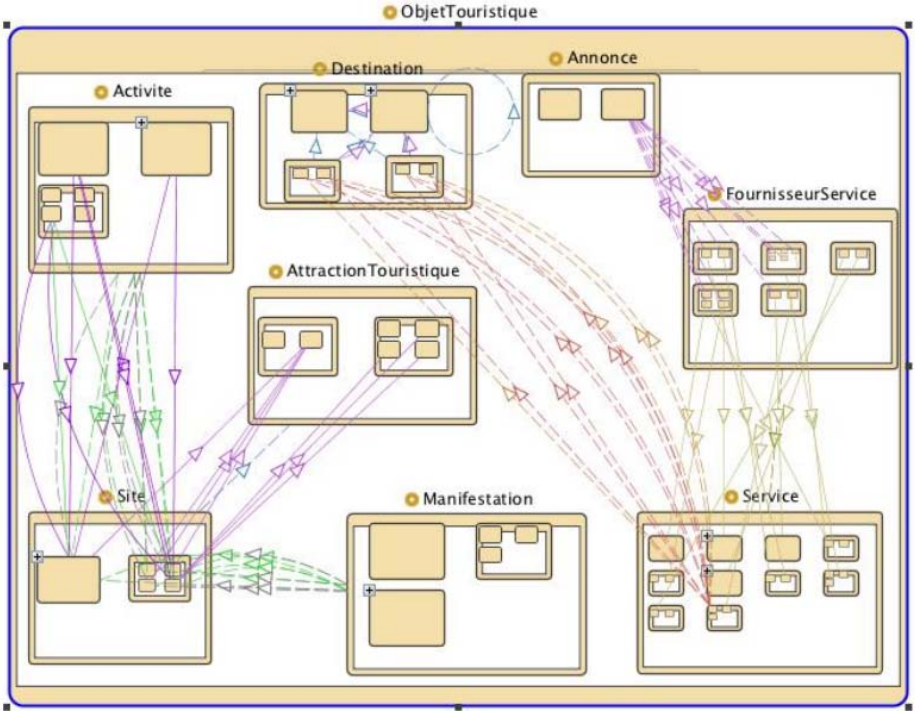


Fig. 3. Key ontology sub-categorizations (using *Jambalaya* visualization plugin within *Protégé*)

An overview of the ontology infrastructure is provided by Figure 3. It shows the recursive nesting of main classes of the ontology together with the relational properties that connect them, i.e., *Protégé* slots whose values are instances of classes (see the arrows on the diagram). The connection correspond to a set of 18 inter-class relationships. The large majority thereof define part-whole relations such as Service to ServiceProvider.

5 ETP: An Ontology-Driven E-Tourism Portal

The e-tourism ontology served many purposes in the project. In addition to capturing consensual knowledge of the various tourism services—a necessary exercise regardless of the implementation technology used for ETP—it was used to varying degrees to *drive* the development or customization of the various ETP components. We first talk about the role of the ontology in the development of *Data Collector* (DC). In fact, *most* of the functionality of Data Collector was *generated* directly *from the ontology*. In section 5.2, we explain how the ontology was used to configure the CMS—we used OpenCms (see www.openscms.org). Section 5.3 will talk about how the ontology was used to drive the publication component.

5.1 Data Collector

As mentioned in section 4.1, *Data Collector* is essentially a data entry application for tourism information. The tourism ontology that we developed identified over *fifty* (50) categories of tourism-related information that we may want to describe and publish on a tourism portal. This, with the likely continual evolution of the ontology meant that we had to use a computer-manipulable form of the ontology to generate the data entry forms. As explained in section 4.3, we used *protégé* to encode our tourism ontology.

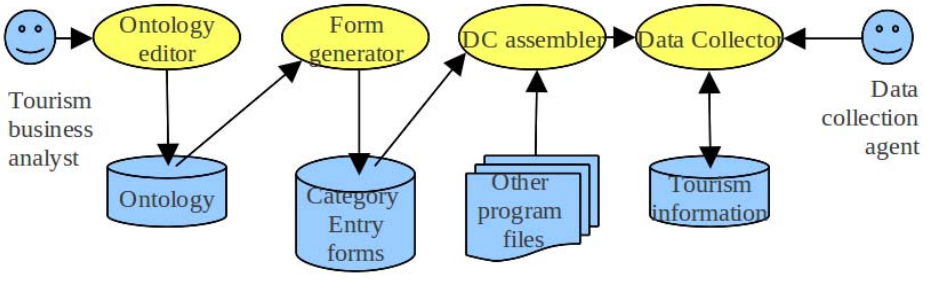


Fig. 4. Overall process from creating ontology to entering tourism information

Figure 4 represents the end-to-end process from the creation/editing of the tourism ontology to the actual data collection (data entry). This process involves a tourism business analyst who creates and maintains the ontology using an ontology editor—*protégé* in our case. A *form generator* reads the description of the ontology and builds input forms for the various categories of the ontology. While input forms constitute the bulk of *Data Collector*, some pieces need to be coded manually, like the login screen, the search functionality, and the synchronization functionality. Hence the *DC assembler* tool, which packages the generated forms along with the other programs to get the *Data Collector* application. A *data collection agent* can then use Data Collector to enter tourism information.

Let us consider, first, some of the design choices made for Data Collector, and then explore in more detail the kind of information that we needed from our ontology to support the generation of Data Collector.

From a technical standpoint, Data Collector is a set of input forms that access a database. Hence the question of which kind of database to use, and what technology for the input forms. Because of the number of categories (fifty), their complexity (what is needed to describe a full-service four-star hotel), and their volatility, it was clear to us that a 'literal' relational database was out of the question[4]. Not only would we have hundreds of tables, but each ontology change—regardless of how trivial—would require nightmarish data migrations. A 'flattened' relational data model, with a table for entities ("entity ID, entity Type ID") and one for attribute values ("entity ID, attribute ID, attribute value/value ID") would have a prohibitively sluggish save & load performance. This, along with the textual nature of many of the fields, and the need to communicate with a CMS, represents a textbook case for XML databases. We evaluated a couple of open-source databases, and settled on eXist (see http://exist-db.org).

Next came the question of input forms. Because Data Collector is meant as a desktop application, we had the choice of implementing the forms as Java (Swing) components, or as HTML forms that can be edited and visualized with a browser. Having chosen an XML database as a back-end, the W3C XForms standard came as a natural candidate[5]. (Very) roughly speaking, an XForm is described by a two-part schema, one specifying the data model, and the other specifying the presentation/view of that model, with a submit function that collects the data entered in the various fields, producing an XML document that conforms to the data model part. The presentation part of XForms comes with a set of predefined tags that implement the most common widgets (buttons, drop-down lists, single and multiple selection lists, text fields, date editors, etc).

We first considered generating XForms directly from the ontology. However, we quickly realized that we needed to support *run-time generation* of XForms, as opposed to compile-time. Indeed, more often than not, the contents of the drop-down lists depended on the *current contents of the database*, and thus, could not be generated off-line. For example, to enter an address, we needed to specify a city and a province or state. The prompt for province or state would have to show the list of provinces or states for the country or region at hand, and that list can only be obtained by querying the database live. Thus, instead of generating XForms directly from the ontology, we generated Java code that *generated* XForms during run-time. Because eXist is a web application, and it comes with a lightweight web server (jetty) that includes a servlet container, we chose to generate JSP pages that generated XForms, and deploy them on the same server as the database.

[4] Literal in the sense that each category is represented by a table, with repeating items themselves represented by separate tables.

[5] See http://www.w3.org/MarkUp/Forms

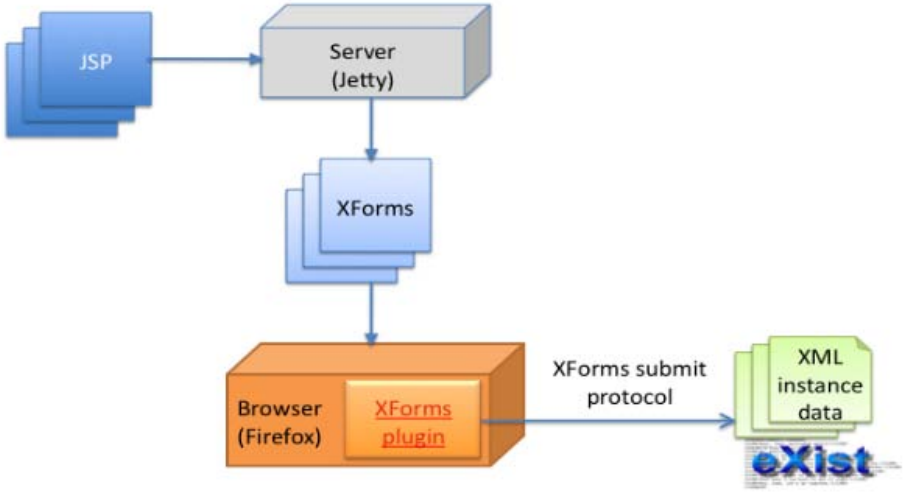


Fig. 5. Data Collector's actual run-time architecture

Thus, instead of being a desktop application, it is a web application. However, most of the time, it is installed as a *local web* application[6] on the laptops of the so-called data collection agents. Figure 5 shows the overall architecture of Data Collector.

Let us now consider the kind of information we needed, and used from the ontology. Roughly speaking, given a class/category in the ontology that represents a tourism product, service, or service provider, with a set of properties, we generated a form that included one prompt or field for each property. However, we did not generate a form for *each* category—auxiliary categories notwithstanding (see section 4.3). For example, we have **Service** as a category, with subcategories for **Lodging**, **Transportation**, etc., and **ServiceProvider**, with subcategories for **LodgingProvider**, **TransportationProvider**, etc, with subcategories for **Hotel**s, **Bed&Breakfast**s, **YouthHostel**, **BusCompany**, **TrainCompany**, **TaxiForHire**, **Airline**, and so forth. The higher-level categories are similar to *abstract classes* in OOP languages, and should not be instantiated; only 'concrete categories' will map to forms. However, not all leaf categories should map to forms either. For example, none of the auxiliary categories (see section 4.3) will map to forms. Thus, we had to tag the ontology categories to identify the ones for which we needed to generate forms.

Let us now talk about the properties. In Protégé, properties have names, types, and a bunch of *slots/facets*. To prompt for the value of a property in a form, we needed a display name or *label*, a *data type* and/or *value type* to pick the appropriate widget to enter the value. For example, a price, which is a *type-in* (modality) number (data type), calls for a simple textual prompt that accepts only digits and signs (+, -), whereas a property whose value is one of *an enumerated* set of strings

[6] A web application running on localhost.

(e.g. a currency) might call for a single-selection drop-down list. And so forth. In fact, Protégé's slot mechanism enables us to add numerous constraints on property values, and we were only limited by what we could enforce in XForms. Finally, note that for a particular category, we need to pull out all of the inherited properties and include them in the corresponding generated form; the Protégé (Java) API enables us to easily navigate the taxonomy.

As we started worrying about usability and presentation issues, we quickly realized that we needed to *decorate* the semantic information that is embodied in the ontology, with syntactical and graphical information. First, there is the issue of *labels*. As mentioned above, for each property, we needed a textual label that will appear in the input form around the value editor for that property. Then, there is the issue of *localization*. Because we want ETP, in general, and Data Collector, in particular, to be localized, we need to provide labels in each of the target languages. Then there is the issue of form layout, with many sub-issues. First, there is the question of ordering. Take the example of a **Hotel**. A hotel has a name, an address, a phone number or two (e.g. mainline and reservations), a short (punchy) textual description, possibly a URL, and a detailed description of the services offered by the hotel (room types and rates, amenities, etc.). Whether we are creating, editing, or visualizing the form for a hotel, we expect to see the fields/properties appear in the order given, i.e. first the name, then the address or the phone number, and so forth. This order has to be specified somewhere for proper form generation.

Then there is the issue of form length. A full-service hotel may require the entry of dozens of properties, when we consider repeating groups such as different room types, bed sizes, or amenity packages. Entering all of this information in one long form is tedious and user-*un*friendly. Thus, we set out ways to automatically split long forms into multiple pages. To do that, we need to estimate the vertical height taken by each property, based on its type, and based on the minimum and maximum size of a page, distribute the different properties on different pages, while ensuring that *composite properties* such as addresses (with streetNumber, streetName, streetDirection, city, provinceOrState, postalCode), stay on the same page.

From a methodological point of view, we were a bit reluctant to include non-semantic information in the ontology, and least of all *presentation*-level information. As is customary in language development frameworks, for example, we considered keeping the two pieces of information separate: 1) only-semantic information in the Protégé ontology, and 2) syntactic or presentation-level information of the kind discussed above in an external resource, with anchor points to the ontology. For example, the syntactic or presentation-level information can be represented in some sort of a property file with key-value pairs of the form "ontology element". "syntactic / presentation property" = "value". The problem with such a solution is the need to edit two separate media whenever the ontology evolves, and the difficulty in keeping them in sync in case of non-trivial evolutions. Hence, we kept it all in the ontology. Interestingly, we supported

'run-time localization'[7] of *Data Collector* thanks to such a file, where the "syntactic / presentation property" could be things such as label.en_us or toolTip.fr, and the property value is the corresponding string. However, such a property file is *generated automatically* from the Protégé ontology.

5.2 Content Management: OpenCms

ETP requires a CMS component to centralize data coming from distributed Data Collector clients, to control data access, to serve the portal with tourism information and services, and for other administrative and publication functionalities. One of the development scenarios that allowed to evaluate the CMS candidates was to make the CMS functionalities, documents, and schemas accessible as web services. We chose to adopt the open source CMS OpenCms (www.opencms.org) for this component because it is easy to program, allows to control the visibility of resources, has a light code, and is provided with a good and up-to-date documentation.

OpenCms is based on Java and XML and can be deployed in an open source environment (e.g. on Linux, with Tomcat and MySQL) as well as on commercial components (e.g. Windows NT, with IIS, BEA Weblogic, and Oracle). Resources are connected from the database to a *virtual file system* (VFS), which can in turn be accessed through OpenCms so-called *workplace*. OpenCms is provided with a project mechanism that offers an integrated workflow environment with Offline "staging" and Online "live" systems on the same server. All content is maintained in projects. The number of projects is unlimited. Files (JSP, XML, folders, etc) can be created and edited, changes to the contents can be reviewed, thoroughly tested and approved in the Offline "staging" project before the content is published. To make content visible to the public, we simply need to publish it to the Online view.

Based on our analysis, we extended OpenCms with the following settings and functionalities:

1. **A *directory structure*:** tourism information collected by Data Collector clients, to be later exposed on the portal, is stored and organized in a directory structure on the CMS. This directory structure is in fact derived from the ontology by the TP&S form generator. The latter takes as input the ontology hierarchy structure and generates a directory structure. We created a JSP script in the VFS to generate such a structure. It takes as input a compressed file containing the directory structure[8] and generates it on the VFS together with a properties file reflecting the hierarchy structure.
2. ***Web service access*:** based on the need to make the CMS functionalities and resources accessible for the Data Collector and the portal components, we integrated an open source JAX-RPC implementation of OpenCms web

[7] Users can switch languages while Data Collector is running.

[8] The compressed file also contains XSLT sheets for layout customization. This is discussed in section 5.3.

services stack (http://sourceforge.net/projects/opencmsws-jaxrp/). This service provides an API for logging onto the CMS, creating and editing files on the VFS, and setting their properties (e.g. in our case, *owner*, *required*).

3. ***DMO administrator functionalities*:** as the DMO administrator needs to manage instance data ownership, visualization, edition, and web customization, we extended OpenCms menu accordingly.

Synchronization

Instance data collected by Data Collector components are synchronized with OpenCms through the procedure depicted on Figures 6 and 7. The Data Collector agent must first authenticate on OpenCms before he can synchronize his resources. To this end, he uses a credentials file generated by the DMO administrator. Indeed, each agent is assigned to ask the DMO administrator to provide him with a credentials file encrypted with the agent machine's MAC address. This file must then be stored on the agent's Data Collector home directory to be later used by the synchronization tool.

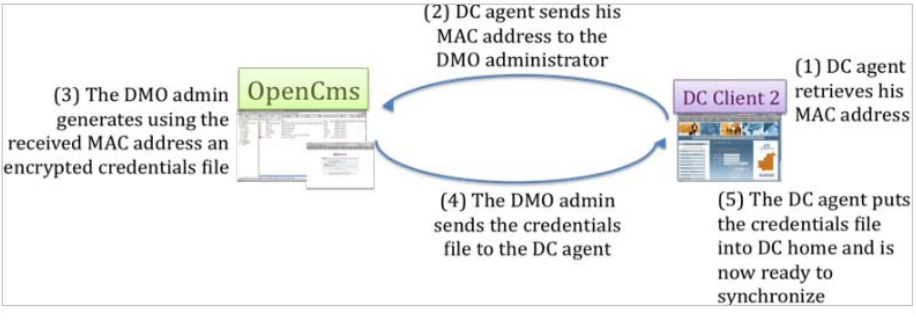


Fig. 6. Authentication procedure before synchronization

We implemented the synchronization tool as a Java application that executes on Data Collector client side to enable him importing and exporting instance data from/to OpenCms. An agent who creates a new resource and exports it to OpenCms is by default its owner. The DMO administrator can change the ownership of the resource and assign it to another agent. Therefore, the resource becomes accessible in readonly mode for the creator agent (see Figure 7).

The required resources, those can be referenced by others such as **Country** and **City**, are explicitly preselected during the import procedure.

The incorporated DMO administrator menu options, specifically instance data visualization and edition, required to dispose of a server version of Data Collector. Using this server version, an administrator can view and edit XML data using the appropriate XForms, and directly save it on OpenCms. XForms of Data Collector server are generated simultaneously with those of Data Collector client by DC application generator.

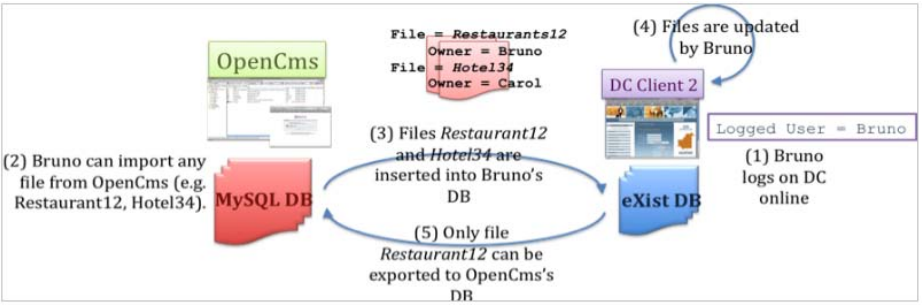


Fig. 7. Synchronization protocol

5.3 Publication

Instance data synchronized from Data Collector clients to OpenCms lands in the Offline staging project. It takes an explicit action from the DMO administrator to publish it on the Online live project. Therefore, only validated data is visible by the portal.

The portal component in ETP is meant to expose to the end-user TP&S data and to support search functionalities, layout customization, and subscription to specific events and RSS feeds. Development scenarios were implemented in order to distinguish the portal candidates to evaluate. For instance, the welcome page needed to contain three types of content: static for all users, dynamic for all users (e.g. weather of the day, daily activities), and user specific content. Also, portlets deployed within the portal needed to communicate, so that their content is refreshed accordingly (e.g. a click on a category in the menu portlet has to imply the display of that category instance data in the TP&S portlet, and the corresponding promotions in the promotions portlet).

We adopted the open source enterprise portal Liferay (http://www.liferay.com/) for this component because it is easy to acquire and test, and it handles nicely portlet integration, server configuration, and web content customization[9].

Liferay Portal is written in Java and can run both on J2EE application servers (to exploit EJB) and lighter servlet containters such as Tomcat. This portal system is built on portlets and as such, there are many third-party community-contributed plugins and add-ons. It also includes a suite of applications such as blogs and instant messaging.

Based on our analysis, we implemented the following functionalities on Liferay:

1. ***Portlets*:** several portlets have been implemented to organize and expose the tourism information collected: A menu portlet with the data categories (e.g. Destination) and sub-categories (e.g. City, Village), a TP&S portlet to display instance data for each sub-category, a promotions porlet to display valid promotions when a given sub-category is selected, a headlines portlet to

[9] Also, Liferay was named by Gartner a leader in the magic quadrant for horizontal portals.

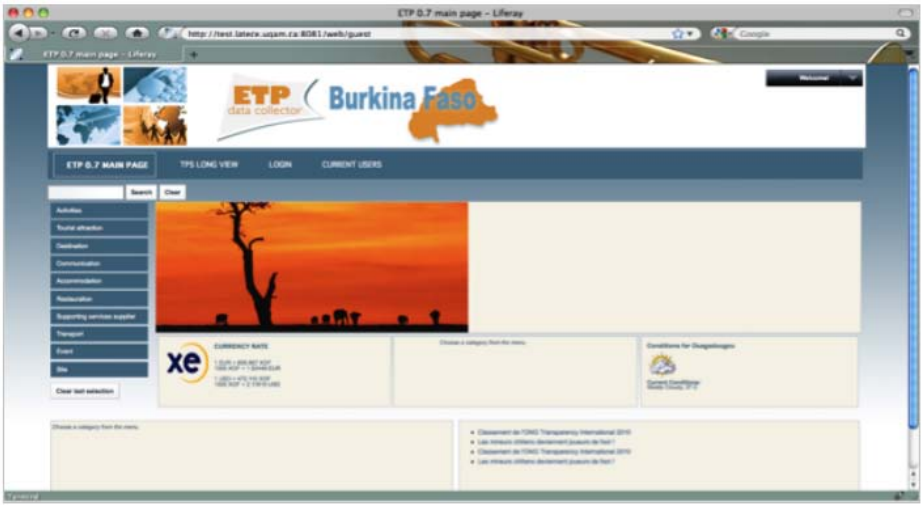


Fig. 8. Screenshot of the ETP's welcome page

expose news, weather and currency rate portlets, a search portlet to retrieve instance data containing keywords, and a login portlet.

2. ***Layout*:** a new layout template has been designed to arrange the way portlets need to be arranged on the portal page (see Figure 8).
3. ***Theme*:** in Liferay, a theme controls the whole look and feel of the portal pages. As such, a new theme template has been implemented with customized banners and CSS.

Tourism information exposed on the portal is basically imported from the CMS through services invocation. Three services have been implemented on the CMS for that purpose: 1) a service that returns instance data given a category or a sub-category, and eventually keywords, 2) a service that returns valid promotions given a (sub-)category, 3) a service that returns valid headlines. In order to avoid recalculating the set of valid promotions and headlines at each service invocation, the two latter services use caching.

Using the promotions and headlines inherent attributes (see Figure 9), more specifically *expiry-date*, *showcase*, and *category*, the services can determine which promotion or headline can appear on the portal's main page.

Once an instance data is imported on the portal, its content is being formatted using XSLT sheets. Similarly to XForms and the hierarchy directory, the XSLT style sheets are derived from the ontology. Given a tourism TP&S schema, the TP&S generator produces three XSLT sheets for each TP&S sub-category (e.g., City, Museum, SportActivity). The three XSLT correspond to the tabular, short, and long display options of instance data on the portal. For each sub-category, the corresponding XSLT sheets are stored on the CMS within the directory structure.

Fig. 9. Excerpt of the Promotion entity object model

We wanted the portlets content to be customizable according to each authenticated user. For later version of ETP, we aim to incorporate a recommender system fed with the user actions on the portal. Therefore, as a proof of concept we added a simple field that a user can fill with keywords on the registration portlet. A portlet filter handles then filtering the service results according to this principled field content.

6 Discussion

6.1 Lessons Learned

The first version of ETP is about "the collection, storage, manipulation and distribution of [tourism] information in all its forms". A first step in eliciting what needed to be done consisted of identifying and characterizing our shared understanding of tourism information. Hence, the idea of an ontology. The relative simplicity and uniformity of the handling of this information (creation, editing, and viewing) suggested that a good part of the functionality could be generated from a description of the data. This was particularly true for Data Collector, whose functionality was, for all practical purposes, limited to creation, editing, and viewing of tourism data. Indeed, over 95% of the functionality of Data Collector was automatically generated. However, as shown in sections 5.2 (CMS component) and 5.3 (portal), the ontology was also used to configure the other components of the system, namely OpenCms's *virtual file system*, and the category-specific XSL style sheets, stored in OpenCms, and used by the portal (Liferay).

At a general level, the generative approach adopted for Data Collector was critical to the success of the project. Our customer was nervous at first when we could not produce a single form for the simplest of tourism services for the first few weeks: while part of the team was working on the ontology, the other members were working on the generator. However, once the ontology was completed, any enhancement to the generator functionality was propagated to the fifty or so forms. If we think of individual forms as increments of functionality, our approach was certainly not incremental: we went from none to a 'poor quality' fifty, and then kept enhancing those fifty. However, our approach was incremental in terms of the generator functionality, and to some extent, the ontology itself. It was important to explain the generative paradigm to the customer, and to define with them palpable measures of progress; we figured that the same must

be true for projects that use the generative approach. This approach pays off handsomely during maintenance. Late in the project, some user tests with the tool revealed a number of problems with the forms / tourism categories: a) some were overly detailed, given the target countries, and needed to be simplified, b) other categories were too conceptually close, and c) others were plain wrong. This led to a substantial reorganization of the ontology. The reorganization simply required re-running the generator, and some of the associated configuration scripts: no Data Collector code was manually edited. The speed with which this could be done was quite a relief to the customer.

This experience can be seen within the context of the now-familiar MDD paradigm. To the extent that the ontology captures the *semantics* of the *business data*, it plays the role of the *computation independent model* (CIM). However, as shown in section 5, some very computation-specific information crept into the ontology: indeed, we had to add a number of syntactic as well as presentation-level properties to the ontology to make the generated code of reasonable quality. This is a general dilemma that we, software engineers, face when we try to translate design ideas and ideals such as separation of concerns, design patterns, layered architectures, or MDD, into reality within the context of real projects: we have to balance purity of concept/process with observable qualities of the product. The following table compares common design principles, common transgressions, and the reasons for those transgressions.

We will refrain from drawing broad conclusions regarding what we called *ontology-driven development* from this *one* experiment. Further, as a computation-independent model (CIM), the ontology said nothing about the *behaviour* of the entities; it did not need to, because the processing was the same for all of the entities: create, read, update, and delete. In other work, we are exploring ways of *assisting* in the generation of analysis-level models (PIMs) from business process models (CIMs), and it is the behavioural models that are the most challenging.

6.2 Related Work

Several case studies have been performed to report experience on benefits of MDD ([16], [1], [25]) and ontology-driven development ([13], [6]).

For instance, [16] propose to develop software with the same functionality twice: 1) "by hand" in a code-centric, conventional style, 2) and using Eclipse-based MDD tools. The authors show that developing functionally comparable software took about nine-fold more time if the source code was written by hand, compared to MDD. Time was saved thanks to existing transformations with the first development cycle, to good MDD templates that lowered the probability of failure for the generated code, and to existing framework and the infrastructure offered by EMF, GMF, and GEF. Similarly, [1] adopt MDD to address the problem of developing enterprise-class eBusiness solutions in a more economically viable and time-effective way. It elaborates on a six-years experience of applying MDD to a set of enterprise-scale applications using WebRatio, an MDD methodology and tool suite based on the WebML meta-model. These approaches map a PIM to a Platform Specific Model (PSM) whereas we are more

Design principle	Justification	Transgression	Justification
CIM / PIM / PSM separation	The very idea that we could build a succession of models that, a) separate business concerns (CIM), analysis concerns (PIM) and design concerns (PSM), and b) could be transformed automatically, promotes the reuse of these models across a wider solution space	To ready a model for a transformation to the next level, we have to *manually* (essentially) annotate it with information unrelated to the concern at hand	Fill-in the knowledge gap to ensure that the right transformation is applied to the right elements
MVC	Separating business logic from presentation logic so that both can evolve independently	Model is made aware of interaction	Semantic feedback (e.g. meaningful error messages)
Layered architectures	Each layer accesses only the functionality provided by the layer below it so that the different layers can be interchanged	A layer digs (accesses functionality) several levels deep	Performance optimization

interested in CIM as a starting point of the development. In contrast, [8] propose to drive the development process by CIM. The latter includes the business analysis model and the global business model which is composed of function/process models, organization models, and information models. This approach does not however give details on the CIM to PSM transformation.

Ontologies have for the last decade been firmly making their way among software practitioners (see [19] for a pre-Semantic web report) and the variety of uses within the development life-cycle is increasing steadily. A good, albeit somewhat outdated overview of the ontology use-cases in software engineering is provided in [13]. The authors argue that ontologies have a place in the development process both as architecture guide (e.g. as versatile domain models) and as infrastructure component (semantic web services enabler for SOA). According to the proposed categorization of ontology roles, our own approach qualifies as ontology-driven development (see [27]). In this respect, a thorough report of the experience with a project similar to ours, yet of a much larger scope may be found in [20]. The authors share the lessons learned from using several ontologies (within the Protégé framework) as complete representational infrastructure for the content of a Web portal. On the opposite side of the aforementioned categorization, [6] use an ontology as a contract between business and IT to agree on the meaning of concepts, and as a mechanism that allows the business to formalize their specification. The actual application development is then to be realized by transferring the knowledge expressed in the ontology to suitable

objects, types and constructs in the programming language of choice. The process for transferring knowledge from the ontology to the programming language is by automatic generation of source code.

7 Conclusion

In this paper, we presented a case study of developing an internet application based on an ontology of the business domain, i.e. a description of the semantics of the business data. Seen within the context of *model-driven development* (MDD), this project showed that it is possible to develop an application starting with a *computation-independent model* (CIM). This was made possible, in part because the behavioural aspects of our application are simple and common to the various business entities. Seen within the context of *generative development*, this project confirmed the effectiveness of this approach when the application domain is well circumscribed. Either way, it showed the effectiveness of specification-level maintenance: important changes to the domain model were handled at the ontology level and automatically propagated to the software. The project also highlighted the kinds of compromises we need to make to turn somewhat lofty design principles into practical—and acceptably high quality—solutions.

References

1. Acerbis, R., Bongio, A., Brambilla, M., Tisi, M., Cerri, S., Tosetti, E.: Developing eBusiness Solutions with a Model Driven Approach: The Case of Acer EMEA. In: Baresi, L., Fraternali, P., Houben, G.-J. (eds.) ICWE 2007. LNCS, vol. 4607, pp. 539–544. Springer, Heidelberg (2007)
2. Albin-Amiot, H., Guéhéneuc, Y.G.: Meta-modeling Design Patterns: application to pattern detection and code synthesis. In: Proceedings of ECOOP Workshop on Automating Object Oriented Software Development Methods (June 2001)
3. Alencar, P.S.C., Cowan, D.D., Dong, J., Lucena, C.J.P.: A transformational Process-Based Formal Approach to Object-Oriented Design. In: Formal Methods Europe FME 1997 (1997)
4. Baxter, I.D.: Design Maintenance Systems. Communications of the ACM 35(4), 73–89 (1992)
5. Biggerstaff, T.J.: A New Architecture for Transformation-Based Generators. IEEE Transactions on Software Engineering 30(12), 1036–1054 (2004)
6. Bossche, M.V., Ross, P., Maclarty, I., Van Nuffelen, B., Pelov, N.: Ontology Driven Software Engineering for Real Life Applications. In: Proceedings of the 3rd International Workshop on Semantic Web Enabled Software Engineering, SWESE (2007)
7. Budinsky, F.J., Finnie, M.A., Vlissides, J.M., Yu, P.S.: Automatic Code Generation from Design Patterns. IBM Systems Journal 35(2), 151–171 (1996)
8. Che, Y., Wang, G., Wen, X.X., Ren, B.Y.: Research on Computational Independent Model in the Enterprise Information System Development Mode Based on Model Driven and Software Component. In: Proceedings of the International Conference on Interoperability for Enterprise Software and Applications, pp. 85–89 (2009)
9. Elaasar, M., Briand, L., Labiche, Y.: A Metamodeling Approach to Pattern Specification and Detection. In: Proceedings of ACM/IEEE International Conference On Model Driven Engineering Languages and Systems (MoDELS), Genoa, Italy, October 1-6 (2006)

10. El-Boussaidi, G., Mili, H.: Detecting Patterns of Poor Design Solutions Using Constraint Propagation. In: Busch, C., Ober, I., Bruel, J.-M., Uhl, A., Völter, M. (eds.) MODELS 2008. LNCS, vol. 5301, pp. 189–203. Springer, Heidelberg (2008)
11. e-tourism working group, DERI, http://e-tourism.deri.at/
12. Gruber, T.R.: Toward principles for the design of ontologies used for knowledge sharing. International Journal of Human Computer Studies 43(5-6), 907–928 (1995)
13. Happel, H.-J., Seedorf, S.: Applications of Ontologies in Software Engineering. In: International Workshop on Semantic Web Enabled Software Engineering, SWESE (2006)
14. HarmoNET, the harmonisation Network for the exchange of travel and tourism information, http://www.etourism-austria.at/harmonet/
15. Haydar, M., Malak, G., Sahraoui, H., Petrenko, A., Boroday, S.: Anomaly Detection and Quality Evaluation of Web Applications. In: Handbook of Research on Web Information Systems Quality, pp. 86–103. IGI Publishing (2008)
16. Krogmann, K., Becker, S.: A Case Study on Model-Driven and Conventional Software Development: The palladio editor. In: Software Engineering Workshops, vol. 106, pp. 169–176 (2007)
17. Mili, H., Leshob, A., Lefebvre, E., Lévesque, G., El-Boussaidi, G.: Towards a Methodology for Representing and Classifying Business Processes. In: Babin, G., Kropf, P., Weiss, M. (eds.) E-Technologies: Innovation in an Open World. Lecture Notes in Business Information Processing, vol. 26, pp. 196–211. Springer, Heidelberg (2009)
18. Mili, H., El-Boussaidi, G.: Representing and Applying Design Patterns: What Is the Problem? In: Briand, L.C., Williams, C. (eds.) MoDELS 2005. LNCS, vol. 3713, pp. 186–200. Springer, Heidelberg (2005)
19. Mussen, M.: Domain ontologies in software engineering: use of Protégé with the EON architecture. Methods Inf. Med. 37(4-5), 540–550 (1998)
20. Nyulas, C.I., Noy, N.F., Dorf, M.V., Griffith, N., Musen, M.A.: Ontology-Driven Software: What We Learned From Using Ontologies As Infrastructure For Software. In:5th International Workshop on Semantic Web Enabled Software Engineering (SWESE) at ISWC 2009(2009)
21. Noy, N., McGuinness, D.: Ontology Development 101: A Guide to Creating Your First Ontology (2001), http://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html
22. Open Travel Alliance, http://www.opentravel.org/
23. Pollock, A.: Destination management systems, reported By Travel Daily News (March 2003), http://www.travel-dailynews.com/makeof.asp?central_id109&permanent_id=12 (2001)
24. Pollock, A.: Taking Off: e-Tourism Opportunities for Developing Countries. In: Information Economy Report, United Nations Conference on Trade and Development, UNCTAD, ch. 4 (2005)
25. Staron, M.: Adopting Model Driven Software Development in Industry – A Case Study at Two Companies. In: Wang, J., Whittle, J., Harel, D., Reggio, G. (eds.) MoDELS 2006. LNCS, vol. 4199, pp. 57–72. Springer, Heidelberg (2006)
26. Implementing an e-tourism portal for UNCTAD within the context of the e-tourism initiative: a proposal, technical report, UNCTAD (2006)
27. Ontology Driven Architectures and Potential Uses of the Semantic Web in Systems and Software Engineering, W3C (2006)
28. World tourism organization, http://unwto.org/

Toward a Goal-Oriented, Business Intelligence Decision-Making Framework

Alireza Pourshahid[1], Gregory Richards[2], and Daniel Amyot[1]

[1] School of Information Technology and Engineering, University of Ottawa, Canada
alireza.pourshahid@gmail.com, damyot@site.uottawa.ca
[2] Telfer School of Management, University of Ottawa, Canada
richards@telfer.uottawa.ca

Abstract. Decision making is a crucial yet challenging task in enterprise management. In many organizations, decisions are still made based on experience and intuition rather than on facts and rigorous approaches, often because of lack of data, unknown relationships between data and goals, conflicting goals, and poorly understood risks. This paper presents a goal-oriented, iterative conceptual framework for decision making that allows enterprises to begin development of their decision model with limited data, discover required data to build their model, capture stakeholders goals, and model risks and their impact. Such models enable the aggregation of Key Performance Indicators and their integration to goal models that display good cognitive fit. Managers can monitor the impact of decisions on organization goals and improve decision models. The approach is illustrated through a retail business real-life example.

Keywords: business process management, business intelligence, decision support systems, goal-oriented modeling, indicators.

1 Introduction

Decision making is a crucial yet challenging task for many managers. Many challenges arise from the rapid growth of data within an operating environment of continuous change and increasing customer demands. Although many enterprises have applied different decision aids such as Business Intelligence (BI) tools in an attempt to improve decision-making capability, these approaches have not always met with success. We believe that one of the problems with the use of such tools is the lack of approaches that integrate goals, decision-making mechanisms and Key Performance Indicators (KPIs) into a single conceptual framework that can adapt to organizational changes and better fits manager's cognitive decision models. A secondary issue relates to the unavailability of sufficient data when performance models are first put in place.

The purpose of this paper is to describe the development of such a BI framework, and also the technical means to implement it in the enterprise. The paper first describes some of the issues related to decision making using BI tools. It then describes extensions of the Goal-oriented Requirement Language (GRL) used as

G. Babin, K. Stanoevska-Slabeva, P. Kropf (Eds.): MCETECH 2011, LNBIP 78, pp. 100–115, 2011.

a modeling environment to support the aggregation of KPIs (whose values are either coming from external data sources through Web services or simulated as part of what-if strategies) and their integration to the rest of the goal model during formal analysis. A new formula-based goal evaluation algorithm is introduced that takes advantage of this aggregation of KPIs. In addition, the paper provides the implementation steps of the proposed BI-supported decision framework, which are applied iteratively to a retail business example where little data is available at first. Finally, lessons learned and conclusions are discussed.

2 Background Review

2.1 BI-Based Decision Making

Over the past 30 years, the growth of BI technology has helped managers make better decisions through improved organization of information, better data quality, and faster and more effective delivery of information. It has been estimated, however, that more than 50% of BI implementations fail to influence the decision-making process in any meaningful way [10]. Reasons for this include cultural resistance, lack of relevance, lack of alignment with business strategy, and lack of actionable and "institutionalized" decision support technologies [8]. Many of these problems could be attributed to approaches used for defining the data to be delivered by the BI tool.

Most data delivery schemes are based on dimensional models of the data. This approach often leads to a sound technical data model, but this view of the data might or might not fit with the user's *decision model*. Indeed, Korhonen *et al.* [11] point out that one of the key challenges faced in institutionalizing decision aids is validation of decision models used by the *decision maker*. These authors argue that problems with model validation occur when relationships between the variables included in the decision model are not accurate and when the available data does not match the model's specifications. Although the data model is often developed by first defining user needs in terms of the variables (i.e., data values) required, this approach does not necessarily illustrate *relationships* between the variables nor does it define variables in a cause-effect framework that matches the decision model used by decision makers. Thus the technical data model differs from the decision model.

Vessey [18] further suggests that more effective decision making results when the decision aid directly supports the decision task. The essence of this argument revolves around the notion of *cognitive fit*, which results when a good match exists between the problem representation (i.e., in this case the way data is presented by the BI tool) and the cognitive task (the way data is used) involved in making decisions.

The concept of cognitive fit is supported by research in the field of behavioral decision making which demonstrates that decision makers tend to make better use of information that is explicitly displayed. Moreover, they tend to use it in the form in which it is displayed. Slovic [17] for example points out that "information that is to be stored in memory, inferred from the explicit display,

or transformed tends to be discounted or ignored." Therefore, cognitive fit is enhanced when data is presented in a form that fits well with the processes the decision maker uses to make decisions. This results in lower "cognitive load" (i.e., less manipulation of the data by the user), which facilitates the decision-making process.

In terms of the decision-making process itself, the key impact of a decision model in *goal-directed* systems is improving the probability of goal accomplishment. The "cause-effect" nature of such decisions is related to resource allocation. For example, should a manager invest more in advertising in order to improve revenues or would an investment in training have more of an impact? According to Vessey [18], these types of decisions call for an understanding of associations between variables (i.e., impact of advertising and training on revenue growth). The problem is that in most BI tools, such associations are not defined. Decision-makers need to process the data by estimating whether the cause effect model is correct then estimating the strength of the relationships. According to Popova and Sharpanskykh [13], even when relationships can be defined such as in the ARIS model [5], which allows users to define cause-and-effect relationships using Balanced Scorecards and connect KPIs to strategic goals, the analysis options are inadequate due to a lack of formal modeling foundations and proper representations. The more processing the decision-maker has to do, the higher the cognitive load and the less efficient the decision-making environment. The graphs versus tables literature [6,18] for example, argues that decisions can be improved (i.e., faster and more accurate decisions can be made) when cognitive load is reduced and when values for each of the variables in the model are displayed in their proper context.

This literature suggests that the failure of many BI tools to enhance decision making could be related to the lack of cognitive fit. The underlying data models used by multi-dimensional tools for example, provide data in tabular or graphical format, but these formats do not explicitly identify the variables important to the decision, the relationships between the variables, or the context for the decision itself. Therefore, it seems reasonable to assume that the use of business intelligence tools for decision making can be enhanced if the decision model (i.e., the cause effect relationships among variables relevant to the decision) is displayed by the tool, if the model is linked to the decision's context (in this case, the desired strategic outcomes), and if the associations between the variables can be readily understood.

2.2 Goal-Oriented Requirement Language

Goals are high-level objectives of an enterprise, organization, or system. The requirements engineering community has acknowledged the importance of goal-oriented approaches to system development many years ago. Yu and Mylopoulos [20] observed that goals are important not just for requirements elicitation, but also to relate requirements, processes and solutions to organizational and business contexts, and to enable trade-off analysis and conflict resolution.

Complete goal-driven development approaches now exist to support software development [19].

The Goal-oriented Requirement Language (GRL) is a graphical notation used to model and analyze goals. Although many goal-oriented languages exist, GRL is the first and currently only standardized one. GRL is part of the User Requirements Notation (URN), a standard of the International Telecommunication Union intended for the elicitation, analysis, specification, and validation of requirements using a combination of goal-oriented and scenario-based modeling [9]. In URN, GRL is complemented by a scenario notation called Use Case Maps, which offers an operational or process-oriented view of a system.

GRL enables the modeling of stakeholders, business goals, qualities, alternatives, and rationales. Modeling goals of stakeholders with GRL makes it possible to understand stakeholder intentions as well as problems that ought to be solved. GRL enables business analysts to model strategic goals and concerns using various types of *intentional* elements and relationships, as well as their stakeholders called *actors* (◯). Core intentional elements include *goals* (⬭), *softgoals* (⬭) for qualities, and *tasks* (⬭) for activities and alternative solutions. Intentional elements can also be linked by AND/OR *decompositions*. Elements of a goal model can influence each other through *contributions*, displayed as arrows. Qualitative positive (make, help, some positive) and negative (break, hurt, some negative) contribution levels exist, as well as quantitative contribution levels on a scale going from −100 to +100.

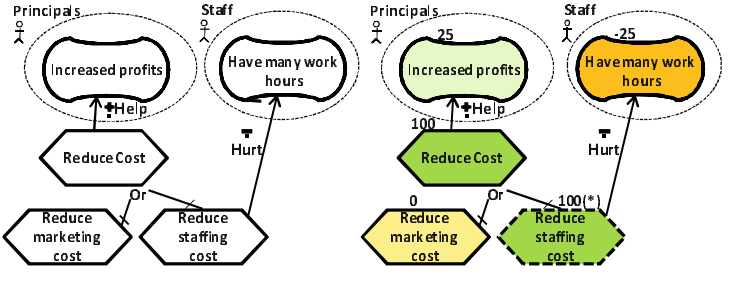


Fig. 1. Example of GRL model (left), with evaluation (right)

Fig. 1 (left) illustrates some of the above concepts with a toy retail store example, where principals (actor) want increased profits (softgoal) and the staff wants to have many work hours. Reducing costs (task), which can help satisfying the principals' objective, can be decomposed into two non-mutually exclusive options: reducing the marketing cost or reducing the staffing budget. The latter option however can hurt the staff's objective. As modelers get deeper knowledge of these relationships, they can move from a qualitative scale (e.g., *Help*) to a quantitative one (e.g., 35) for contributions and for satisfaction values. Such models can help capture stakeholder's objectives as well as their relationships

in an explicit way in terms understandable by managers, and hence improve cognitive fit.

GRL *evaluation strategies* enable modelers to assign initial satisfaction values to some of the intentional elements (usually alternatives at the bottom of a goal graph) and propagate this information to the other elements through the decomposition and contribution links. Strategies act as *what-if* scenarios that can help assess the impact of alternative solutions on high-level goals of the involved stakeholders, evaluate trade-offs during conflicts, and document decision rationales. Different goal evaluation algorithms (using qualitative values, quantitative satisfaction values between −100 and +100, or mix of both types) for GRL are discussed in [1].

jUCMNav is an open source URN tool for the creation, analysis, and management of URN models [12]. It allows for the qualitative, quantitative, or hybrid evaluation of GRL models based on strategies. To improve scalability and consistency, jUCMNav also supports the use of multiple diagrams that refer to the same model elements.

Fig. 1 (right) illustrates the result of a strategy for our example where the reduction of the staffing budget is selected, i.e., the satisfaction value of this task is initialized to 100. In jUCMNav, initialized elements are displayed with dashed contours. This strategy eventually leads to a weakly satisfied (+25) "Increased profits" softgoal and to a weakly denied (-25) satisfaction level for "Have many work hours". The resulting satisfaction values in top-level goals and actors should be used to compare different strategies and find suitable trade-offs rather than be interpreted as some sort of satisfaction percentage. jUCMNav also uses a color-coding scheme to highlight satisfaction levels, i.e., red for denied, yellow for neutral, and green for satisfied (with various shades for values in between), which again improves intuitive understanding.

2.3 GRL and KPI for Business Modeling

Although the primary application domains for URN target reactive systems and telecommunications systems, this language has also been applied successfully to the modeling and analysis of business goals and processes [21]. Goal models combined to process models have been used elsewhere to assess the risk and viability of business solutions [2] and model different concerns of interest to different stakeholders [4]. However, in order to better support business process monitoring and performance management, Pourshahid *et al.* [14] have extended standard GRL with the concept of *Key Performance Indicators* (⬭). KPIs can also be analyzed from various angles called *dimensions* (⬚), in a way similar to what is found in common BI systems. Dimensional data allows one to look at the data from different points of view and filter or aggregate the data based on the defined dimensions. For instance, in Fig. 2, staffing cost can be aggregated in all locations in all years of store operations or can be analysed for Store1, 2, 3 and the online store individually and in a specific month or year.

KPIs include specifications of *worst*, *threshold*, and *target* values in a particular *unit*. For example, a Staffing cost KPI (see Fig. 2) could have a target value

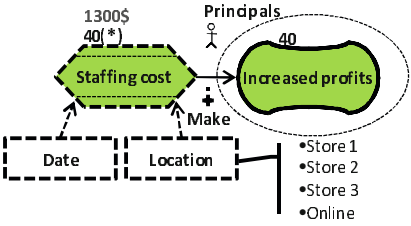


Fig. 2. Example of a KPI with dimensions and evaluation

of $1000, a threshold value of $1,500, and a worst value of $2,500. KPIs also contain a *current* value, which is either defined in a GRL evaluation strategy or provided by an external source of information such as a database, an Excel sheet, a BI tool, external sensors, or Web services. The KPI is a metrics of the system that normalizes the current value to a scale of −100 to 100, which enables it to be used like any other intentional element in a GRL model. For instance, if the current Staffing Cost is $1300, then the normalization function, which takes here $|threshold - current|/|threshold - target| * 100$, will result in a satisfaction level of 40. Furthermore, when the current value is between the threshold value and the worst value (e.g., 2500), then the normalization function becomes $|threshold - current|/|worst - threshold| * (-100)$, which results in a negative value (e.g., −100). If the result is higher than 100, then it becomes 100 (symmetrically, if it is lower than −100, then it becomes −100). Such an evaluation strategy was used in Fig. 2. Note also in this model that Staffing cost could be drilled down (e.g., explored) according to the Date and Location dimensions.

Although goal modeling and scorecards have been used in combination in the past [3,15], we believe KPIs are also necessary because they act as an interface between conventional goal models and quantitative, external sources of information.

Furthermore, Pourshahid *et al.* [14] have introduced and implemented a service-oriented architecture enabling the use of underlying data and BI reports by the jUCMNav tool. jUCMNav is connected to BI systems via a Web service. All the information generated by the BI system, from raw data to very complex data warehouses, can hence be used as qualitative data to initialize the KPIs used in the GRL model, and against which goal satisfaction is evaluated.

Although several other goal modeling languages exist (e.g., *i**, Tropos, KAOS, and the Extended Enterprise Modeling Language), the combination of support for KPIs and performance modeling, the ability to combine process and goal models and perform analysis on both, existing tool support for using BI systems as sources of data, and the fact that URN is a standard modeling language, all together have convinced us that URN is the best language to be used in the context of our research.

3 New Formula-Based Evaluation Algorithm

Although several GRL evaluation algorithms (qualitative, quantitative and hybrid) already exist [1], none of them provides the formula-based KPI aggregation required for the type of cause-effect analysis performed in our decision making context. As illustrated in the previous section, the current algorithms allow modelers to specify the contribution level of a KPI on another GRL intentional element and to calculate the satisfaction level of that target element [14]. However, these algorithms prevent one KPI from driving the computation of the *current value* of another KPI. Although the current evaluation methods allow computing the impact of one KPI on another KPI in terms of *satisfaction level*, when it comes to showing the impact of several KPIs on one KPI (e.g., their *aggregate* effect), the current evaluation methods quickly become a bottleneck and thus obstruct the cause-effect analysis.

Other modeling languages and enterprise modeling frameworks exist that can be used to model KPIs, however many have a limited computational power and do not allow one to define proper relationships between KPIs for advanced analysis [13]. In addition, there have been recent efforts in industry to use strategy maps and measurable objectives to help with decision making and process improvement [16]. However, influence of KPIs on one another has not been discussed.

In order to address this issue, we introduce further extensions to GRL and a novel evaluation algorithm that allow analysts and decision makers to define mathematical formulae describing relationships between the model elements. This method, which extends the bottom-up propagation algorithm defined in [1], enables the precise definition of accurate relationships between these elements. Analysts gain full control of the model and can change the impact of one element on another as desired.

The algorithm uses current/evaluation values of the source KPIs as inputs for the formula (described as metadata, see Fig. 3) and calculates the target KPI evaluation value using these inputs. Then, the satisfaction level of the KPI is calculated using the KPI's target/threshold/worst values as discussed previously. The impact of KPIs on other types of intentional elements (e.g., goals, softgoals and tasks) is computed using conventional GRL quantitative and qualitative algorithms. This unique combination allows one to have both quantifiable KPIs and strategic-level softgoals that are hard to quantify together in the same model and to show and monitor the impact of KPIs on the goals of the organization.

Fig. 3 shows a simple example where the current KPI values are displayed, with their units, above the usual satisfaction values. Note that the inputs can be of different units; the formula in the target KPI must take this into consideration. In this example, the current value of $Profit$ is computed as $Revenue - Costs - Stolen * 50$ (the first two are in dollars and the third is a number of items). Note also that the contributions have no weight; the satisfaction of the $Profit$ KPI is based on the normalization of its computed current value ($39,000) against its specified target, threshold and worse values. We have prototyped this new algorithm in the jUCMNav tool.

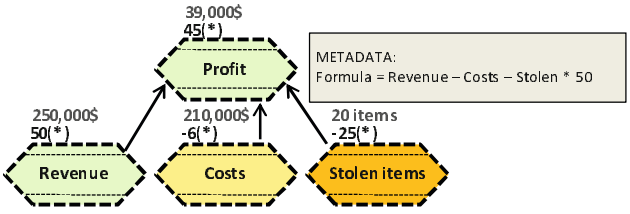


Fig. 3. New extension to KPI evaluation

Another benefit of this approach is the ability to account for risk. In organizations, cause-effect analysis and decision making usually involve an element of *risk*. Even though we could show risks as model elements in GRL diagrams (e.g., using softgoals stereotyped with «Risk»), it is very hard to quantify the impact of risks on the value of a KPI and consider it in the evaluation algorithm. In our new algorithm, we use risk as yet another input to the target KPI and connect it using a contribution link. The target KPI *threshold value* changes based on the level of contribution of the risk factor on the target element. This allows the modeler to vary the acceptable range of values for a KPI when there is expected risk involved.

4 Business Intelligence Decision-Making Framework

Based on the reasoning behind the notion of cognitive fit, the framework we are proposing defines the organizational goals and links these explicitly to a decision model and relevant Key Performance Indicators. The framework can be used by organizations at any level of maturity and readiness in terms of gathering and monitoring data for BI-based decision making. In particular, unlike many simulation approaches, it does not necessitate up front large quantities of data to be useful. We believe different organizations however have different needs and may be in different states when they decide to incorporate such a framework. Consequently, we are suggesting a spiral method consisting of three basic steps involving many iterations that build upon each other (Fig. 4).

In the **first step**, an initial model of the organization's goals is created [7]. This model can be built based on interviews with executives and operational managers as we experimented with in our example. This goal model can consist of long term, short term, strategic and operational goals of the organization as well as contribution and decomposition relationships between them. Furthermore, in this step we define *the KPIs that support the goals* (e.g., financial KPIs) and add them to the model. This can be a challenging task and is very dependent on the level of maturity of the organization. For instance, in two cases we have studied as part of this research, one small organization had a very limited set of data and was using a spreadsheet to monitor the business while the other one had many indicators available and was using a sophisticated Business Intelligence system. Our discussions with both organizations however demonstrate that any

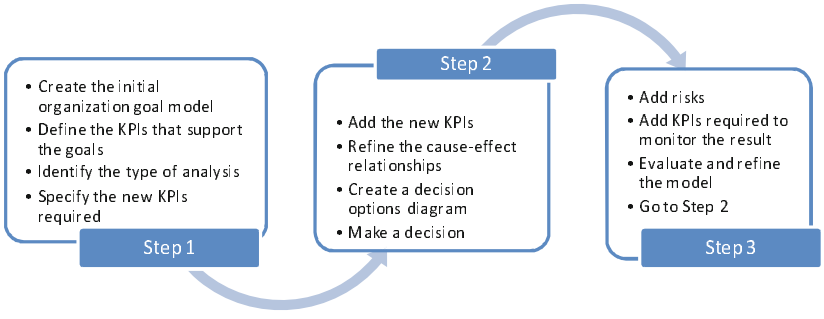


Fig. 4. Business Intelligence Decision-Making Framework

organization at any point within this wide range of information management capabilities can benefit from applying this goal-based model. After defining the model, we *identify the type of analysis* we want to perform on the model and *specify the new KPIs required* to do so.

In the **second step**, we *add the new KPIs* and the new dimensions to the model. Note that not all the KPIs need to be dimensional and if the available data is not as granular as is required for a dimensional model, or if all the data is not available, a step-by-step approach can be used leading to a number of model iterations as additional data becomes available.

In addition, during this process we *refine the cause-effect relationships* between KPIs in the goal model (hence improving cognitive fit). These relationships create a BI-enabled decision framework which can be used to document the rationale for goal accomplishment, the decision context, and to analyze what-if scenarios. The framework also helps one to evaluate the impact of a decision on the enterprise's goals through the use of historical and trend data.

In cases where an organization does not have historical data and is in its early iterations of BI-based decision making, the initial formula used to define the decision framework can be based on industry standards. As the organization gathers more information, this historical data can be integrated to the model. As will be seen later in the retail example, a decision framework can be used to illustrate the expected impact of actions taken by managers. Furthermore, they can be continually adapted by saving the initial iteration as a "snap shot" and comparing it to actual results achieved by decisions. Gathering these snap shots will eventually create a "decision trail" that displays context, decisions taken and results of these decisions allowing managers to make better decisions in the future. In addition, decision trails allow organizations to refer back to the rationale they used for making successful or unsuccessful decisions.

We also add a *decision options diagram* (in the same GRL model) and connect these options to the goals and KPIs of the organization. A decision options diagram outlines the different options available to an organization to achieve a goal.

In the **third step**, we add the expected impact of the decision made in the second step to the model and *include risks* involved in the decision. In using GRL softgoals, we are able to show qualitatively the impact of risks on the rest of the model. The most challenging aspect of this step is modeling a qualitative risk factor that influences a quantitative KPI. In this case, we model the impact by increasing the range of acceptable values for a KPI. In other words, once we have estimated the inherent risk, we allow the acceptable range of the measured KPI to deviate accordingly from its target value.

In this step, we also add the *required KPIs* and dimensions to the model that allow one to better observe the impact of decisions. If we expect a decision to change anything in the organization, we will examine that hypothesis using the appropriate KPIs and GRL strategies. Finally, we *monitor* the impact of the decision and compare expected results against actual results. Based on this comparison, we *adjust* the decision framework as required and record the data.

In summary, the iterative cycle is based on creating an initial model which is then refined by expanding data sources, capturing decisions made and the results of those decisions, and building historical decision trails that informs future models.

5 Retail Business Real-Life Example

In this example, we describe an initial application of the framework to a real, Ontario-based retail business. The retail business would be categorized as a small enterprise (revenues less than $50 million) with 4 local stores and planned expansion nationally. The business has existed for over 15 years, establishing a strong foothold in one neighborhood. Three years prior to the study, the business was purchased by new owners who set national growth as a key strategic objective. As part of the expansion plans, market and competitive studies were conducted. In addition, the owners had created a scorecard that tracked key operational indicators and provided the ability to conduct an assessment of business results. Some data however, for example, the flow of customers through each of the locations, was not yet available in the scorecard.

At the time of the study, most revenues were earned through consignment sales. The business however had started selling new items as well and was planning to invest in an online business. Revenue was driven by ensuring that stock was properly displayed which in turn depended on assuring that enough staff were available to sort, tag, and lay out the products. The supply side of the business depended on the number of consigners available, the amount of product they brought to each store, and the speed at which these products could be displayed. The demand side depended on local advertising and word of mouth that stimulated traffic flow. All stores were situated in prominent locations with good visibility that stimulated walk-in traffic.

To begin our investigation, we interviewed the CEO in order to identify the high-level goals as well as KPIs and drivers of organizational success. As depicted in Fig. 5, the goals of the principals were related to market growth: they wanted

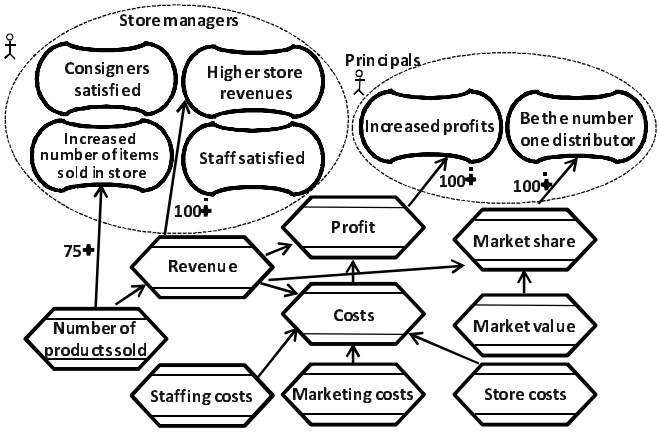


Fig. 5. First iteration model (aggregation formulas not shown here for simplicity)

to be the number one distributor within their geographical market. Store managers were aware of the growth objective, but on the short term, they focused on increasing revenues and the number of items sold in their stores.

As discussed above, the first iteration of the model provides an initial alignment of higher-level goals and KPIs. We started with a minimal decision model and limited set of data just to illustrate the business goals and financial targets and to identify the indicators and driver KPIs required to monitor the business and to make informed business decisions. Fig. 5 illustrates the first iteration of the model. At this stage, we also developed a rough dimensional model (Fig. 6) in order to ensure that the data needed for the decision model would be available. The dimensional model helps the store to analyze the impact of the KPIs on goals based on their different store locations, in each period of time, by product type (e.g., clothing, electronics, etc.), and by product category (i.e., retail and consignment).

In Step 2 of the process, more KPIs were added to the model as drivers and then linked to the high-level financial KPIs and organizational goals. In addition, we added the new KPIs as well as a new dimension called "marketing type" (e.g., outreach, online advertising, etc.) to the dimensional model. This new dimension allows decision makers to analyze which marketing initiative has

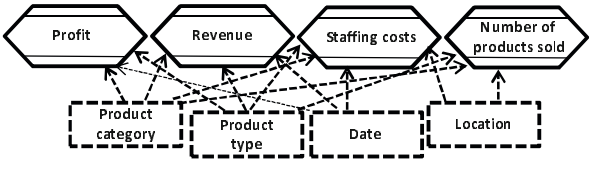


Fig. 6. First iteration dimensions

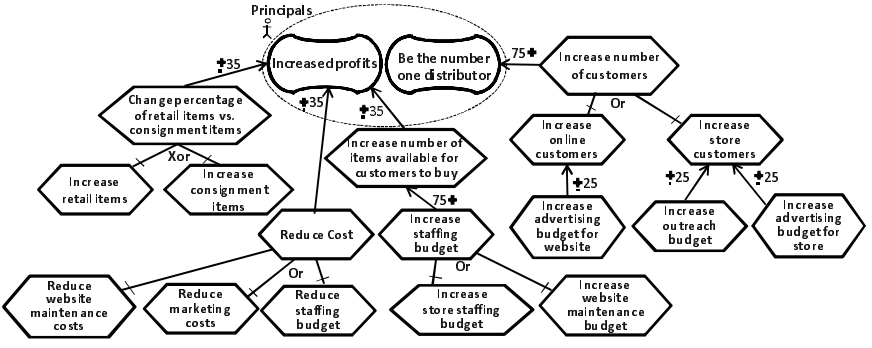


Fig. 7. Second iteration decision options diagram

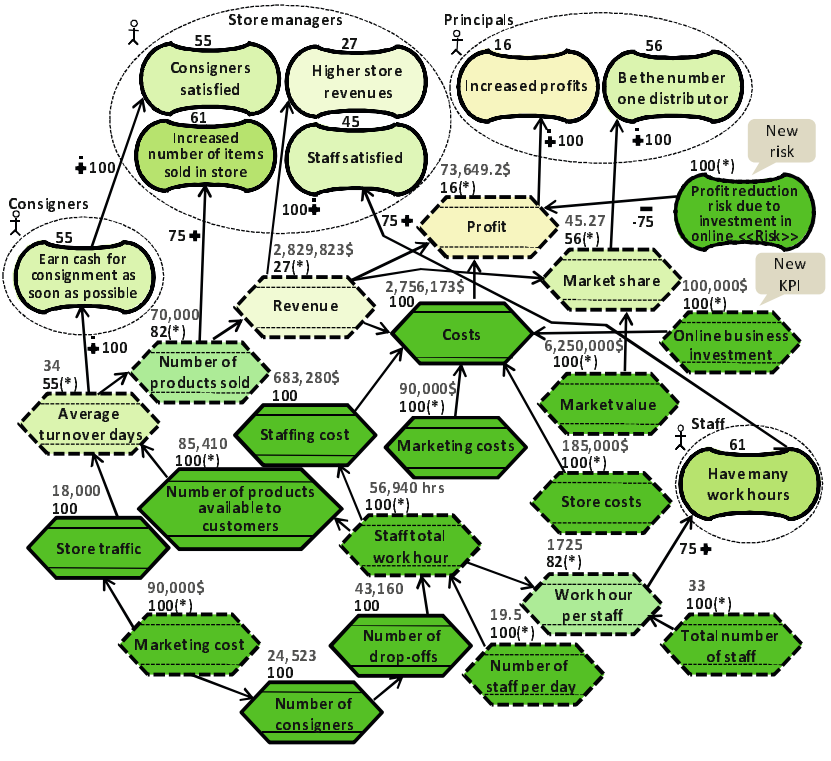


Fig. 8. Third iteration model – evaluated

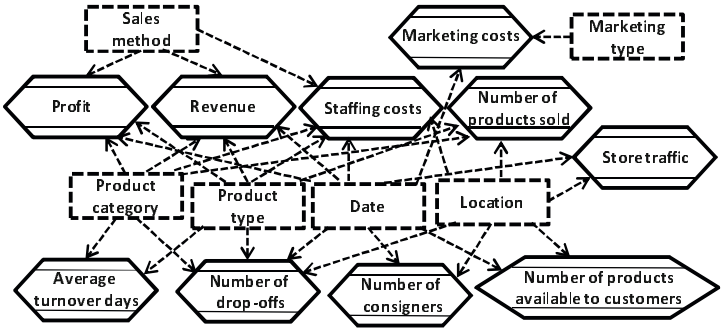


Fig. 9. Third iteration dimensions

a more significant impact on the goals. In this step we also created a decision options diagram (Fig. 7) illustrating the specific actions managers can take to improve goal accomplishment. One of the decision options available to managers in this case, is to invest in an online business. We consider this option as the decision made by managers and update the models accordingly in step 3.

Fig. 8 depicts the complete model, which defines the expected impact of the actions identified in Fig. 7 along with the KPIs and acceptable ranges for each of the relevant goals based on the risks associated with each goal. There is one new risk factor in the model that is associated with the investment in the online business. Furthermore, there is also a new KPI used to monitor the investment made in the online business and its impact on the costs. The GRL strategy used for the evaluation here focuses on the use of the online business investment (other GRL strategies were defined to evaluate different sets of options and find the most suitable trade-off). Fig. 9 depicts the final dimensional model (including its new "Sales method" dimension) used to ensure that the relevant data can be delivered to decision makers. Note that, in jUCMNav, such figures can be split over many diagrams when they become complex.

6 Lessons Learned

The development of our framework and its application to the real world retail business data led to several lessons learned. From a business management perspective, we observe the following:

– Modeling goals and defining drivers and KPIs (i.e., creating the cause-effect decision model) not only helps to document the known aspects of the business but also helps to clarify unknown factors that might be driving goal accomplishment. Validation of the model through interviews with decision makers ensures that data and KPIs included are indeed relevant to the business. This can have a great impact especially for small businesses where initial goals might not have been clear.

- Even though modeling the indicators helps define the required information and the relationships between variables, we are still unsure about where to draw the line regarding the data we need to show in the model versus the data maintained in source systems (e.g., databases or BI reports). We still need to explore how to find the appropriate balance so we do not omit important information in the model for decision making while preventing the inclusion of too much data which can complicate the decision-making environment. We believe however that getting feedback on the right balance is facilitated by the use of graphical goal models with rapid evaluation feedback as provided by GRL strategies, which provide better cognitive fit than conventional BI reports. Note however that the goal-oriented view introduced here is complementary to what is found in BI tools, not a substitute.
- Defining relationships between the model elements without historical data is difficult. In some cases, managers themselves are not aware of the linkages because they have not had the historical data available to create cause-effect models. In this case, we first create the models using industry standards or "best guesses" and then use the different iterations of the framework to improve the cause-effect decision model.
- The ability to adjust the range of acceptable values for a KPI is useful for registering risk. For example, one might establish a wide range of acceptable values for an objective that carries a high level of risk, such as expected sales for a new product. On the other hand, objectives with lower profiles, such as sales of well-established products, might have a narrower range of acceptable values.

From a technical point of view, we have learned that:

- Our new extensions to GRL and the new formula-based algorithm provide a great deal of flexibility for model evaluation, especially as they are combined with standard goal satisfaction evaluation, hence offering the best of both worlds. However our new algorithm still has room for improvement, especially when it comes to using other intentional elements (e.g., goals) as contributors to KPIs. We have had limited experience with this idea by considering risk as an input to KPIs, but this type of modeling may be useful in other situations that require further investigation.
- Creating different versions of a model in different iterations and keeping them consistent for comparison purposes can be painful with current tool support. Saving separate files for each version of the model quickly becomes a maintenance issue that requires a better technical solution.

7 Conclusions

There are critical issues related to the use of conventional Business Intelligence technology for decision making. The gap between the technical data model and the decision model creates a lack of cognitive fit, especially for supporting cause-effect decisions. This represents a challenge that will become even more

important in the future given that organizations are nowadays gathering terabytes of information.

In this paper, we have provided several contributions toward a goal-oriented business intelligence decision framework, where we integrate in a novel way goal models, decision frameworks, action models, and risk, together with analysis capabilities. By integrating the decision framework into the BI system, we attempt to improve cognitive fit between the decision making task and the representation of information needed to complete the task. To do so, we extended a standard goal-oriented language, GRL, to better display relationships between Key Performance Indicators and objectives, enable formula-based evaluations of goal models, and integrate risk through the notion of acceptable ranges for KPIs. These extensions enable the combination of quantifiable KPIs with strategic-level softgoals in the same model, which in turn allows analysts to assess and monitor the impact of KPIs based on existing values and to explore what-if scenarios through GRL strategies. Tool support is provided as extensions to the open source jUCMNav tool.

From an implementation perspective, we also introduced a framework with iterative steps that support the construction of goal models (including KPIs and dimensions) even in situations where little information is available. Such models can be refined as more knowledge is gained about the organization and its context. Models can also be compared (as historical data) to validate newer models to assess the impact of past business decisions, leading to an ongoing system of record that permits continual adaptation. Our retail business example helped illustrate the framework in a real context, and suggests the feasibility of the approach. We also believe that such a graphical, goal-oriented approach, which delivers data values used to make decisions in context, supports the comprehension of important cause-effect relationships in a way that could complement existing current BI technologies, which often lack an appropriate goal view.

The framework is still evolving, and limitations and potential work items have been identified in our lessons learned. However, this framework brings new contributions and good value to the BI table, and we believe it has a promising future.

Acknowledgments. This research was supported by the Business Intelligence Network (BIN), a strategic network funded by the Natural Sciences and Engineering Research Council of Canada.

References

1. Amyot, D., Ghanavati, S., Horkoff, J., Mussbacher, G., Peyton, L., Yu, E.: Evaluating Goal Models within the Goal-oriented Requirement Language. International Journal of Intelligent Systems (IJIS) 25(8), 841–877 (2010)
2. Ansar, Y., Giorgini, P., Ciancarini, P., Moretti, R., Sebastianis, M., Zannone, N.: Evaluation of Business Solutions in Manufacturing Enterprises. Int. J. Bus. Intell. Data Min. 3(3), 305–329 (2008)

3. Babar, A., Zowghi, D., Chew, E.: Using Goals to Model Strategy Map for Business IT Alignment. In: BUSITAL 2010, CEUR-WS, vol. 599, pp. 16–30 (2010)
4. Colombo, E., Mylopoulos, J.: A Multi-perspective Framework for Organizational Patterns. In: Embley, D.W., Olivé, A., Ram, S. (eds.) ER 2006. LNCS, vol. 4215, pp. 451–467. Springer, Heidelberg (2006)
5. Davis, R., Brabänder, E.: ARIS Design Platform: Getting Started with BPM. Springer, Heidelberg (2007)
6. De Sanctis, G.: Computer graphics as decision aids: Directions for research. Decision Sciences 15, 463–487 (1984)
7. Feng, X., Richards, G., Raheemi, B.: The Road to Decision-Centric Business Intelligence. In: 2nd Int. Conf. on Business Intelligence and Financial Engineering, Beijing (2009)
8. Hackathorn, R.: Making Business Intelligence Actionable. DM Review 32 (2002)
9. International Telecommunication Union: Recommendation Z.151 (11/08), User Requirements Notation (URN) – Language definition (2008), http://www.itu.int/rec/T-REC-Z.151/en
10. Ko, I.S., Abdullaev, S.R.: A Study on the Aspects of Successful Business Intelligence System Development. In: Shi, Y., van Albada, G.D., Dongarra, J., Sloot, P.M.A. (eds.) ICCS 2007. LNCS, vol. 4490, pp. 729–732. Springer, Heidelberg (2007)
11. Korhonen, P., Mano, H., Stenfors, S., Wallenius, J.: Inherent Biases in Decision Support Systems: The Influence of Optimistic and Pessimistic DSS on Choice, Affect, and Attitudes. Journal of Behavioral Decision Making 21, 45–58 (2008)
12. Mussbacher, G., Ghanavati, S., Amyot, D.: Modeling and Analysis of URN Goals and Scenarios with jUCMNav. In: 17th IEEE Int. Requirements Eng. Conf., pp. 383–384. IEEE CS, USA (2009), http://jucmnav.softwareengineering.ca/jucmnav/
13. Popova, V., Sharpanskykh, A.: Modeling organizational performance indicators. Information Systems, Vocabularies, Ontologies and Rules for Enterprise and Business Process Modeling and Management 35(4), 505–527 (2009)
14. Pourshahid, A., Chen, P., Amyot, D., Forster, A.J., Ghanavati, S., Peyton, L., Weiss, M.: Business Process Management with the User Requirements Notation. Electronic Commerce Research 9(4), 269–316 (2009)
15. Siena, A., Bonetti, A., Giorgini, P.: Balanced Goalcards: Combining Balanced Scorecards and Goal Analysis. In: Third Int. Conf. on Evaluation of Novel Approaches to Software Engineering (ENASE 2008), Funchal, Portugal (2008)
16. Silver, B.: Collaborative Business Process Discovery and Improvement. Industry Trend Reports, Bruce Silver Associates (May 2010)
17. Slovic, P.: Psychological Study of Human Judgment: Implications for Investment Decision Making. The Journal of Finance 27(4), 779–799 (1972)
18. Vessey, I.: Cognitive Fit: A Theory-Based Analysis of the Graphs versus Tables Literature. Decision Sciences 22(2), 219–240 (1991)
19. van Lamsweerde, A.: Requirements Engineering: From System Goals to UML Models to Software Specifications. John Wiley & Sons, Chichester (2009)
20. Yu, E., Mylopoulos, J.: Why Goal-Oriented Requirements Engineering. In: Fourth Intl. Workshop on Req. Eng.: Foundation for Software Quality (REFSQ 1998), Pisa, Italy (1998)
21. Weiss, M., Amyot, D.: Business Process Modeling with URN. International Journal of E-Business Research 1(3), 63–90 (2005)

Softw*I*re Integration – An Onto-Neural Perspective

Hendrik Ludolph[1], Peter Kropf[1], and Gilbert Babin[2]

[1] University of Neuchâtel, Institute of Computer Science,
2009 Neuchâtel, Switzerland
{hendrik.ludolph,peter.kropf}@unine.ch
[2] HEC Montral, Information Technologies,
3000, ch. Cte-Ste-Catherine, Montréal (QC), Canada H3T 2A7
gilbert.babin@hec.ca

Abstract. We propose a framework for automated point-to-point application integration. Functional and technical aspects to consider are presented. We advocate using intelligible (ontological) and distributed (neural networks) knowledge representations to guide design and implementation of application interfaces. We believe this yields benefits over manual approaches currently used, in which business specifications are first captured, then translated into technical specifications, and finally implemented, often by costly third-party integration specialists. The present work initiates research efforts towards automation of the latter two activities, with a potential impact on professional-services costs and time-to-operation.

Keywords: ontology matching, neural network, application integration.

1 IT Landscapes and Challenges

Generally, corporate IT organizations use a wide range of software applications to maintain IT operations. For example, such applications are deployed for:

- managing (e.g., discovering, inventorying) network attached devices, such as laptops, servers, or workstations; or
- managing (e.g., storing, tracking) configuration items[1] and relationships among them, or
- managing (e.g., logging, dispatching) IT-related incidents and problems.

Often, the activities are part of good-practice standards for IT operations, e.g., *Infrastructure Information Library* (ITIL) [1], *IT Asset Management* (ITAM) [2], or *Control Objectives for Information and related Technology* (COBIT) [3].

[1] A configuration item is an asset, service component or other item controlled by the configuration management process, stored within the configuration management database [1, p.112].

G. Babin, K. Stanoevska-Slabeva, P. Kropf (Eds.): MCETECH 2011, LNBIP 78, pp. 116–130, 2011.

Software vendors align their product portfolio to meet these standards, seeking economies of scale. Consequently, applications often follow an "off-the-shelf" approach. By the same token, companies adopt IT standards and expect lower costs through "out-of-the-box" implementations. For example, ITIL defines a process "Service Catalog Management," with associated tasks, such as providing, updating, deleting. A variety of service catalog management applications are offered within the market, which are tailored to support these standardized tasks. However, they may vary as to how functionality is provided or implemented. Furthermore, the service catalog management process is linked to other management processes, e.g., change, configuration, or IT financial management[2]. Again, a company may choose from a considerable number of applications tailored for supporting these processes.

Eventually, the adoption of IT standards leads to application pools working in concert within corporate IT landscapes. Within such application pools, a given application is connected to one or more other applications with respective data input and output interfaces[3].

Despite the above standardization efforts, such as ITIL, important challenging issues still can be identified:

- IT architectures evolve due to changing business needs and the need for IT-business alignment.
- Software providers seek and succeed to replace competitors' products. This may or may not lead to improvements. It however leads to changes within the application pool.
- Data quality (e.g., global naming conventions) is a difficult task for companies [5]. Decentralized setup often entails increasing syntactic divergence not accounted for within existing application interfaces.
- Applications are upgraded to more reliable versions.

In each case, continued supervision, manual design, and adjustments of application interfaces are needed.

With these challenging issues in mind, the remainder of the document introduces an approach towards the automatic setup of interfaces between applications. In Section 2, we outline our rationale and give an overview of the basic concepts used, namely ontologies and neural networks. We also discuss scope and limits of the proposed approach. In Section 3, we present methodological and functional aspects. Section 4 further details these aspects and depicts underlying technical design principles. Practical examples are given for each step of the method to clarify the approach. Note that the paper presents research in progress. Experimental results are currently being gathered and will be presented and discussed later on.

[2] For more details, the interested reader is referred to respective sources on ITIL.
[3] A point-to-point connection [4] is emphasized (cf., chapter 3).

2 The Onto-Neural Approach to Integration

Our work focuses on automated integration of applications considering the constant changes which pertain to "off-the-shelf" applications in standards-driven parts of corporate IT landscapes, while assuming that these applications are relatively stable. In other words, changes in functionality and data model are gradual and aligned to the evolution of IT standards.

We claim that application integration can be accomplished through low-level attribute mapping scripts which are automatically constructed or adapted by using high-level business specifications. This approach can then replace the typical one which consists in manually translating those business specifications into technical requirements and subsequently implementing these requirements.

To achieve this goal, we explore techniques inspired by *neural-symbolic integration* (cf., [6,7]) and formal concept description and reasoning systems, namely *ontologies* (cf., [8]).

Ontologies capture conceptual views of a domain of interest and express complex types of entities and relations among them [9]. Through automatic reasoners, they also allow for plausibility checks and inference of implicit knowledge. We investigate how these two characteristics can be used as the basis to encode high-level business specifications.

It should be noted however that formal inference based on such ontologies typically relies on unique naming in order to connect facts. In other words, ontology-based systems assume that a single, well-formed ontology applies throughout [10]. It is our understanding that end-to-end corporate IT does not currently respect this assumption. We therefore intend to use neural networks to formalize the notion of distributed knowledge. Neural networks are made up of a connected network of individual computing elements (mimicking neurons). Often, they are deployed for classification tasks [11]. Specifically, we explore how neural networks can be used to obtain a fixed-length distributed representations of structured data such as graphs. We will use these representations to find pairs of applications suited for integration.

Within the scope of this paper we aim at providing a mere proof of concept. We entirely focus on the goal of constructing adequate mapping scripts by technological means, that is by ontologies and neural networks. We do not discuss secondary aspects to integration, such as performance, scheduling, triggers, or else IT as politically (as opposed to technically) driven domain.

3 Integration Method: Functional Design

In the following sections, we discuss methodological aspects. These aspects are combined within the subsequent technical design principles and address standardized IT processes automated by several independent applications.

The approach we introduce is constructed around the following concepts:

- **Ontology:** is a rigorous and exhaustive organization of some knowledge domain that is usually hierarchical and contains all the relevant entities and their relations[4].
- **Application:** is a software program that gives a computer instructions[4]. It provides the user with functionality to accomplish a task. It relies on a set of data objects comprised within a data model.
- **Application integration:** is a point-to-point communication directly between individual IT applications to enable collaborative business processes [4]. Thereby, the applications' underlying database attributes are purposefully linked to each other.
- **Process:** is a goal-oriented collection of activities that take one or more kinds of input and create an output[5]. Activities are supported by applications which provide necessary functionality per activity.

In light of these concepts, the integration method proceeds as follows:

Step 1: An ontological representation for causally related activities, that is a process, is created. Within the resulting process ontology, any pair of consecutive activities is thus identifiable.

Step 2: For each relevant business application that can implement activities within the process, an application ontology is created. The resulting ontologies encode functionality and data model. More than one application per activity might exist. Consequently, several independent ontologies describing one activity might exist.

Step 3: The process ontology resulting from Step 1 is used as context for coupling (and *not* matching) application ontologies for any selected pair of consecutive activities. The resulting *compound* ontology fulfills the axiomatic restrictions of the process ontology for the selected pair of activities. As mentioned, several application ontologies for the same activity can exist. As a result, several compound ontologies can exist.

Step 4: In order to compare them, the process ontology and the application ontologies are transformed into neural networks.

Step 5: A similarity measure (e.g., Euclidean vector distance) is computed between the neural networks resulting from Step 4. For further processing, we select the compound ontology whose "compound pattern" is the most similar to the "process pattern."

Step 6: An integration script for the selected compound ontology is constructed.

4 Integration Framework: Technical Design

The integration method introduced in Section 3 uses ontologies and neural networks in order to obtain an integration script. In this section, we describe in more

[4] Definitions from *Wordnet*, http://wordnetweb.princeton.edu/perl/webwn

[5] Definition of business process from *Credit Research Foundation*, http://www.crfonline.org/orc/glossary/b.html

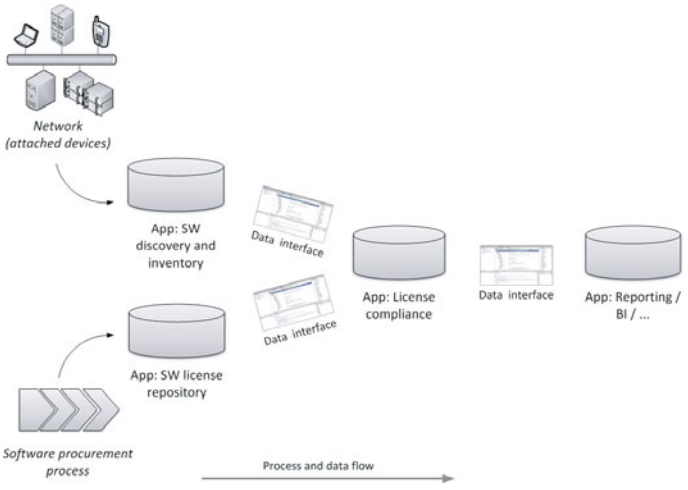


Fig. 1. The SAM process

details how ontologies and neural networks are used, and how the integration script is constructed. As recurring example, we use the simple *Software Asset Management* (SAM) process (cf., Figure 1). Therein, *discovery and inventory* is used to capture and classify software installed on network-attached devices. In parallel, the company's *procurement* stores and tracks purchased licenses. Both sets of data are then compared in order to establish *license compliance*. The results are further processed, such as for management reports. The functions are supported by independent applications offered by competing software vendors. Experience shows that companies do generally not use sophisticated middleware to implement standardized SAM. Rather, point-to-point application interfaces are used.

4.1 Ontology Building

Ontologies represent knowledge capturing and logic reasoning systems. They provide explanations for conclusions, or enrich explicit axiomatic assertions by providing a precise notion of logical consequence through deductive inference [8]. In order to fully exploit them for application integration, we define the following building blocks:

Process ontology (Step 1). The process ontology describes consecutive causal activities which are conceptual and idealistic. Completeness of the representation depends on the company's domain expert who encodes these activities.

Examples of typical representations are given below: the SAM process is restated as a UML activity diagram in Figure 2, followed by a verbose OWL-DL process annotation (stemming from [12]) in Figure 3, and a simple ontology in Figure 4[6].

[6] Axiomatic restrictions, such as quantification, are omitted.

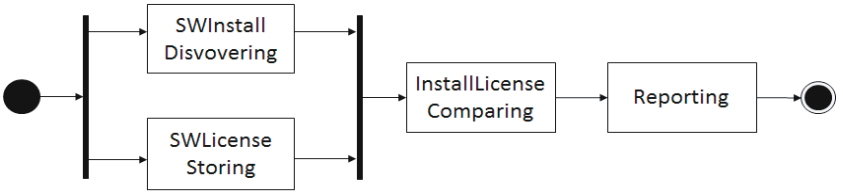


Fig. 2. The SAM process – UML activity diagram

$$\begin{aligned}\texttt{Process} \equiv\ & \texttt{Start} \sqcap \exists \texttt{followedBy.(SWInstallDiscovering} \sqcap \\ & \exists \texttt{followedBy.(InstallLicenseComparing} \sqcap \\ & \exists \texttt{followedBy.(Reporting} \sqcap \exists \texttt{followedBy.End))) } \sqcap \\ & \exists \texttt{followedBy.(SWLicenseStoring} \sqcap \\ & \exists \texttt{followedBy.(InstallLicenseComparing} \sqcap \\ & \exists \texttt{followedBy.(Reporting} \sqcap \exists \texttt{followedBy.End)))}\end{aligned}$$

Fig. 3. SAM process in DL

Henceforward, we will concentrate on two consecutive activities, namely *software installation discovery* and *comparison*. They are used to illustrate the construction of one interface.

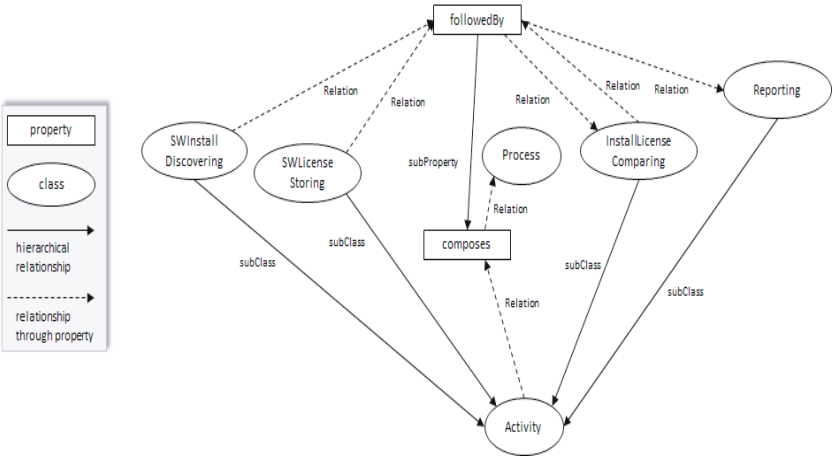


Fig. 4. The SAM process ontology

Application ontology (Step 2). The application ontology describes the functionality supported by the application and the data model used to implement that functionality. In this context, we refer to functionality as the set of functions offered to the user, as implemented by the application.

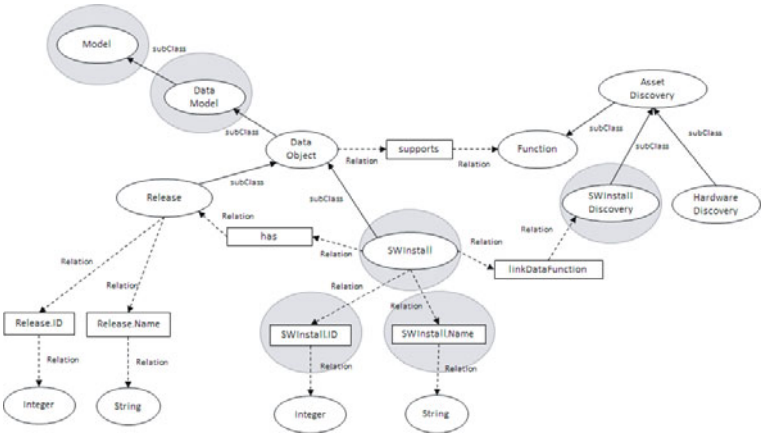


Fig. 5. The *disc*1 application ontology

Going back to the SAM example, let us consider two types of application: asset discovery tools[7], which implement the software installation discovery activity, and IT asset management (ITAM) tools[8], which implement the software license installation comparison activity.

Let us assume that two applications are available that provide the functionality required to support the software discovery activity, namely tools *disc1* and *disc2*. They are provided by two software vendors. Figures 5 and 6 show two different ontological representations. These ontologies are slightly different (grey shaded area) as are the underlying applications. Differences can be found within the functional description and the data model. For the sake of simplicity, we assume furthermore that one application provides the functionality to support the software license and installation comparison activity, namely *itam*. Figure 7 shows its application ontology.

In all figures, the property *supports* relates data model and functionality. The property *linkDataFunction* relates specific data objects to specific functions.

Ontology coupling (Step 3). Given two activities that need to be connected, and two applications providing the functionality supporting these activities (one application per activity), the goal of ontology coupling is to "integrate" the corresponding application ontologies, constrained by the process ontology.

At the heart of the coupling process is the sequence ontology (cf., Fig. 8) which describes the ontology of a generic activity sequence. It will serve two purposes within the coupling process. First, it will be used to construct a "reference" ontology describing the business specifications. Second, it will serve as backbone of the compound ontology describing how applications implement the reference ontology. The coupling process proceeds as follows:

[7] e.g., HP DDMI, CA Infrastructure Management.
[8] e.g., HP Asset Manager, CA Software Compliance Manager.

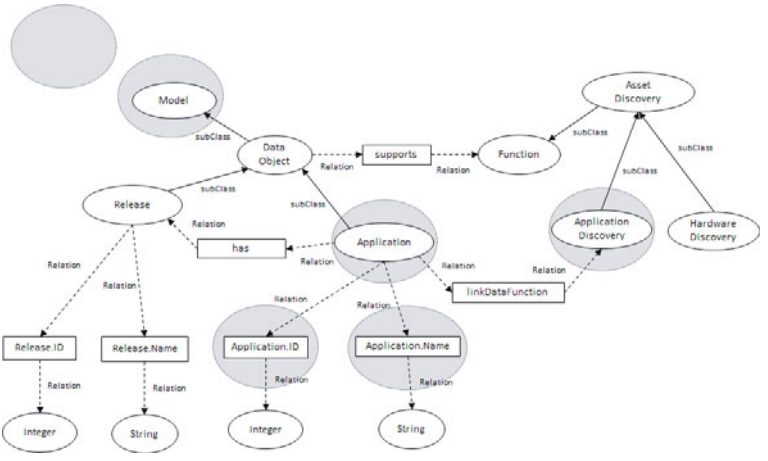


Fig. 6. The *disc*2 application ontology

1. Within the process ontology, select the pair of consecutive activities A_1 and A_2 for which supporting applications must be integrated.
2. Construct a reference ontology that models the specifications of the sequence of activities to integrate. The reference ontology is set up from the sequence ontology, using the process ontology (cf., Fig. 9). To this end, extract the ontological fragment representing the selected pair of activities[9]. The extraction algorithm uses keywords and axioms to accomplish this task, such as:
   - $Activity \equiv Function$,
   - $A_1 \equiv SourceFunction$,
   - $A_2 \equiv DestinationFunction$,
   - $supports(x,y) \rightarrow DataObject(x) \wedge Function(y)$,
   - $supportsSource(x,y) \rightarrow SourceDataObject(x) \wedge SourceFunction(y)$,
   - $supportsDest(x,y) \rightarrow DestinationDataObject(x) \wedge DestinationFunction(y)$.

   *Function*s are added as defined. The result serves as *the* reference for further processing steps. Clearly, a detailed representation within the process ontology is warranted to obtain the best possible comparison basis between integration alternatives (cf., Section 4.2). Also note that information about data objects is not present. This fulfills our objective to only use high-level descriptions, ignoring the specific implementation details.
3. Construct the compound ontology. To this end, use the reference ontology as context and the available application ontologies representing A_1 and A_2. If there are more ontologies per activity, construct several respective compound ontologies. Possible results are sketched in Figures 10 and 11. Differences to the reference ontology are in grey, while differences between the two compound ontologies are in dark grey, shaded.

[9] Here, we already selected software installation discovery and comparison.

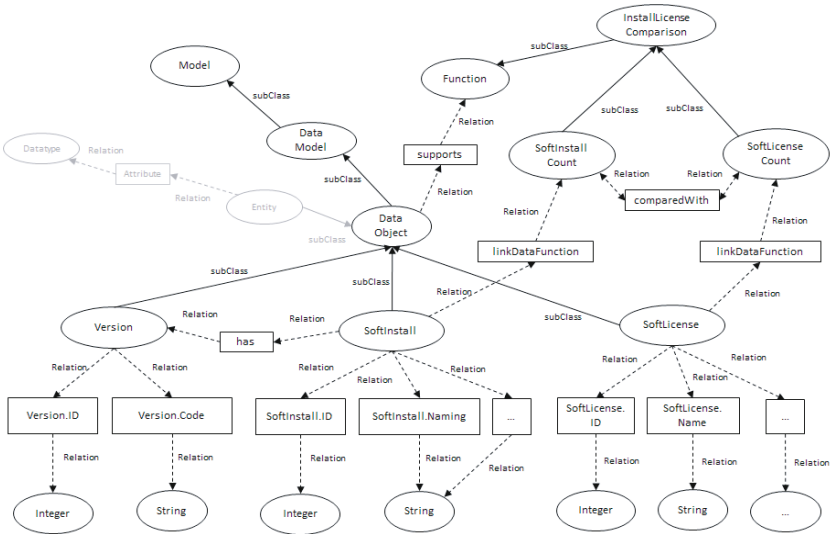


Fig. 7. The *itam* application ontology

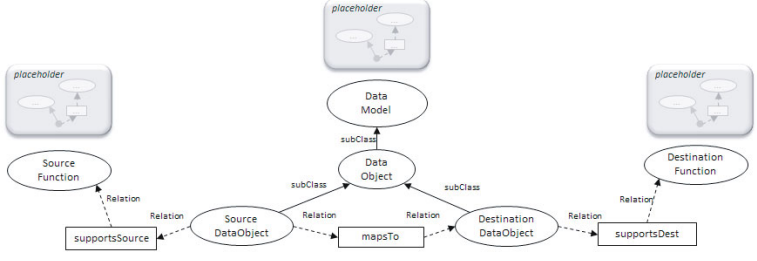


Fig. 8. The sequence ontology. The placeholders are replaced with business specifications, extracted from the process ontology, to construct the reference ontology.

The coupling procedure uses element- and structure-level techniques stemming from ontology matching, such as described in [13]. When applied to applications *disc1* and *itam*, the algorithm may proceed as follows:

(a) Find the *Functions* within *disc1* which are most similar to members of *SourceFunction* within the reference ontology (e.g., using graph-based or name-based matching techniques). Example: $(SWInstallDiscovery \subseteq AssetDiscovery) \Leftrightarrow (SWInstallDiscovering)$.

(b) Add the *Functions* to the compound ontology as members of *SourceFunction*.

(c) Find the *Functions* within *itam* which are most similar to members of *DestinationFunction*. Example: $(SoftInstallCount \subseteq InstallLicenseComparison) \Leftrightarrow InstallLicenseComparing$.

(d) Add the *Functions* to the compound ontology as members of *DestinationFunction*.

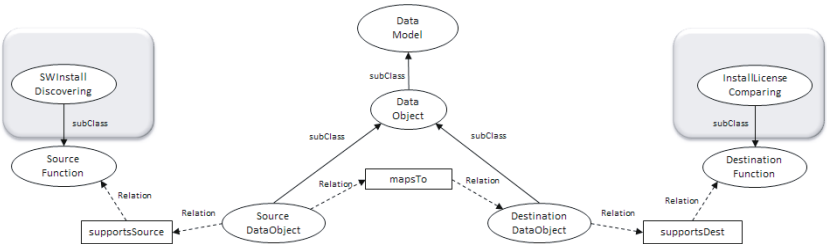


Fig. 9. The reference ontology for the pair of activities $SWInstallDiscovering \otimes InstallLicenseComparing$

(e) Find all *DataObject*s related to the above found members of *SourceFunction* (*DestinationFunction*) and add them as *SourceDataObject*s (*DestinationDataObject*s) within the compound ontology[10].

(f) Search for and store mappings between members (and their datatype properties/attributes) of *SourceDataObject*s and *DestinationDataObject*s. For each source and destination object a mapping is suggested. The mapping may concern one or more attributes per data object (excluding the compulsory key mapping). Thereby, integrity constraints, also encoded within the compound ontology, need to be respected (cf., [14,15,16] for more details). Example: $(SWInstall \Leftrightarrow SoftInstall) \wedge (SWInstall.ID \Leftrightarrow SoftInstall.ID \wedge SWInstall.Name \Leftrightarrow SoftInstall.Naming)$, and $(Release \Leftrightarrow Version) \wedge (Release.ID \Leftrightarrow Version.ID \wedge Release.Name \Leftrightarrow Version.Code)$ (e.g., using string-based, property, datatype, or internal structure comparison).

4.2 Comparing Ontologies through Neural Networks

The next step within the integration method consists of pairwise comparing the compound ontologies with the reference ontology. Our objective is to find the most similar one for further processing. The similarity measure is based on a neural network representation of the reference ontology and the compound ontologies. Specifically, activation patterns, i.e., real vectors, of the same size are generated and compared by using an Euclidean vector distance. This section outlines the approach in more details.

Using neural networks (Step 4). For the reference ontology o_r and each of the compound ontologies o_1, o_2, we want to capture every details at the same time. To this end, we define a vector $\mathbf{z} \in \mathbb{R}^m$ serving as comparison pattern and $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_g]^T$ as a a collection of comparison patterns, with g as the number of ontological concepts within an ontology. The patterns $\mathbf{Z}$ for o_r, o_1, and o_2 are constructed using the Labeling RAAM technique[11] (cf., [17]). The technique has been shown to encode compositional structures such as labeled directed graphs [18]. The resulting patterns are sensitive to the compositional

[10] Integrity constraints are respected (not shown within the ontological descriptions).
[11] Neural network implemented as Recursive Auto-Associative Memory.

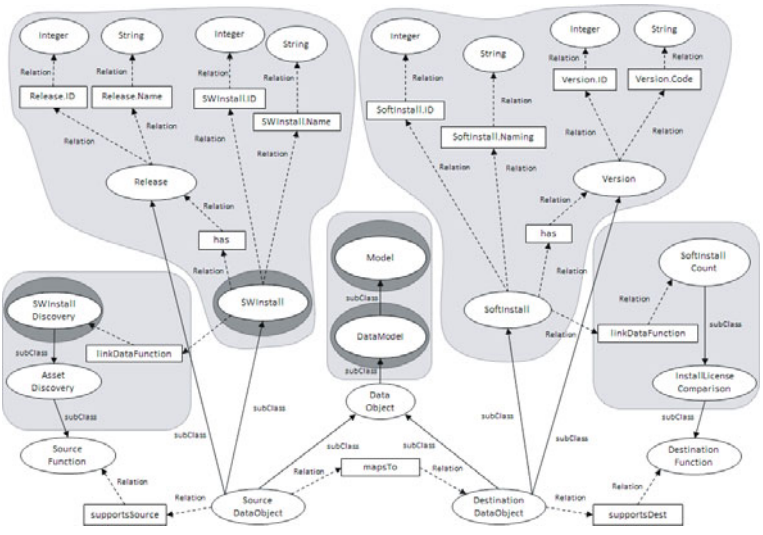


Fig. 10. The compound ontology for $disc1 \otimes itam$

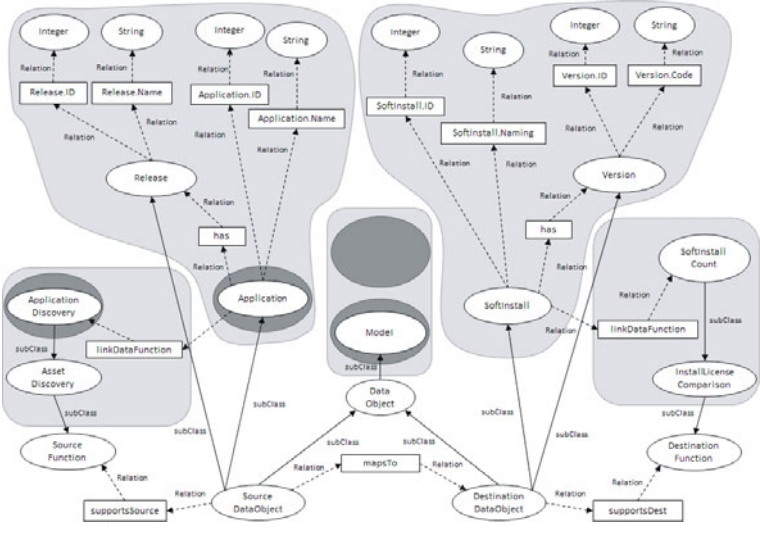


Fig. 11. The compound ontology for $disc2 \otimes itam$

structure they represent. Following [19], they may be used for similarity analysis. We explore the use of this technique for o_r, o_1, and o_2 as they can be seen as directed graphs $G = (N, E)$. Concepts, such as classes or properties are represented as nodes (N) and the connections among them as edges (E) [20].

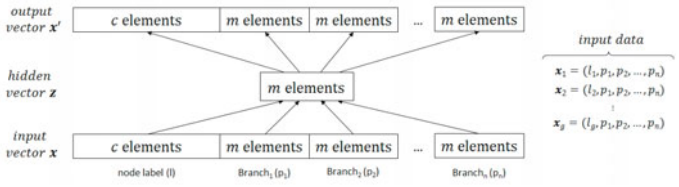


Fig. 12. The LRAAM network (cf., [19])

The general setup of a LRAAM neural network is depicted in Figure 12. Each node within G serves as a single input vector $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$ to the neural network. By extension, $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_g]^T$, the collection of node vectors for a specific ontology, comprises the complete input data to the network. As mentioned, g is the number of nodes within G.

In principle, the network is trained through backpropagation to learn an identity function $F : \mathbf{x} \rightarrow \mathbf{x}'$, where $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^n$. A node vector is compressed by using the function $F_c : \mathbf{x} \rightarrow \mathbf{z}$. Then, the compressed representation is reconstructed using the function $F_r : \mathbf{z} \rightarrow \mathbf{x}$. The node vector $\mathbf{x}'$ is thus an approximated output equal to $\mathbf{x}$. The network is trained by presenting the input vectors repeatedly, one vector at the time.

To obtain $\mathbf{z}$ of a graph node, part of the input (output) vector is allocated to represent the label and the rest to represent pointers (edges) p to connected nodes. There are n pointer slots reserved, where $n = max\{degree(v)\}$, with $v \in G$. Therefore, the input vector is composed of $c + n \cdot m$ elements, where c is the number of elements used to represent node information (label plus pointer existence), and m is the number of elements used to represent pointer values. If a node has less than n pointers, a *nil* pointer is used. The hidden representation $\mathbf{z}$ of a specific node is understood as the pointer for that node. As part of other input vectors, it will thus be used as pointer and recursively fed to the network. Eventually, we obtain the collection $\mathbf{Z}$ of fixed-sized vectors which represents the ontology.

Comparing Ontologies (Step 5). Comparison of the reference ontology and compound ontologies will be performed using $\mathbf{Z}$. Each ontology will be used to train a new network, producing a $\mathbf{Z}_{o_r}$, $\mathbf{Z}_{o_1}$, and $\mathbf{Z}_{o_2}$ respectively. Assuming that there exists a similarity measure $\sigma : \mathbf{Z} \times \mathbf{Z} \rightarrow \mathbb{R}$, such as Euclidean distance, we can determine the best compound ontology by choosing $\overline{o} \in \{o_1, o_2\}$, such that $\sigma(\mathbf{Z}_{o_r}, \mathbf{Z}_{\overline{o}})$ is minimized.

4.3 The Integration Script (Step 6)

The comparison of the distributed representations yields $\overline{o}$ which is chosen for integration. We claim that the underlying applications are technically more suited to support the sequence of activities. Let us assume that o_1 is selected over o_2. An algorithm parses the mappings, established in Step 3 (Sect. 4.1), and derives assertions, such as:

```
SWInstall ∈ SourceDataObject ∈ DataObject ∈ DataModel ∈ Model
SoftInstall ∈ DestDataObject ∈ DataObject ∈ DataModel ∈ Model
∀ x, z.(SourceDataObject(x) ∧ SWInstall.Name(x, z)) ⊃ String(z)
∀ y, z.(DestDataObject(y) ∧ SoftInstall.Naming (y, z)) ⊃ String(z)
mapsTo(x, y) → SourceDataObject(x) ∧ DestDataObject(y)
```

The algorithm then constructs an integration script using these assertions, such as:

```
<STRUCTURE Name="SoftInstall">
  <ATTRIBUTE Name="Naming">
   <PROPERTY Name="Mapping" TYPE="String">
     RetVal = [SWInstall.Name]
   </PROPERTY>
  </ATTRIBUTE>
</STRUCTURE>
```

5 Related Work

Similar approaches to application integration (excluding the use of neural network part) can be found in [21] and [22]. Both suggest enterprise-wide integration architectures, based on semantically enriched web services. In contrast, our research focuses on the micro-level. Here, local, changing business specifications and application pools, beyond the control of a corporate ontology, demand more flexibility than encompassing EAI[12]. More specific aspects such as ontology matching via background knowledge are discussed in [23], [24], or [25]. The matching procedures described therein use upper-level or domain, yet general-purpose (e.g., Foundational Model of Anatomy [FMA]), ontologies as background knowledge. We differ in that we use ad-hoc process ontologies as context for matching, and moreover, coupling. Relevant semantic annotation for underlying processes are depicted in [26], [12], or [27]. Their focus is around the process itself (e.g., reasoning, retrieval, consistency). We aim their use for point-to-point data mapping. The usefulness of a LRAAM's hidden activation pattern is documented in [18]. A simple medical conceptual graph is encoded for querying. We differ in that we use LRAAM for ontology and therefore application selection.

6 Conclusion

We proposed an onto-neural – softw*I*red – method towards application integration. Contribution are identified at various levels: The technical part of application integration is automated. It can therefore be managed through non-technical personel, which clearly has a positive financial impact for the enterprise. In order to reach this goal, ontologies are explored for suitability to capture high-level specifications, and by means of context-driven ontology coupling, for the

[12] Enterprise Application Integration as a company-wide integration standard.

construction of mapping scripts. This approach goes beyond "classic" ontology mapping for data mediation or translation. Nevertheless, interesting insights for the field of ontology matching are expected. Furthermore, to the best of our knowledge, the use of LRAAM for ontology selection is original.

A number of questions still have to be addressed. For instance, we need to determine if ontologies are sufficiently expressive to capture business specifications. If so, what happens when they evolve over time. Clearly, changes need first to be recorded within the ontologies, for example through ontology learning[13]. Once the changes are made, we execute the integration method again which yields an updated integration script (cf., [28][p.46–47]). The behavior of this procedure needs extensive experimentation which clearly also depends on the LRAAM's capability to produce reliable patterns. A prototype is being implemented to address the questions raised.

References

1. ITSMF International: IT Service Management Based on ITIL V3 - A Pocket Guide (2007)
2. International Association of Information Technology Asset Managers: IAITAM Best Practice Library - IBPL (2008)
3. The IT Governance Institute: COBIT 4.1. (2007)
4. Lam, W., Shankararaman, V.: Enterprise Architecture and Integration: Methods, Implementation and Technologies. IGI Global (2007)
5. Hner, K.M., Ofner, M., Otto, B.: Towards a maturity model for corporate data quality management. In: Proceedings of the 2009 ACM Symposium on Applied Computing, pp. 231–238. ACM, New York (2009)
6. Hammer, B., Hitzler, P.: Perspectives of Neural-symbolic Integration. Springer, Heidelberg (2007)
7. d'Avila Garcez, A.S., Lamb, L.C., Gabbay, D.M.: Neural-Symbolic Cognitive Reasoning. Springer, Heidelberg (2009)
8. Antoniou, G., van Harmelen, F.: A Semantic Web Primer, 2nd edn. The MIT Press, Cambridge (2007)
9. Rittgen, P.: Handbook of Ontologies for Business Interaction. IGI Global (2008)
10. DARPA: Darpa-baa-09-03 machine reading broad agency announcement (baa). Online (November 2008)
11. Jones, T.M.: Artificial Intelligence - A Systems Approach. Infinity Science Press (2008)
12. Groener, G., Staab, S.: Modeling and query patterns for process retrieval in OWL. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) ISWC 2009. LNCS, vol. 5823, pp. 243–259. Springer, Heidelberg (2009)
13. Euzenat, J., Shvaiko, P.: Ontology Matching. Springer, Heidelberg (2007)
14. Xu, Z., Zhang, S., Dong, Y.: Mapping between relational database schema and owl ontology for deep annotation. In: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2006, pp. 548–552. IEEE Computer Society, Washington, DC, USA (2006)

[13] We do not address this within the scope of this work.

15. Sane, S.S., Shirke, A.: Generating owl ontologies from a relational databases for the semantic web. In: Proceedings of the International Conference on Advances in Computing, Communication and Control, ICAC3 2009, pp. 157–162. ACM, New York (2009)
16. Bagui, S.: Mapping owl to the entity relationship and extended entity relationship models. Int. J. Knowl. Web Intell. 1, 125–149 (2009)
17. Sperduti, A.: On some stability properties of the lraam model. Technical report, International Computer Science Institute (1993)
18. de Gerlache, M., Sperduti, A., Starita, A.: Using labeling raam to encode medical conceptual graphs (1994)
19. Ellingsen, B.K.: Distributed representations of object-oriented specifications for analogical mapping. Technical report (1997)
20. Eder, J., Wiggisser, K.: Detecting changes in ontologies via dag comparison. In: Krogstie, J., Opdahl, A.L., Sindre, G. (eds.) CAiSE 2007 and WES 2007. LNCS, vol. 4495, pp. 21–35. Springer, Heidelberg (2007)
21. Bouras, A., Gouvas, P., Mentzas, G.: Enio: An enterprise application integration ontology. In: DEXA 2007: Proceedings of the 18th International Conference on Database and Expert Systems Applications, pp. 419–423. IEEE Computer Society, Washington, DC, USA (2007)
22. In: ONAR - An Ontology-based Service Oriented Application Integration Framework, pp. 65–74. Springer, London (2005)
23. Aleksovski, Z., ten Kate, W., van Harmelen, F.: Ontology matching using comprehensive ontology as background knowledge. In: Shvaiko, P., et al. (eds.) Proceedings of the International Workshop on Ontology Matching at ISWC 2006, CEUR, pp. 13–24 (2006)
24. Fatemi, H., Sayyadi, M., Abolhassani, H.: Using background knowledge and context knowledge in ontology mapping. In: Proceeding of the 2008 Conference on Formal Ontologies Meet Industry, pp. 56–64. IOS Press, Amsterdam (2008)
25. Mascardi, V., Locoro, A., Rosso, P.: Automatic ontology matching via upper ontologies: A systematic evaluation. IEEE Trans. on Knowl. and Data Eng. 22, 609–623 (2010)
26. Francescomarino, C., Ghidini, C., Rospocher, M., Serafini, L., Tonella, P.: Semantically-aided business process modeling. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) ISWC 2009. LNCS, vol. 5823, pp. 114–129. Springer, Heidelberg (2009)
27. Weber, I., Hoffmann, J., Mendling, J.: Semantic business process validation. In: SBPM 2008: 3rd International Workshop on Semantic Business Process Management at ESWC 2008 (June 2008)
28. Babin, G.: Adaptiveness in information systems integration. PhD thesis, Troy, NY, USA (1993)

Flexible Communication Based on Linguistic and Ontological Cues

Jean-Paul A. Barthès

UMR CNRS 6599, HEUDIASYC,
Université de Technologie de Compiègne, France

Abstract. The paper addresses the issue of communication between a human and a set of distributed agents. The approach consists of two steps: a first step to identify possible actions requested by the human, and a second step where natural language data are interpreted by agents that possibly could execute the selected action. Action selection involves using linguistic cues, processing the input sentence uses ontological cues as well as the knowledge-base structure. The architecture can be extended to information systems, an example of which is given in the paper.

1 Introduction

Working with complex distributed applications like Terregov, a European project dealing with social services[1], we found that communication between a human and a complex system constitutes a serious bottleneck. Indeed, interfaces for such systems are difficult to implement and maintain, using traditional point and click interaction. We found that vocal interfaces could offer a more flexible approach (Paraiso [11]). Complex systems are often distributed. When such systems are built on multi-agent technology, interaction is done through the use of personal assistant agents (PA). However, implementing a personal assistant is a difficult enterprise and leads to complex pieces of software that become rapidly unmanageable. To decrease the complexity of a PA, Negroponte has proposed to use the concept of *digital butler* [9], limiting the role of the personal assistant to handling the conversation between the human master and the system, and presenting data at relevant times. The PA knows about a number of actions that can be undertaken. It tries to determine which action is intended by its master, and subcontracts the execution of the action to more specialized agents belonging to its *staff*. Interpreting the request is thus a shared responsibility between the personal assistant and the staff agents. The approach is interesting since, contrary to web services, the responsibility of interpreting the (informal) data lies not with the sender, but with the receiver, and depends on the knowledge of the receiver, which is more in line with the agent paradigm. This type of architecture can be extended to build for example distributed information

[1] The Terregov project (European IP) aimed at developing a cooperative platform for helping civil servants do a better job in the domain of social care. It adopted a semantic web service approach. Results can be found at http://www.terregov.org.

G. Babin, K. Stanoevska-Slabeva, P. Kropf (Eds.): MCETECH 2011, LNBIP 78, pp. 131–145, 2011.

systems, in which each agent is responsible for the management of a particular set of data. In particular, we have built an application, HDSRI, for handling the international exchanges of our laboratory. We use it to illustrate our approach.

Currently the "intelligent" personal assistant approach is promoted by important projects like the DARPA PAL (Personal Assistant that Learns) program that aims at providing a personal assistant to each participant in a war room[2]. A consortium of laboratories lead by SRI is working on this initiative on a project called CALO (Cognitive Assistant that Learns and Organizes)[3]. In Australia Nguyen developed a vocal interface with a set of agents [10]. In a related domain, a number of projects are researching the best way of interacting in a Q/A (question answering) approach using ontologies and a combination of various techniques [1].

The paper describes a more modest approach dealing with interacting with a complex system in a professional environment, using ontologies and knowledge bases. It describes an architecture implementing the idea of digital butler, introduces briefly the formalism used to model ontologies and knowledge bases, explains the two-step communication mechanism, and details how informal requests are processed by each agent of the staff.

2 Overall Approach

The overall approach consists in using a multi-agent platform to implement the digital butler architecture, and in designing and implementing a processing mechanism adapted to the selected environment.

2.1 Multi-agent Platform

We use the OMAS platform [2], designed for building cognitive agents[4]. The platform offers four types of agents: service agents (SA), personal assistants (PA), transfer agents (XA) and inferer agents (IA). Agents are organized around a single net loop and share messages (broadcast mode). A physical loop is called a *local coterie*, transfer agents are used to connect different physical loops or different platforms acting as gateways. A set of loops in a particular application is called a *coterie*. The system is open, all communications are P2P meaning that there is no central directory (white, yellow, nor green book). Any agent can join or leave the system at any time. A typical coterie is shown Fig.1, in which we can see service agents, a transfer agent and a personal assistant with its staff.

2.2 Digital Butler Architecture

A personal assistant (PA) is an agent in charge of interfacing a particular person. It is associated with this person referred to as its *master*. The role of the PA is to

[2] www.darpa.mil/ipto/programs/pal/pal.asp
[3] www.ai.sri.com/project/CALO
[4] Available at www.utc.fr/∼barthes/OMAS/

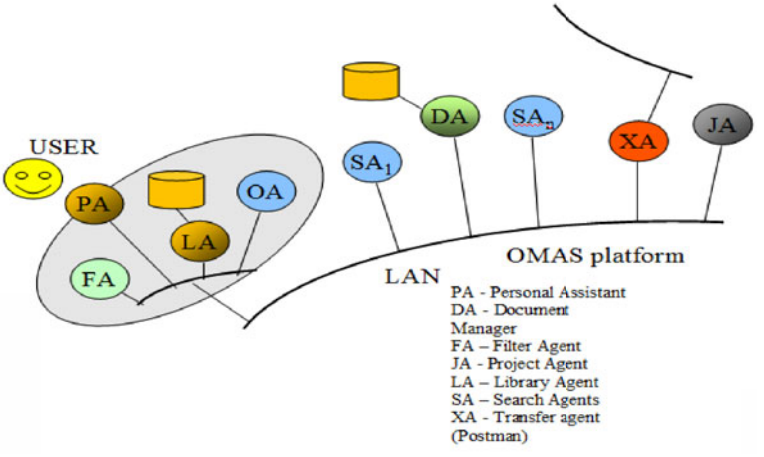


Fig. 1. Coterie architecture showing Personal Assistant and staff agents (FA, LA and OA) on a local coterie, and Transfer Agent XA connecting another remote local coterie

simplify the interface with the multi-agent system. Fig.1 shows a PA interfacing a user and its set of staff agents. In the particular application consisting of organizing page references, staff agents include a filtering agent (FA), a library agent (LA) and an organizing agent (OA). Staff agents answer their PA and nobody else. However, they can access any other agents in the coterie.

In many approaches communication with a PA is usually done through a natural language dialog, using traditional NL processing. However, as already mentioned, this leads to complex programming of the PA, and to a major problem when it needs to be maintained or extended. Furthermore, in complex applications, we want to interact vocally, which means that grammar-based interactions usually break down either because the system fails to recognize some words or because the person who speak uses ungrammatical language (see the thesis of Anh Nguyen [10] for an example of vocal interaction using a BDI architecture).

2.3 The 2-step Approach

In our case, we consider that the PA is used in a professional environment by opposition to open public environments. A consequence is that the number of possible actions is not too large and the context is rather limited. The problem is thus simplified and can be tackled by a 2-step approach: (i) recognizing what action is requested; and (ii) extracting the relevant parameters from a noisy input. The first step can be solved by using a library of possible actions or tasks, the second step can be solved by triggering a special purpose dialog in the PA, or letting the staff agents deal with the raw input. In this paper we only consider the second method although both methods are used simultaneously.

3 Ontology and KB Formalism

An ontology is necessary for defining the domain and exchanging information. It is used to model the tasks (library of tasks), to model the domain and to produce various knowledge bases. We use a representation language called MOSS [2], summarized in the next paragraph[5].

3.1 The MOSS Representation Language

MOSS is a complex frame-based representation language, allowing to describe concepts, individuals, properties, classless objects, default values, virtual concepts or properties. It includes an object-oriented language, a query system, multilingual facilities, OWL and Jena compilers[6], and much more. Interested readers are referred to the online documentation. We only describe here the features used for communication, using examples from the HDSRI application. MOSS is centered on the concept of property and adopts a descriptive (insisting on typicality) rather than prescriptive approach, meaning defaults are privileged and individuals may have properties that are not recorded in the corresponding concept.

Concepts are represented as objects with properties (*attributes* and *relations*). Table 1 shows how the concept of (international) contact is defined.

Relations are binary. They link two objects, e.g. a contact and a country. Relations are automatically inverted so that the country is linked to the contact. The inverse link however bears no semantics. It is a simple means to traverse an arc of the semantic graph in the reverse direction. An inverted relation is called an *inverse link*.

Attributes have associated values. All attributes have multiple values that may be restricted by various types of constraints. An attribute can be inverted, providing direct access to the associated object. Thus a value becomes an index. In case of multiple values, each single value may give birth to several index entries. Each index entry is an individual, and is called an *entry-point*. Table 2 shows the object representing Japan and an associated entry point. Entry points play a central role for interpreting user inputs. In addition they allow optimizing queries.

The MOSS formalism is used to represent ontologies and the corresponding knowledge bases (a detailed example is given in Bettahar et al. [3]).

3.2 Sharing Ontologies

In order for the agents to communicate they share part of their ontologies. The PA has a somewhat sketchy ontology allowing to determine what action is requested, the staff agents, being specialized, have an extended domain ontology

[5] Available at http://www.utc.fr/~barthes/MOSS/
[6] In the SOL sublanguage

Table 1. The concept of contact

```
(defconcept "contact"
            (:att "contact type" (:one-of :occasional :permanent))
            (:att "start date" (:unique))
            (:rel "country" (:to "country"))
            (:rel "foreign correspondent" (:to "person"))
            (:rel "partner" (:to "organization"))
            (:rel "UTC correspondent" (:to "person"))
            (:att "comment"))
```

Table 2. The object representing Japan and the JAPAN entry point

Internal id	properties	values
<country.22>		
	TYPE:	<country>
Attributes:		
	NAME:	"Japan"
Relations:		
Inv-links:		
	IS-COUNTRY-OF:	<contact.21>, <contact.22>
JAPAN		
	TYPE:	<entry point>
Attributes:		
Relations:		
Inv-links:		
	IS-NAME-OF:	<country.22>

corresponding to their domain. Their ontology includes most of the ontology of the PA. For example when a PA asks something to a staff agent, it may associate a pattern to its request specifying the format of the expected return information. It is the responsibility of the staff agent to fill the pattern with relevant data. The pattern uses the terms of the PA ontology.

For example the PA may have a concept representing a "person" with an attribute stating its "address." The ADDRESS staff agent will have a more detailed model of "address" including "street name and number," "city," "zip code," "state," and "country." If the PA sends a pattern like

```
("person" ("name") ("address"))
```

It is the responsibility of the ADDRESS agent to fill the "address" field with data extracted from its representation of an address (this is implemented by a summary method).

The following paragraphs explain how the interaction is handled.

4 Selecting an Action (PA)

4.1 The Library of Tasks

Whenever the user asks something, its PA first tries to determine what action is actually meant by the request (or assertion). To do so it uses a library of possible actions, defined as individuals of the concept of task. Table 3 shows how the task for obtaining project statistics is defined, using a deftask macro that produces a task individual. The :indexes parameter specifies a list of linguistic cues. Each phrase (cue) has a weight between -1 and 1. Note the parameters *:dialog* and the pair *:where-to-ask*, *:action*. The first one, *:dialog*, is used for calling a dialog to analyze the input and ask for missing information before calling the staff agent(s). It is used when access is done through a PA client interface. The pair *:where-to-ask*, *:action* indicates which staff agent and what skill are concerned with the input, and is used when no dialog is feasible (e.g. batch processing of an email), which is the case considered in this paper.

Table 3. Defining a task individual

```
(deftask "get project"
        :doc "Task for getting project statistics"
        :performative :request
        :dialog _get-project-statistics-conversation
        :indexes
            ("project" .4 "projects" .5 "statistics" .4 "stats" .4
            "current" .4 "active" .4 "on going" .4 "actual" .4)
            )
        :where-to-ask" :PROJECT
        :action :statistics-html)
```

4.2 The Selection Mechanism

Selecting an action (a task) is done as follows:

1. The user tells something to her PA, like "*what are the current projects?*";
2. For each task in the library the PA checks the sentence for phrases specified in the index pattern describing the task, and computes a score by using a MYCIN-like formula[7];
3. tasks are then ordered by decreasing scores;
4. the task with the higher score is selected if the score is above a specific threshold.

Two points must be made here regarding the choice of the indices and the choice of the weights.

[7] If 2 cues are present, the combined score is computed by the formula a+b-ab.

- Indices use terms from the ontology. However, one needs additional terms like "*on going*" or abbreviations that are linguistic additions. Also, one could lemmatize the input sentence, but this has not been found very useful. Selecting the right linguistic cues should be the result of experiments using for example a magician of Oz set up.
- Specifying the weights manually is tricky. The resulting score is used to differentiate among tasks. Thus the weights have to be fine tuned to produce the right answer. An algorithm developed by Gonzalez allows computing the weights using a neural network, but it has not been published yet.

Finally, when using vocal input the sentence received by the PA may be garbled like "*water the current projects*", in which case a grammatical analysis is not very useful.

4.3 Sending Information to the Staff Agents

The request sent to the staff agents uses a very simple content language in which two fields are specified. In our example:

```
:data "what are the current projects"
:language :EN
:pattern ("project" ("title")("start date")("partner")("UTC correspondent"))
```

The pattern argument is optional in case the PA wants to build an individual using the response and keep it in its memory. When no pattern is specified, the answer is meant to be a literal (string).

5 Interpreting the Data (Staff Agent)

Before detailing the way requests are processed, we first remark that they are most of the time very simple as shown by examples in Table 4.

5.1 Global Approach

The main idea is to use a graph-covering algorithm on the combination of the semantic nets representing the local ontology and the knowledge base. The goal is to determine target objects at the root of a spanning tree containing the properties or concepts (or sub-concepts) present in the data. Nothing really new here since it constitutes an extension of what Quillian's proposed in the 60s [12]. The spanning tree is constructed by traveling from the entry points contained in the request to a target concept determined by the skill we are executing. For example if we are executing the skill :get-contact-statistics, the target concept will be contact. The fan out is done breadth-first.

5.2 Types of Requests

Because we do not allow negations or disjunctions most of the requests are very simple. They can be sorted out into several categories:

Table 4. Examples of requests

What contacts do we have with Japan?
What are Moulin's contacts in Australia?
Do we know somebody at Todai?
Do we have projects with China?
Give me statistics about our projects?
Contact stats?
Who is Moulin?
What is the phone number of the president of UTC?
Where does Dominique live?
...

- requests that do not contain individual entry points, e.g. "*Give me statistics about our projects?*";
- requests that contain a single individual entry-point, e.g. China in "*Do we have projects with China?*";
- requests that contain individual entry points and relation entry points, e.g. UTC and president in "*What is the phone number of the president of UTC?*";
- requests that contain individual entry points and relation and concept entry points, e.g. CIT, Moulin, Japan and partner in "*Is CIT a partner in Moulin's projects with Japan*?".

5.3 Mechanism

The approach is implemented as follows:

1. Determine the target concept (depends on the skill).
2. Split the input text into a list of words.
3. Combine the input words to extract from the ontology and knowledge base entry points specifying concept, properties, or individuals.
4. If the result does not contain individual entry points, do a special processing.
5. Otherwise, extract the individual entry points and mark the concepts and relations in the ontology.
6. Start from each individual entry point, constructing a set of paths breadth first until either an instance of target concept has been found and all marked items have been traversed, or we have reached a limiting maximal length for the paths.
7. Intersect all sets of paths keeping the shared individuals of the target concept.

Entry points and special processing. It can be seen that the role of entry points on individuals is essential. If there are no individual entry points (step 4), then either the request is asking for a special treatment (e.g. get statistics) or we do not have enough information to a answer it. Special processing is done in the corresponding agent skill.

Table 5. Example of adding (fake) data (brackets mean optional data)

add contact : / new contact :	
country :	zimbabwe
{correspondent} name :	Mbozza
{{correspondent} first name :	Manfred}
{{correspondant} initial :	M}
{url : URL reference}	
partner name :	University of Harare
{partner acronym :	ZNU}
HDS :	barthès, mueller
{date :	2010}
{comment :	met in Cape Town}

Subsumption. It can be seen that the paths are built using the graph of the knowledge base. When traversing an individual, we check whether it is an individual of a sub-concept of a marked concept, taking subsumption into account.

Fan out. Extending a path from a specific individual is a complex action. There are two cases:

1. If the current object cannot have a relation or inverse-link corresponding to one of the relations that have been marked, then we extend the path as follows: we create a new path to all neighbors of the current object following all existing relations and all existing inverse links.
2. Otherwise, we check if the current object has the marked relation instantiated, and if so we add paths following this relation; otherwise, we kill the current path.

5.4 Adding Information

Adding information is a little more constrained since we must parse the input text. Therefore one must obey a predefined format, which simplifies the application-dependent parser. For the HDSRI application, possible formats are given on an example in Table 5.

6 The HDSRI Example

One of the applications developed using this approach is the HDSRI application for managing international contacts and projects of the HEUDIASYC laboratory (HDS). The system comprises four agents (Fig.2): ADR containing the name and addresses of known people or members of the laboratory, CNT managing a list of international contacts, PRJ managing the set of international projects, and HDSRI PA. An additional agent, FINANCING, managing the potential funding programs has been added recently and is not shown on the figure.

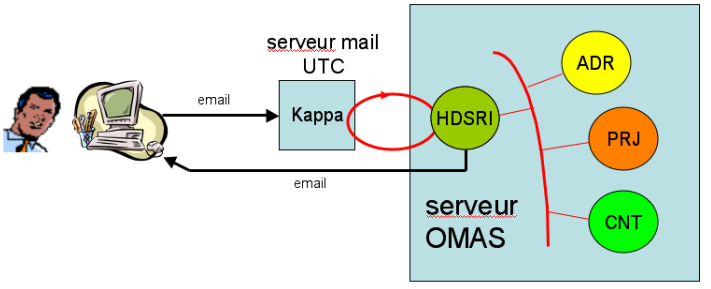
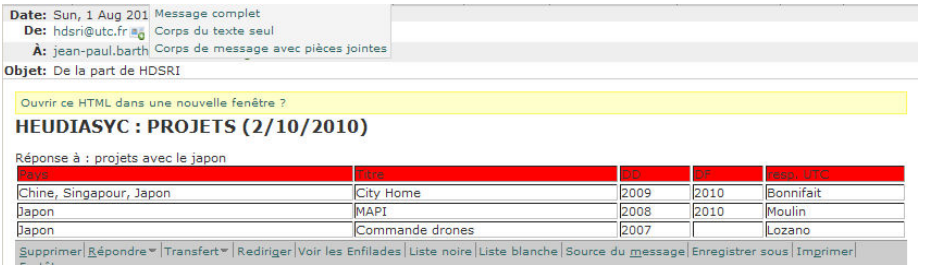


Fig. 2. Architecture of the HDSRI prototype


Date: Sun, 1 Aug 201 Message complet
De: hdsri@utc.fr Corps du texte seul
À: jean-paul.barth Corps de message avec pièces jointes
Objet: De la part de HDSRI

Ouvrir ce HTML dans une nouvelle fenêtre ?

HEUDIASYC : PROJETS (2/10/2010)

Réponse à : projets avec le japon

Pays	Titre	DD	DF	resp. UTC
Chine, Singapour, Japon	City Home	2009	2010	Bonnifait
Japon	MAPI	2008	2010	Moulin
Japon	Commande drones	2007		Lozano

Supprimer | Répondre | Transfert | Rediriger | Voir les Enfilades | Liste noire | Liste blanche | Source du message | Enregistrer sous | Imprimer

Fig. 3. Answer from HDSRI concerning projects with Japan

The application has two interfaces: (i) an email interface letting people send emails to the HDSRI agent; and (ii) a direct interface with the PA, used currently for maintenance. We only describe here the email interface.

In the email interface, there is in practice no dialog. Processing is done on the data and an answer is expected containing the result. The HDSRI PA executes a periodic goal, examining its mailbox for incoming requests and sends an answer mail as shown Fig.3.

The interactive interface allows dialogs with the PA and is much more complex. Dialogs are constructed as dialog graphs with transitions between the states represented by the nodes. Dialogs must be developed for each task of the task library. This particular aspect of interactions is outside the scope of this paper.

6.1 Simple Email Request

If the user sends an email containing the following questions:

- *what are the projects with Japan?*
- *projects with Japan?*
- *Japan projets?*
- *In Japan, have we a project?*

Such sentences are trivial variations on the same theme and are processed as follows:

- HDSRI selects the task "get project" that ranks first due to the presence of the word "project" or "projects" in the input.
- it ships the input to the PROJECT agent a pattern:

```
((:data "projects" "with" "Japan")
 (:pattern "projet" ("title")("start date")("end date")
        ("UTC correspondent")))
```

- The project agent uses the "Japan" entry point to build a path from the "project" concept to the individual representing Japan.
- the path is used as a query to retrieve information.
- the information is sent back to the HDSRI PA as a list of data structured according to the pattern.
- HDSRI produces an HTML table with the results, and sends a message back to the user.

6.2 More Complex Requests

More complex questions come from the way they are processed, not from their initial expression. For example a request equivalent to "Do we know somebody in Japan at CIT ?" leads to exactly the same processing than in the previous paragraph, the difference being the presence of two entry points (Japan and CIT) rather than one. The following requests however lead to a different processing:

- *"Do we know somebody in Sydney?"*
- *"What are our contacts in Sydney?"*

The main problem is that "Sydney" is not an entry point of the CONTACT agent. The CONTACT agent only knows about countries, not towns. Therefore the CONTACT agent will need help from other agents or from the user (in interactive sessions). The request is processed as follows:

- the HDSRI agent selects the get-contact task from the input;
- it sends the input to the CONTACT agent with a pattern

```
((:data "What" "are" "our" "contacts" "in" "Sydney")
 (:pattern "contact" ("person" ("name"))("start date")("country")("UTC correspondent")))
```

- the CONTACT agent receives the request, but cannot find any known data entry point in the input.
- the contact agent will then broadcast successively two messages:

```
((:find "country")
 (:pattern "country" ("name")))
```

and, if unsuccessful

```
((:find "person")
 (:data "What" "are" "our" "contacts" "in" "Sydney")
 (:pattern "person" ("name")))
```

- If one of the query returns an answer, it will use it to retrieve the requested information as previously
- otherwise, the CONTACT agent will return a :failure answer to the HDSRI PA
- the HDSRI PA will inform the user

In this approach the CONTACT agent strategy is contained (wired in) its get-contact skill, which for simple action is easy to realize. The system however cannot process very long requests or requests containing negations or disjunctions (see next section).

7 Discussion, Limits and Further Work

Differences with web service. The approach we presented in this paper amounts to spreading the analysis of a request among a set of agents. In the case of a digital butler approach the agents participating in the analysis are the staff agents, making the process rather localized. However, as shown by the HDSRI example, the approach can be extended to any set of agents that share a minimal part of ontology. The interpretation of a request is thus done by the agent that receives the message rather than by the sender of the message as for example in a web service approach. Doing so implies that the meaning is a function of the representation (ontology) and knowledge (knowledge base) of the receiver, which is in line with the agent paradigm.

Advantages of the two-step process. The two-step process has other advantages: one is to decrease the load of a personal assistant, since its work consists then in finding the type of service requested and in distributing the raw data to the agents more susceptible to process it; another one is that the analysis of the data is done in the context of specialized services, being thus more focused and making it easier to implement.

Limited request format. Of course there are limits to the approach. Requests must not be too complex and we do not allow negation nor disjunctions. The latter restrictions could be cancelled by using local grammars (like link-grammars), which amounts to grouping sequences of words in the input data. We also currently do not use references to attributes, outside values that are entry points, with the exception of dates. Taking account attributes amounts to building filters on the resulting set of target individuals and could be done easily.

Alternative approach. Another approach quite similar to the one developed in the paper consists of using the semantic graph representing the ontology in the staff agent, together with entry points from the knowledge base, and reconstruct a set of formal queries, based on the same concept of spanning tree. The reconstructed formal queries are then executed against the local knowledge base. Although this gives good results, it is less efficient that traversing the knowledge base directly because part of the work is done twice.

Multilinguisms. Currently our ontologies are multilingual (hence the language parameter in the requests), but a PAs can "understand" a single language. Multilingual applications must use several PAs in parallel, which does not constitute a problem.

Salient features and case bases. To improve the processing of raw natural language inputs one could take into account past requests by keeping stacks of salient features. However, we do not think that the increased complexity is worth it at the level of staff agents. We have tested the possibility of giving a PA a short term and long term case memory (Chen [4]), but this makes it quite complex and may not be worth it.

8 Relevant Works

As mentioned in the introduction research on "intelligent" personal assistants is very active, as well as research on Question/Answering systems.

The work mostly resembling our approach is that of Wong, Nguyen and Wobke [15]. In their approach a user is talking to his PDA and the system is trying to execute his request like displaying the emails, or organizing meetings. The implementation is however entirely different, since it uses the inference engine of the BDI agent platform to reason about the user's utterance. This mixes the agent execution mechanism with the mechanism of processing the data and leads to rather complex programming detailed in Nguyen's doctoral thesis [10]. However, the assumptions of a limited context and fairly simple questions is similar to ours.

The recent PAL program and CALO project intend to build intelligent personal assistants that can help people in a control room or in a meeting. They involve a large number of people as can be seen from the number of authors in the publications. Within the CALO project a personal assistant, PExA (Project Execution Assistant), devoted to aid people in the context of office work has been developed [8]. The implementation is based on a BDI platform. The goal of the assistant is to aid the user to manage time and tasks in connection with other personal assistants in the system. The interface with the user is not the main point of the project. It is based on producing context dependent interaction frames, and also accepts typed sentences in natural language. The CALO project however developed vocal interfaces, proposing a Meeting Assistant that can analyze vocal inputs from the participants in a meeting and provide an indexed summary at the end. Encountered problems are described in [13].

In the domain of multi-agent sytems, Knott and Vlugter [6] report work in understanding multi-agent dialogs, focusing on the problem of dialog management (determining the addressee, resolving pronouns) among multiple independent agents. The domain is open and the sentences are very short.

On the side of Question/Answering approaches the system developed by Lenat et al. [7] for clinical research uses the Cyc ontology in a restricted but significant

domain. The system can answer complex questions typed on several lines, after breaking them into parts, process each part, getting the results back and using consistency to filter irrelevant answers. Inputs are typed and are much more complex than in our case. In the DeepQA project reported by Ferrucci et al. [5] questions are short but the domain is fairly open, and answer time is a critical factor.

Conclusion. The approach presented here is both simple and efficient, in particular when faced with noisy data. It is also easy to program. Some people may be worried to let service agents interpret the content of a request and propose their answers. Of course this can lead to quid pro quo, which in the case of the simple information systems or exchange of information we have tested, does not appear to be a major problem. Furthermore, the approach can be extended and refined to minimize such possibility. Finally, broadcasting a request could generate serendipity.

References

[1] Special issue of AI Magazine on Question/Answering, 31(3) (Fall 2010)
[2] Barthès, J.-P.A.: OMAS - A Flexible Multi-Agent Environment for CSCWD. To appear in Future Generation Computer Systems (2010)
[3] Bettahar, F., Moulin, C., Barthès, J.-P.A.: Ontology support for knowledge management in e-government environments. In: ECAI 2006 - Workshop on Knowledge Management and Organizational Memories, Riva del Garda, Italy, pp. 19–24 (2006)
[4] Chen, K., Barthès, J.-P.: Giving an Office Assistant Agent a Memory Mechanism. In: Proc. ICCI 2008 (2008)
[5] Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A.A., Laly, A., Murdock, J.W., Nyberg, E., Prager, J., Schlaefer, N., Welty, C.: Building Watson: An overview of the DeepQA Project. AI Magazine 31(3), 59–79 (2010)
[6] Knott, A., Vlugter, P.: Multi-agent human-machine dialogue: issues in dialogue management and referring expression semantics. Artificial Intelligence 172(2-3), 69–102 (2008)
[7] Lenat, D., witbrock, M., Baxter, D., Blackstone, E., Deaton, C., Schneider, D., Scott, J., Shepard, B.: Harnessing Cyc to answer clinical researchers' ad hoc queries. AI Magazine 31(3), 13–32 (2010)
[8] Myers, K., Berry, P., Blythe, J., Conley, K., Gervasio, M., McGuinness, D.L., Morley, D., Pfeffer, A., Pollack, M., Tambe, M.: An Intelligent Personal Assistant for Task and Time Management. AI Magazine 28(2), 47–61 (2007)
[9] Negroponte, N.: Being Digital. Alfred a Knopf Inc., Westminister (1995)
[10] Nguyen, A.: An agent-based approach to dialogue management in personal assistants, PhD thesis, University of New South Wales, Australia (2007)
[11] Paraiso, E.C., Barthès, J.-P.A.: An intelligent speech Interface for personal assistants in R&D projects. Expert Systems with Applications 31, 673–683 (2006)
[12] Quillian, M.R.: Word Concepts: A Theory and Simulation of Some Basic Semantic Capabilities. Behavioral Science 12, 410–430 (1967)

[13] Tur, G., Stolcke, A., Voss, L., Peters, S., Hakkani-Tur, D., Dowding, J., Favre, B., Fernandez, R., Frampton, M., Frandsen, M., Frederickson, C., Graciarena, M., Kintzing, D., Leveque, K., Mason, S., Niekrasz, J., Purver, M., Riedhammer, K., Shriberg, E., Tien, J., Vergyri, D., Yang, F.: The CALO Meeting Assistant System. IEEE Transactions on Audio, Speech, and Language Processing 18(6), 1601–1611 (2010)

[14] Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A.A., Laly, A., Murdock, J.W., Nyberg, E., Prager, J., Schlaefer, N., Welty, C.: An overview of the DeepQA Project. AI Magazine 31(3), 59–79 (2010)

[15] Wong, A., Nguyen, A., Wobcke, W.: Robustness of a spoken dialogue for a personal assistant. In: Proc. of the 2007 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, pp. 123–127. IEEE Computer Society, Los Alamitos (2007)

Decentralized Task Allocation Mechanism Applied to QoS Routing in Home Network

Emna Ghedira, Lionel Molinier, and Guy Pujolle

Paris Universitas - LIP6, Paris, France
{emna.ghedira,lionel.molinier,guy.pujolle}@lip6.fr

Abstract. We present in this paper a task allocation mechanism constrained by a distributed, complex environment namely in our case a home network. Thus, added to multi agent systems constraints, we have restrictions in terms of communication (available throughput) and resource (bandwidth). We have a multi agent system with a knowledge base, and we have to ensure a proper quality of service in home Network adjusting the routing in real time by applying alternative route to the main ones. Our approach uses the *first price sealed bid* type auction: when an agent does not have an alternative route, it launches an auction. The agent offering the best price will be the next hop of the route.

Keywords: Home Network, Task allocation, QoS, Knowledge based system, Ontology.

1 Introduction

With the success of the Internet, its expansion in home, the adaptability, flexibility and decentralization of its application, limits had appeared. Those problems affect both users and operators. Because they are neophyte, users are unable to react properly to any event occuring in their home network. This is a direct consequence of high level control lacks. Operators are faced with the inability to properly transcribe business policies to their equipment configuration.

The task allocation problematic is a central issue in software agent systems, distributed systems but also in various areas [11,13,7]. This is a direct consequence of agent's hetereogenity of the multi agent system. Each agent has its own abilities, and a partial view of its environment. The multi agent systems can be assimilated to a company, where an employee does not necessarily know all its coworkers, but all the employees with their own abilities and behaviors converge to a common goal, maximizing corporation's profit [14,13,7].

In this paper, we propose a task allocation mechanism to adjust the routing and also the quality of service in a home network. In this environment, we have constraints in terms of communication as we are limited by bandwidths, in terms of resource which is based on the available throughput and finally because there is only a partial visibility [12].

G. Babin, K. Stanoevska-Slabeva, P. Kropf (Eds.): MCETECH 2011, LNBIP 78, pp. 146–157, 2011.

This paper is organized as follow: we start with an overview of the problem then we are introducing the agent we are using to solve our problematic and one application. Finally we conclude and present some future works.

2 Problem Overview

We call *Home Network* a set of elements that compose our high-tech environment at home with a home gateway (box) and others home devices like phones, television, PC... This type of network can not be studied separately of the Internet thus, the broadband access is the first aspect to be considered [10,9].

No longer than 15 years ago, Internet access at home was reserved to some elite. They were using an analogical modem (56K) connected to their desktop PC, and they were painfully surfing the Internet and reading their e-mails. However, with broadband accesses, the Internet starts to widespread, based on this modem-PC architecture.

This was the first step of the *Home Network*, since triple-play offers introduce the notion of router inside the home. Nowadays there is a home gateway in mostly every home, which provides dedicated interfaces for Internet access, telephony and television. This gateway is providing interfaces for those three services: SCART or HDMI, Ethernet or WiFi, analogical telephony.

In the near future, all those services will merge IP, thus creating only one into network over the Home. The single router architecture will no longer be sufficient because it will have to provide at least 10 Mb/s in each corner of the house (HDTV flow for instance). That is why, standardisation consortiums tend to agree on the architecture illustrated in figure 1, which add several devices named HNID (Home Network Infrastructure Device) in order to improve the coverage (see [6]). Those devices also act as bridges between technologies.

Introducing HNID, the network will cover all the home with good conditions, but also enable the network to suppport more devices. However, creating a real network, with active elements, leverages routing problematic: how can we provide routing in the *Home Network*? The tricky point is that the medium (WiFi) is very perturbation sensitive and its bandwidth may collapse very quickly.

This routing problematic is quite well known in networks. *Home Networks* inherit from ad-hoc networks, but this domain only considers wireless links, excluding PLC[1] benefits. At the opposite, it also inherits from corporate networks which already have such kind of architecture, but in Home Network, the user is neophyte and there is no human network administrator to configure and maintain it. In other words, the *Home Network* has to be autonomic, which means that it has to optimize itself in order to provide the best service possible to the end-user.

[1] PLC stands for PowerLine Communications. In other words, it means that we use power outlet as a network medium.

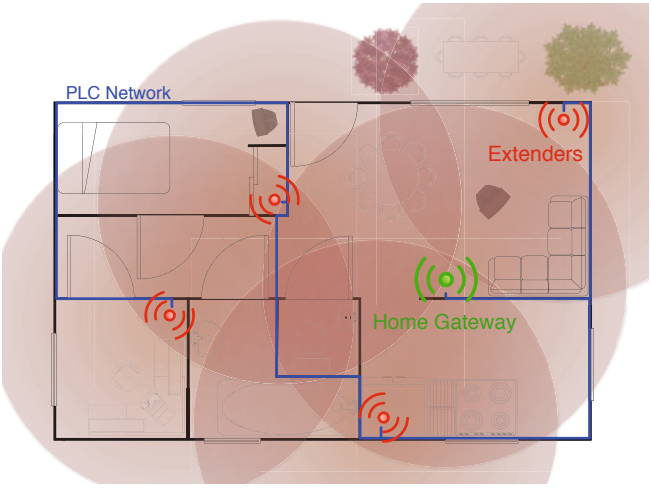


Fig. 1. Architecture of the Home Network

3 Agent-Based Solution

Introducing autonomy in networks is an emerging and widely recognized idea in the telecomunication world. One way to autonomy is to work with a lot of knowledge (see 3.1) within the network but also to take high-level decisions as the network complexity is increasing.

3.1 Knowledge Plane

The knowledge plane has been added upon applications to mutualize information. It has been introduced by [3], a network reseacher, and defined as:

> [...] a distributed and decentralized construct within the network that gathers, aggregates, and manages information about network behavior and operation.

We grasp this concept through a multi-agent system based on knowledge.

3.2 Ginkgo Multi Agent System

What is the Ginkgo Platform? The agent platform can be considered as a middle-ware for Autonomic Networking. It was designed to run onto network equipments: that means the platform is distributed over the network. The platform architecture is presented on figure 2.

Each agent is embedded on a network device and has a partial view of its environment which is defined by the application designer. It communicates only exchanging knowledge with neighbours.

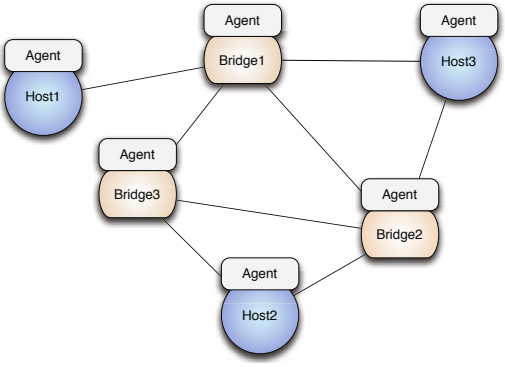


Fig. 2. An agent is embedded each network device

What is an agent in the Ginkgo Platform? The figure 3 presents the structure of ginkgo agent, thereafter the explanation of each component.

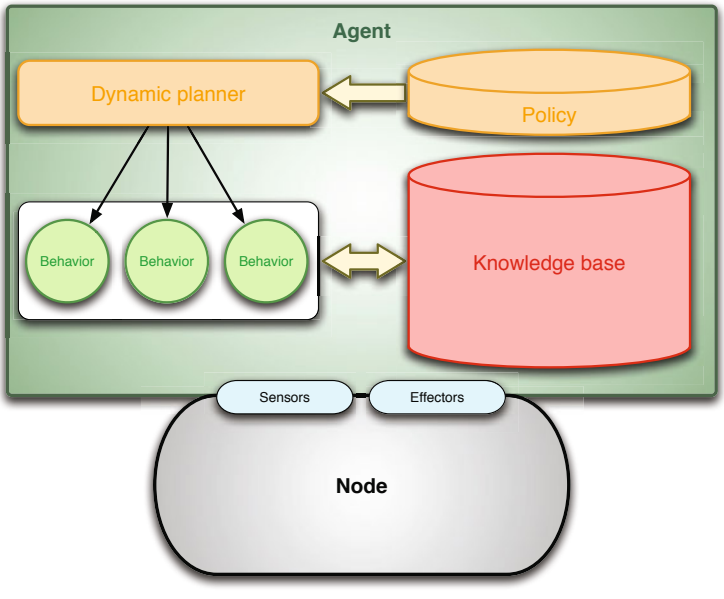


Fig. 3. Ginkgo agent architecture

- **Behavior:** Agent abilities are implemented by the application developer and permanently adapting themselves to environment changes. It can read and write in the knowledge base, sense and act from the node.
- **Policy:** Rule set by operator.

– **Dynamic Planner:** It orchestrates the behaviors. It can start, stop, configure even modify according to the policy.
– **Knowledge Base:** The Knowledge base (KB) is a central functionality of the agent. It stores the data used by an agent in an homogeneous and structured way and provides diffusion mechanisms of this knowledge between the agents.
– **Situated View:** It is the partial view that an agent has of its environment [2]. It gathers local knowledge acquired by examining the device on which is embedded but also the knowledge collected by its peers.

Preliminary work: Ontology, a way to structure the KB. Classicaly, in a network device, there are many of algorithms running simultaneously, each one using its own data. This lead to an important overlap, mandatory, since there is no common way to handle data. KB stores not only information manipulated by the behavior and the dynamic planner but also supplies a representation.

Thus, it stands to reason to define an ontology which allows us to have a common vocabulary for the knowledge representation. This facilitates the communication between agents. The figure 4 represents a subpart of the ontology using Unified Modeling Language (UML).

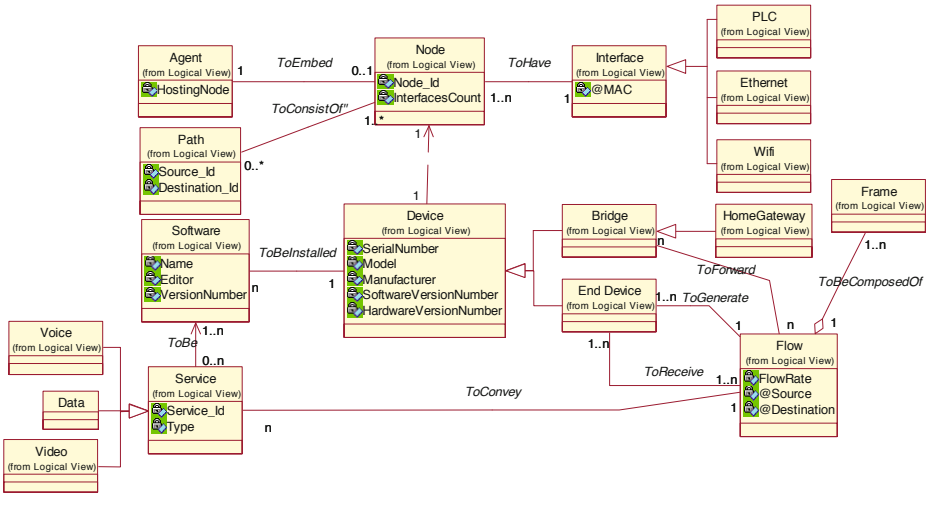


Fig. 4. Extract of the ontology

From this ontology we have derivated a data model and implemented it in the KB. For instance, an agent is *embedded* into a node which *has* interfaces connected to other nodes...

Now, the KB is structured, we have to design the algorithm that relies on this knowledge.

4 Task Allocation in Home Network

4.1 Motivation

In classical approach of routing concerns, we try to tune algorithms to have the lowest convergence time. This can be done by reducing the amount of information taken into account. However, in such a complex environment this may lead to poor routing performances as links quality may fluctuate very quickly.

Our way to handle this problem is to consider that the routing algorithm is performing well in most cases, but need to be by-passed in critical situations, by applying pre-computed alternative routes. This two stages routing assure the quickest response time in case of link failures or link capacity collapse due to external interferences.

Even if the alternative route principle is well-known in the litterature, it is hard to implement, since we first need to monitor the normal operation of the network, and secondly compute (and maintain) in background alternative routes.

When a flow comes to a node, the classical routing process routes it. However, the agent embedded detects this new flow and tries to find out an alternative route. This flow has some specific properties such as type, bandwidth and destination. This make the finding complex (several criteria) and relying on neighbour abilities.

That is why we have chosen to use a task allocation mechanism based on auctions. So, the neighbours can help a node to find out an alternative route offering their ressources. The auction issuer can choose the best offer.

4.2 Formalization

The task allocation problematic is represented by a *triplet*:

$$< T, Ag, c >$$

T : set of tasks

$Ag = 1, \ldots, n$: All agents participating in the execution of these tasks

$C : P(T) \longrightarrow \mathbb{R}^+$: Function defines the execution cost of each task subset and have two constraints:

- Monotony : If we add a task to a set of tasks, the cost of their execution must be superior to the execution cost of the initially set.

$$c(T_1, .., T_n) < c(T_1, \ldots, T_n, T_{n+1})$$

- If a task is not executed the cost must be null.

4.3 Environment

We consider a set of agents $Ag = 1, \ldots, n$ organized according to a network topology as represented in the figure 5. Each agent must handle requests placed in a schedule, knowing that any agent can be the recipient of a query. There are T types of tasks: $T = 1, \ldots, n$.

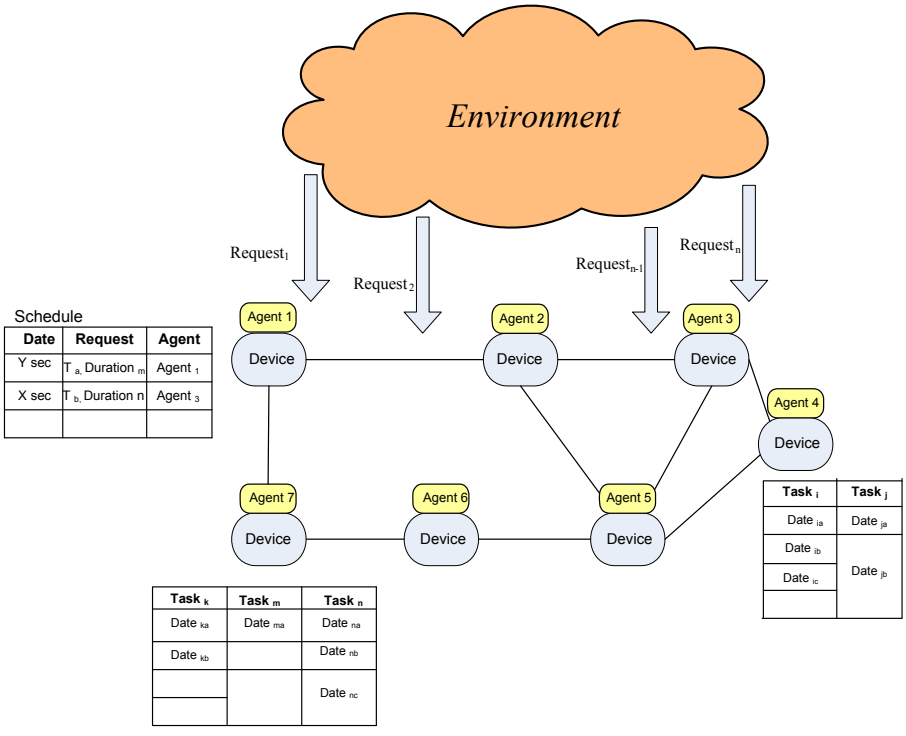


Fig. 5. Illustration of the problem

Agents of the network do not have the same skills, they can not perform all types of tasks. In addition, for a given type of task, they are able to execute only x number of tasks Ti (ie corresponds to the structure that is under the Agent$d$ for instance). The figure 5 illustrates the problem.

Given these constraints, there are two reasons for which a reallocation is necessary:

- The task type of the request received is not within the skill of the receiving agent.
- The agent has reached the maximum number of tasks it can execute.

4.4 Approach and General Principles

The proposed mechanism is based on the theory auction mechanism and inspired by auction *first-price sealed-bid* presented below:

> The initiator starts the auction and each participant submits a bid in one turn, without knowing other participants offer. The one which has made the largest offer wins the subject of the auction and pays the amount its offer [5].

Before we define the general principles, we present the structures that compose the situated view.

- $\{T_{ex}\}$: Queue of task to execute

$Task_i$	$Task_j$
$Task_i$, $Date_a$	$Task_j$, $Date_a$
$Task_i$, $Date_b$	...
...	...
...	

- $\{E_{local}\}$: List of local auctions. This structure brings together all auctions initiated by the agent.

Id_{task}	Reservation Price
$Id(T_i)$	X_i
$Id(T_j)$	X_j
...	...

- $\{E_{global}\}$: List of global published auctions to the network

Editor agent	Id_{task}	Reservation price	Assigned agent	Offers
Ag_1	$Id(T_i)$	X_i		(Ag_4,P_i) ...
...	...	...	...	...

The offer column contains only its own offer except for the editor agent that contains all offers.

- $\{O\}$: List of the offers for a bid published

IdE_{global}	Editor agent	Offer price
$Id(Eglobal_i)$	Ag_1	X_m
$Id(Eglobal_j)$	Ag_4	X_k
...	...	...

These structures are used in algorithms presented below. They are implemented as behaviors and are based on these points:

- A task becomes a bid only when it can not be executed by the receiving agent
- The situated view is only partially broadcasted
- The situated view is transmitted only when there is a change

When an agent receives a task, it checks two conditions:

1. **Condition1:** If the type of task belongs to its skills.
2. **Condition2:** If the maximum number of execution for this type of task is reached.

If **condition 1** is false or **condition 2** is true, then the agent initiate an auction about this task following the algorithm described in figure 6.

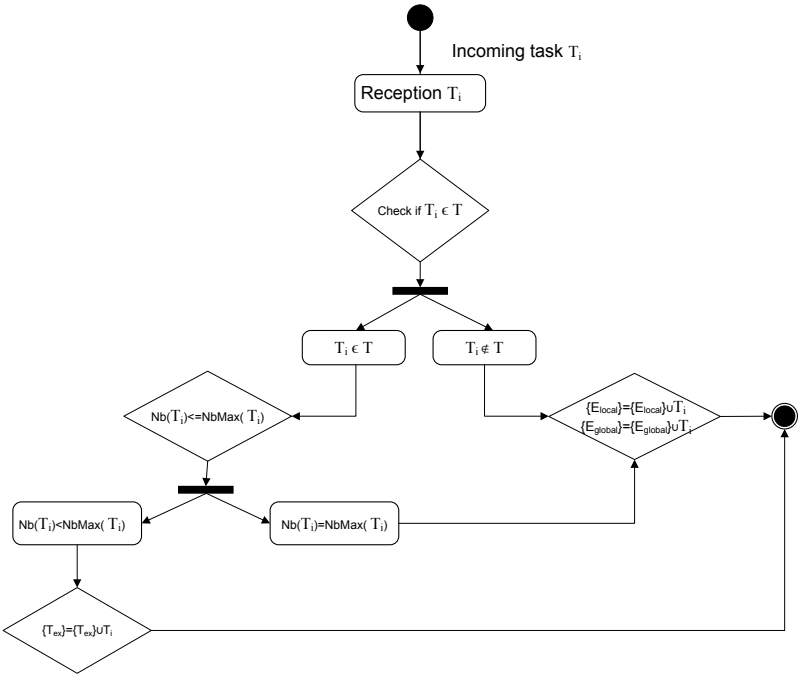


Fig. 6. Behavior that treats a received task

Bidding management by an agent: When an agent receives the structure $\{E_{global}\}$, it processes the bids one by one executing these steps:

▷ **Check** if the auction has already been treated
- **If** true
  Go to next auction.
- **Else**
  ▷ **Check** if it belongs to $\{E_{local}\}$ (it means that I am the editor agent)
  - **If** true
    1- It collects all offers and select the best one
    2- It deletes the auction from $\{E_{local}\}$
    3- Fill informations of the auction in $\{E_{global}\}$ with the assigned agent
    4- Go to next auction.
  - **Else**
    ▷ **Check** if an agent has been assigned
    - **If** true and the assigned agent is myself
      1- Add task to $\{T_{ex}\}$ and execute it
      2- Go to next auction.
    - **If** true and the agent assigned is not myself
      Go to next auction.

- **If** false
  - ▷ **Check** if I have already made an offer
    - **If** true
      Go to next auction.
    - **Else**
      - ▷ **Check** Condition1
        - **If** true
          - ▷ **Check** Condition2
            - **If** true
              1- $\{O\} = \{O\} \cup \{O\}_{Agi}$ with $\{O\}_{Agi} = null$
              2- Go to next auction.
            - **Else**
              1- The determinated offer amount=$\{O\}_{Agi}$
              2- $\{O\} = \{O\} \cup \{O\}_{Agi}$ with $\{O\}_{Agi} \neq null$
              3- Go to next auction.
        - **Else**
          1- $\{O\} = \{O\} \cup \{O\}_{Agi}$ with $\{O\}_{Agi} = null$
          2- Go to next auction.

The initiator agent waits to collect all offers before selecting the best one. Indeed, all neighbours must make an offer, eventually a null one if it can not execute the task.

We suppose that for each task type there is at least two agents can execute it. So, if an agent can not execute a received task and initiate an auction, we assure that there is an other agent can makes an oer.

Moreover, the offers amount is determinated using an algorithm defined by the application designer based on its constraints (here the available throughput).

An agent deletes an auction of its structure $\{E_{local}\}$ and the referenced offers when the auction is completed (this means that an agent was assigned and it added the task to its structure $\{T_{ex}\}$). As for the structure $\{E_{global}\}$, an auction will be deleted by the assigned agent after having added it to its $\{T_{ex}\}$.

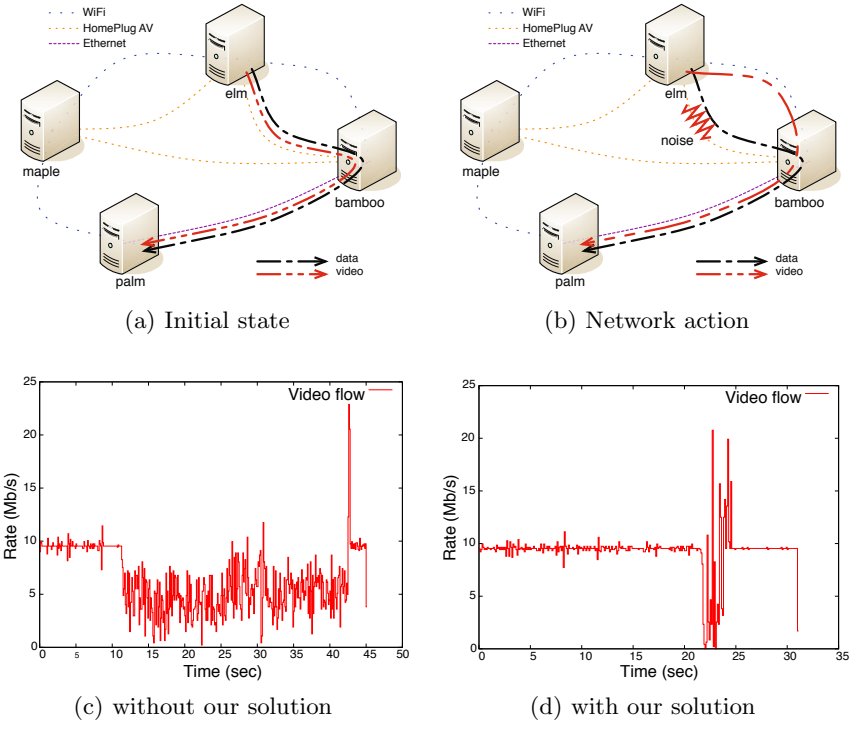
5 Application

Previous algorithms have been implemented and tested on the Ginkgo Platform using a set of unitary tests. Futhermore, the mechanism has been used by some members of the Ginkgo research team working on Home Networks. In this project, it was used to improve alternative routes computation as explained in [8].

In this context, a testbed has been implemented which support the scenario described with figures 7(a) and 7(b). HNID are connected using Ethernet, WiFi and PLC, and the network convey data, tv and voice flows.

At the beginning (figure 7(a)), there are, one data flow over the PLC, and one video flow going from *elm* to *palm* over the same medium. In the next step, we generate a perturbation of the PLC link[2]. The agent detects it and applies the

[2] Switch-on a lamp for instance.



(a) Initial state (b) Network action

(c) without our solution (d) with our solution

Fig. 7. Bandwidth evolution

alternative route which uses the WiFi link: this change is immediately applied and the user does not suffer from any scrambling. However, the data flow is not redirected since it allows easily some bandwidth reduction. The figure 7 illustrates the situation for the video flow with and without our agents.

In the figure 7(c), without agents, we can see, during the first 10 seconds, that the received rate is almost 9 Mb/s. At $t = 11s$, we start the perturbation, and we can see that the rate is down by 50 % : the quality perceived by the user is very poor. On the contrary, the same scenario is done with agent (figure 7(d)) : the rate is unstable during $3s$, after that the user can enjoy a good quality. Those $3s$ can be considered as important, but this is only an implementation concern, since agents are computing proactively alternative route (not adding delay) and the perturbation detection can be done in less than a second.

6 Conclusion and Future Works

The use of this auction based mechanism has provided some ease in its application, because the developer should in no case search a solution to its problems namely the computation of alternative route, but just determine task, and what represent a price. This prove that this proposed solution is generic and can be applied to many network problems.

As future works, we would like to investigate other applications like HandOver or, the virtualization [4,1], even GMPLS.

The use of distributed artificial intelligence allows to overcome the limits of networks solutions. In other words, effectively manage the complexity of growing networks.

As an immediate enhancement, we will do a statistical validation studying extreme cases, for example, additional requests to determinate the limit of this mecanism and improve the algorithm or propose another one complementary.

References

1. Abid, M., Ligocki, M., Molinier, L., Nguenguang, G., Pujolle, G., Gaiti, D., Zimmermann, H.: Practical handover optimization solution. In: IFIP Wireless Days, Dubai, UAE (2008)
2. Bullot, T., Khatoun, R., Hugues, L., Gaïti, D., Merghem-Boulahia, L.: A situatedness-based knowledge plane for autonomic networking. Int. J. Netw. Manag. 18(2), 171–193 (2008), doi:http://dx.doi.org/10.1002/nem.679
3. Clark, D., Partridge, C., Ramming, C.J., Wroclawski, J.T.: A knowledge plane for the internet. In: SIGCOMM 2003: Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, pp. 3–10. ACM Press, New York (2003), http://portal.acm.org/citation.cfm?id=863957, doi:10.1145/863955.863957
4. Fejjari, I., Pujolle, G.: An autonomic system for virtual network piloting. In: WNetVirt First Workshop on Network Virtualization and Intelligence for the Future Internet (2010)
5. Florea, A.M.: Bucharest university online course, http://turing.cs.pub.ro/auf2/html/chapters/chapter5/chapter_5_4_1.html
6. Freiderikos, V., Gaïti, D., Hamon, M.H., Insler, R., Jafré, P., Kortebi, A., Meyer, S., Molinier, L., Pujolle, G., Zimmermann, H.: Piloting home network with intelligent agents. In: Wireless World Research Forum 21 (2008)
7. Ketchpel, S.: Forming coalitions in the face of uncertain rewards. In: AAAI 1994: Proceedings of the Twelfth National Conference on Artificial Intelligence, vol. 1, pp. 414–419. American Association for Artificial Intelligence, Menlo Park (1994)
8. Molinier, L., Ghedira, E., Ligocki, M., Francois, R., Freiderikos, V., Kortebi, A.: Autonomic qos management and supervision system for home networks. In: 2009 2nd IFIP, Wireless Days (WD), pp. 1–6 (2009), doi:10.1109/WD.2009.5449648
9. Molinier, L., Ligocki, M., Pujolle, G., Gaiti, D.: Piloting the spanning tree protocol in home networks using a multi-agent system. In: 1st IFIP Wireless Days, WD 2008, pp. 1–5 (2008), doi:10.1109/WD.2008.4812910
10. Nguengang, G., Molinier, L., Boite, J., Gaïti, D., Pujolle, G.: Intelligent routing scheme in home networks. In: Springer (ed.) Home Networking, pp. 179–196 (2008)
11. Nwana, H.S., Lee, L., Jennings, N.R.: Co-ordination in software agent systems (1996)
12. Rahwan, I.: Interest-based negotiation in multi-agent systems. Tech. rep. (2004)
13. Walsh, W.E., Wellman, M.P.: A market protocol for decentralized task allocation. In: The Proceedings of the Third International Conference on Multi-Agent Systems ICMAS 1998, pp. 325–332 (1998)
14. Wooldridge, M., Jennings, N.R.: Intelligent agents: Theory and practice. Knowledge Engineering Review 10(2), 115–152 (1995), http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.55.2702

A Study of E-Government Architectures

Rim Helali[1], Ines Achour[2], Lamia Labed Jilani[1], and Henda Ben Ghezala[2]

[1] Tunis University, ISG, Computer Science Department, Lab. SOIE and RIADI-GDL
hl.rima@hotmail.fr, Lamia.Labed@isg.rnu.tn

[2] Manouba University, Ecole Nationale des Sciences Informatique, Lab. RIADI-GDL
ines_achour@yahoo.fr, hbg.hbg@gmail.com

Abstract. The success of an e-government initiative depends on different factors such as economic strategies, countries political and decisions initiatives, countries readiness to citizen connectivity, etc. We concentrate in this paper on architectural design of e-government systems according to a software engineering point of view which among all other considerations promises also the success of the final operational platform. In fact, architectural design is a key factor for a success of any system. The purpose of this paper is to study and analyze existing (software) architectures of e-government systems in order to have a better vision of the architecture underlying and characterizing an EGP (E-Government Platform). This is fundamental before proposing our own architecture, particularly for a federated project of research S2EG[1],conducted in the context of Tunisia country e-government initiative. In this presented work, we particularly want to highlight architectural design principles, the high level components that constitute the architecture, specifically the software components and the used technology.

Keywords: Software architecture, m/e-government, design principles, standards.

1 Introduction

Several countries all over the world initiate and conduct e-government projects and initiatives and several E-Government Platforms (EGP) exist nowadays. Some countries like Singapore, Japan, Switzerland, Austria and so on have good ranks (Source: IMD World Competitiveness Yearbook (WCY) 2010) and seem to prove a success in this field by good practices and results. E-government allows mainly the simplification of the governmental processes and the interaction between the citizens and the governmental organizations. The implementation

[1] The Tunisian project S2EG (www.soie.isg.rnu.tn/PRFS2EG) is a federated research project funded by the Ministry of Higher Education and Scientific Research (MESRS) and with the goal of establishing an M/E-Gov system including secure electronic transactions between citizen or company and the government system. This project involves three research laboratories (SOIE, RIADI and CRISTAL) and three socio-economic partners (the Tunisian Prime Ministry, the National Agency of Electronic Certification and the National Agency of Computer Security).

G. Babin, K. Stanoevska-Slabeva, P. Kropf (Eds.): MCETECH 2011, LNBIP 78, pp. 158–172, 2011.

of such a system will have to harmonize the treatment of the governmental documents, their electronic exchanges, security of the exchanged data and the cohabitation of heterogeneous systems. We argue that a success initiative of e-government depends on different factors to take into account but in this paper, we focus on software engineering aspects particularly on architectural design of e-government systems which play an important role in the success of a (software) system. Our main final objective is to build an e-government architecture in the context of Tunisian e-government initiative based on systematic, large scale reuse which seems to be appropriate for the so many applications proposed as services to citizens. But prior to this work, we fell that is very important to study existing e-government software architectures which consider seriously the importance of architecture design before building any operational platform. So, the main purpose of this paper is to synthesize the results of the study and to take lessons before proposing our own architecture which is not in the scope of this paper. Understanding e-government architecture framework among public sector organizations is a significant strategic step towards reliable and effective e-government adoption. In fact, e-government doesn't mean use of IT to offer electronic governments services but a good architecture design satisfying key requirements such as performance, reliability, scalability, portability, security and interoperability is fundamental for a success of such systems. A fundamental benefit of architecture for Information Management and Information Technology in any enterprise is adaptability faced to the inevitable changes in applications systems and other IT assets. As an example, with the mobile branch, many new services in public administrations and governments will grow up in this field. There are a number of architectural frameworks for modeling distributed systems coping with system heterogeneity and openness such as ISO/RM-ODP (Reference Model for Open Distributed Processing), OMG/CORBA (Object Modeling Group/ Common Object Request Broker Architecture), OSF/DCE (Open Software Foundation/Distributed Computing Environment). Several e-government architectures have been proposed in the past. Some are not good designed and practiced because they are only based on technology while others are based on real facts and strategic design principles. So, with the aim of conceiving a large scale reuse based software architecture for e-government, we conduct an in-depth study of a sample of e-government architectures which are not uniquely based on technological aspects but consider an abstraction of the system and qualifying it with various quality attributes as addressed by the software architecture discipline. We analyze each architecture in order to identify their dimension of variance, classify the architectures along these dimensions to better characterize e-government architectures compared with other distributed architectures such as those of e-commerce or e-learning and son on. Interestingly, even though these architectures appear to deal with the same problem, they in fact differ significantly from each other. Some of the features that we put in place and that distinguish between them include:

- Number of architectural principles: ideally includes sufficient information to allow high level analysis and critical assessment for making the appropriate decisions.
- Use of specific architectural standards.
- Use of explicit support for architectural design: notations for architectural representation (architecture description languages and tools for parsing, displaying, compiling, analyzing, or simulating architectural descriptions).
- Exploiting commonalities in e-government applications to provide large scale reusable framework specifically product line families [7] which promises improvements in productivity, time-to-market, quality, and cost.
- Used approach for architecting a system (holistic or not).
- Number of described architectural views: each view captures some aspect of the system.
- Scope of development (country, federal, governmental agencies, cross-border, etc.).
- Field addressed: e-administration or e-government in general.
- One stop portal.
- Client centric architecture.

In section 2, we present a sample of e-government architectures. A comparison of the studied architectures is given in section 3 followed by a discussion. The highlighting of the principle features (attributes) of an e-government architecture is given in section 4. A conclusion summarizes our work and gives some ideas about the e-government architecture that we put in place in the context of the federated project of research S2EG but which is not the subject of this present paper.

2 A Sample of E-Government Architectures

We have investigated several e-government architectures and present here a sample of 7 architectures. These latter are chosen in the sample because: 1) detailed documentation about architectural aspects of their e-government platform are available, 2) the architectures are built in different contexts (a country, a state and federated research project), 3) some consider mobility and others support only e-government. Our sample concerns mainly developed countries where just one deals with the Jordan developing country. We explain this by the fact that most papers describing e-government in developing countries focuses on economic strategies, statistics about connectivity of citizens and the use of e-government services, and operational e-government platforms.

Archi 1: Architecture for mobile government
The architecture proposed by [2] is a system offering citizens the opportunity to establish administrative transactions using an m-gov portal. The portal is the single point of access to services. The principles of the architecture are essentially three: accessibility, integration, transparency. As shown in Figure 1, several components are integrated to satisfy the citizen's request made via the m-gov portal:

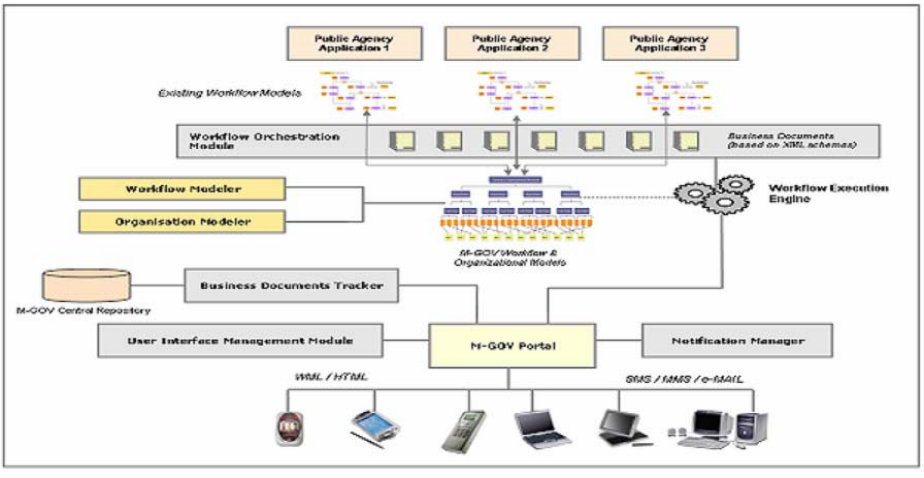


Fig. 1. Architecture for mobile government [2]

- User Interface Management Module: its role consists on synthesizing and deploying the user interface on any mobile device used by the user.
- Business Documents Tracker: offers users the ability to track documents as they pass through different stages of the workflow.
- Notification Manager: is responsible of sending alerts and notifications for the workflow participants (citizens or government employees) via SMS, MMS or emails.
- Workflow Execution Engine: runs workflow models defined by the workflow modeler. In fact, this module enables, through invocations of Web services, applications operating in the various public agencies and controls their execution.
- Workflow Modeler: defines administrative workflows by specifying the public agencies involved in the workflow and the roles of their staff.
- Workflow Orchestration Module: ensures consistency and integration of various public agencies workflows.
- Public Agency Applications: represents information systems and applications already deployed in public agencies that are usually heterogeneous and difficult to interoperate.

Archi 2: Geneva state e-government
According to [9], the author presents a system approach and defines principles for architecting a system which must sustain the entire e-government activity of a mid-level public authority (Geneva). The four principles are: Legality, Responsibility, Transparency, and Symmetry.

Statement of the Legality principle (LP). Any operation suggested to a user on an interface of the EGP and all the consequences of the execution by the

user of this operation must be legal and respect the users legal and civil rights within the jurisdictions under which the referential operates.

Statement of the Responsibility principle (RP). Each operation executed on the EGP can be attributed to a unique identified legal personality. This legal personality is legally responsible for the execution of the considered operation and for all its publicized and certified consequences.

Statement of the Transparency principle (TP). Organizational characteristics (of actors) which are not explicitly necessary to perform an operation on the EGP are not reflected in that operation.

Statement of the Symmetry principle (SP). Any function that is necessary for an EGP to operate correctly, but that is not directly determined by a mandate of the state, should be implemented on the EGP, if at all, in a way that an external service provider can furnish the service.

The architecture of an E-Government system is organized into three layers (see figure 2):

- Front-end: supports the user's request to be processed in the back-end.
- Back-end : integrates several components to satisfy the request of users:
  - Content Management: ensures consistency of document contents.
  - Document Management : supports documents management.
  - Communication Management: ensures the communication of the different participants which is based mainly on the exchange of documents.
  - Authorization Management: checks if the permission is granted or not to a user to perform an operation.
  - Personnel Data Mirroring: is responsible for the proliferation of personal data to prevent their loss.
  - Workflow Repository: includes all workflows that support the different business processes.
  - Workflow Engine: allows electing the most appropriate workflow to satisfy the user's request.
- Agency Legacy Infrastructure: represents the information system already established in the governmental agency.

Archi 3: One-stop government portal architecture
The One-stop government portal architecture proposed by [1] focuses on interoperability and integration of different governmental services based on knowledge management. It consists on a basic structure of a generic e-government one-stop portal based on a SWS (Semantic Web Services) infrastructure. This architecture is organized into three layers (see Figure 3):

- User Interaction Layer: represents the front-end system. It permits to the users to identify the "life events" appropriate to their needs and collect the necessary information needed to execute the requested service.
- Middleware Layer: represents the core of the system. It's based on :

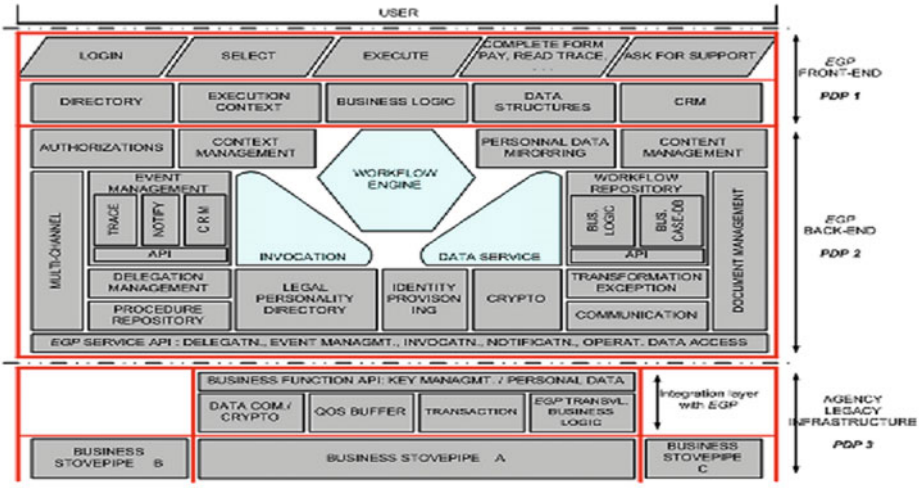


Fig. 2. E-Government architecture [9]

- Life event Manager : responsible of the semantic description, publishing and updating "life events" to provide citizens with a personalized list of available services;
- Internet Reasoning Service (IRS III) : responsible of the description, identification and invocation of multiple and heterogeneous web services, based on WSMO (Web Service Modeling Ontology);
- A knowledge model composed of three ontologies: an e-government ontology, a "life event" ontology and a service ontology.

– Service Layer: ensure the execution of services for a "life event". Each public administration (PA) offers a service that is semantically described by the IRS-III module of the middleware layer.

Archi 4: The Jordan electronic Government Architecture Framework (e-GAF)

The reference model for the Jordon e-Government is captured through this set of artifacts:

– An enterprise architecture for the e-Government central platform.
– A reference architecture framework for ministries, to be developed follows a framework based on the US Federal Enterprise Architecture Framework (FEAF).
– An interoperability framework (GEFI)
– Governance framework
– A set of supporting standards and guidelines.

The e-government central platform architecture proposed in [3] is composed by the following components as mentioned in figure 4:

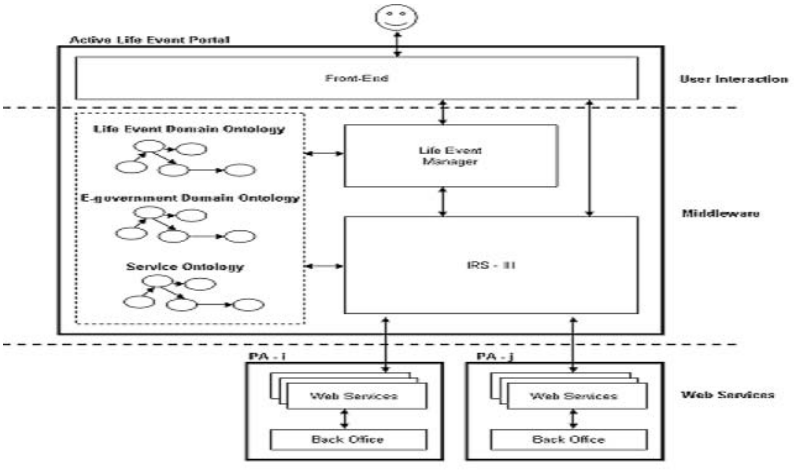


Fig. 3. One-stop government portal architecture [1]

- Access and delivery services: portal, contact center and collaborationrkflow for contractors.
- Business integration services: integration hub, workflow enabler
- Shared services: notification, content management and e-payment
- Platform services: connectivity, identity management
- Administrative services

The architecture adopts 24 e-government architectural requirements such as accessibility, business event-driven systems, defined authoritative sources, security, requirements change, etc.

Archi 5: The architecture of an European e-Government Project
The architecture of " on-line one - stop Government ", presented in [4], realized within the European eGOV project, is represented in the form of a portal, offering to citizens the possibility to have access to public services by a single entrance point to all services and information of government and administrations. It's, in fact, a global entry point to different services and information on local and national institutions The consortium of the project consists of 10 partners coming from Austria, Finland, Germany, Greece and Switzerland reflecting different forms of government and public administrations throughout Europe. Further, the partners represent a balanced mixture of public and private research institutions, local and global public administrations as well as technology providers. Design principles: one global access point with different devices so mobility, adaptability, integration, accessibility, legal issues, laws and prescriptions have to be clarified and updated. The architecture consists of :

- A portal which presents the only point of access to the governmental services offering a number of advanced characteristics as the access through

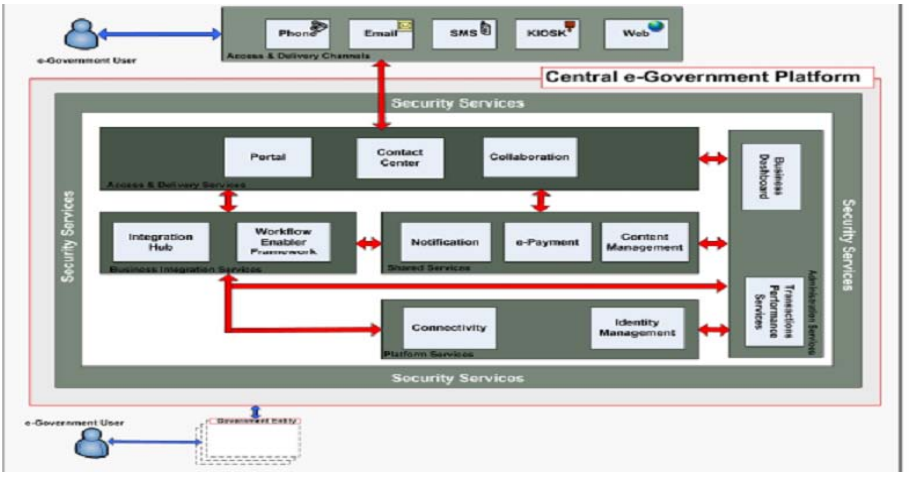


Fig. 4. The Jordan e-Government portal [3]

various devices including mobile devices, the customization(personalization), numeric signatures, and multilingual content.

- A national Service Repository (SR) and some local Service Repositories offering data and information related to public services. These warehouses also supply a structure for the transactional services and integration interfaces for the legacy systems already used in the public administrations.
- A Service Creation Environment (SCE) which is a set of tools serving as front-end for the SR and which allows the maintenance and the update of the public services existing in the SR.
- The GovML language (Governmental Mark-up Language) which connects the portal to all the public local services. It is based on XML and offers a common, flexible and stretchable syntax for the public sector.
- The GovML filter which allows the SR to be easily reached independently of the formats of data warehouses. Appropriate filters allow translating a given format to the required GovML format.

Archi 6 : European Commission e-mayor project (e-mayor, 2004)
The proposed e-government architecture [6] defines ten e-government architectural requirements: interoperability, scalability, transparency, cross boarder characteristics, security and trust, compatibility with existing infrastructures, user-friendliness and accessibility, cost considerations, limited training, and mobility aspects. These architectural attributes are mapped with RM-ODP architectural attributes and uses this standard which considers the following 5 architectural viewpoints: 1)Enterprise viewpoint: the policies which define the behavior of an object in the system and the system's purpose of operation and scope, 2)Information viewpoint: presents and analyses various information objects that will be used by the system, 3) Computational viewpoint: divide the

computational functionalities in distinct packages and depicts their interconnexion and collaboration based on the interfaces exposed; focuses on the way distribution of processing is achieved, 4)Engineering viewpoint: the way different objects of the system use to communicate with each other and the resources that are needed (channels) and 5) Technology viewpoint: the selected technology of a system (channels becomes RMI , TCP).

Archi 7: The Government of Canada, Federated Architecture
The Government of Canada's Federated Architecture [5] is intended to guide the development and construction of the government's common information management and information technology (IM/IT) infrastructure. This architecture defines 13 principles: Reduce integration complexity; Holistic approach; Business event-driven systems; Defined authoritative sources; Security, confidentiality, privacy and protection of information; Proven technologies; Total cost of ownership; Plan for growth; Adopt formal methods of engineering; Extended information and services environment; Multiple delivery channels; Accessible government; and Robustness. The approach taken divides the high level definitions of technology domains groups of related technology enabled capabilities.: 1) presentation, 2)application: combines business processes with data and technology to enable the implementation of business activities (interoperability of applications), 3)services, 4) platform,5)network, 6)information management, 7)system management and 8) security. The architecture components comprise common components needed to ensure that the government meets its on line service delivery goals. Other components are specific to "subgroups" of departments and agencies that share similar needs for which common IM/IT solutions are appropriate.

3 Comparison of the Architectures Sample

Table 1 presents the commonalities and variabilities of the 7 presented architecture according to the fixed features as mentioned in the introduction.

Discussion
As a first notice, we remark that all the architectures don't have the same level and most present a conceptual view with different kinds of design principles varying from philosophical ones (such as the 4 principles presented in archi 2 that consider a system approach for the architectural foundation), conceptual and strategic ones and technological principles characterizing distributed systems in general. All the architectures provide client-centered electronic services. In fact, it is necessary to approach the design of an EGP from the client's side because the client, i.e. society as a whole, is not an enterprise. Some consider one stop portals as an entry to services while others not. The major fields addressed concern e-government in general. The scope of development concerns one state, one country, several countries as European ones, or Federal governments where others don't precise the scope. Some architectures see government

Criterion \ Archi	**Archi1**	**Archi2**	**Archi3**	**Archi4**	**Archi5**	**Archi6**	**Archi7**
Number of architectural principles	not clearly mentionned	4	not clearly mentionned	24	6	10	13
Use of a specific architectural standards	no	no	no	TOGAF (The Open Group Architecture Framework) [10]	no	ISO/RM-ODP [6]	John Zachman's Framework for Information Systems Architecture [11]
Explicit support for architectural design	no	no	no	yes	no	yes	no
Exploiting commonalities	no	no	no	no	no	no	no
Approach	not holistic	holistic	not holistic	holistic	not holistic	holistic	holistic
Number of described architectural views	1 conceptual view	1 conceptual view	1 conceptual view	4 Business viewpoint Data viewpoint Applications viewpoint Technology viewpoint	1 conceptual view	5 Enterprise viewpoint, Information viewpoint, Computational viewpoint, Engineering viewpoint and Technology viewpoint	1 conceptual view
Scope of development	General: not for a specific project	state of Geneva (national and international): mid-level public authority	General: not for a specific project	Jordan Government	European countries	eMayor project between European municipalities	Canada federal Government
Field addressed	e administration	e government in the large	e government in the large	e government in the large	e administration	e government in the large	e government in the large
One stop portal	yes	no	yes	yes	yes	no	no
Client-centered electronic service	yes	yes	yes	yes	yes	yes	yes

Fig. 5. Comparative table of E-Government architectures
Archi 1: Architecture for mobile government- Archi 2: Geneva state e-government - Archi 3: One-stop government portal architecture - Archi 4: The Jordan electronic Government Architecture Framework - Archi 5: The architecture of an European e-Government Project- Archi 6 : European Commission e-mayor project - Archi 7: The Government of Canada, Federated Architecture

as a whole, which tends to favor top-down e-government design and to consider a holistic approach. A simple reason for the need to integrate architecture is the fragmentation of e-government systems that have often been organized vertically around departments (silos). This increases the need for vertically and

horizontally integrated architectures addressing the communication between systems within and between departments and organizations. We note that architectures designing one stop portals don't adopt a holistic approach. This leads to the definition of incremental strategies for e-government but over time, most incremental projects encounter difficulties because they don't consider e-government systems as a whole. The mobility aspect is considered just by some architectures. Only three architectures (archi4, archi 6 and archi 7) use specific architectural standards which is a good initiative because they permit a better reuse of design principles characterizing open distributed systems but adding new philosophical and strategic ones specific to e-government is fundamental. But in archi 2, even if the author doesn't speak about any use of architectural standards, the design principles concentrate on ones specific to e-government and even the security aspect is not considered technically but the 4 presented philosophical principles induce solid security aspects. Only archi4 and archi 6, consider explicit support for architectural design which is profoundly recommended for building good architectures. Finally, no architecture exploits commonalities in specific domains to provide reusable framework for product families in e-government. According to the implementation view, we note that:

- Each is articulated in three layers: the front-office, the back-office and the applications of the governmental organizations.
- The implementation architectures are based on Web services and/or on workflows or GovML language where some use the concept of ontology.
- Most of the architectures assure the follow-up of tracks of the governmental documents and offer the possibility of sending notifications to the citizens.

We can't say that one architecture is the best but we can build an architecture which takes into account the best practices and provides a good architecture suited for e-government systems. We strongly think that a good architecture must present different views and determine the relations between them. We also recommend a construction of higher level architecture which is a line of services and exploit the advantages of product-families. In fact, governments have certainly a long list of services to provide and exploiting commonalities and variances between a set of services will provide a good abstraction and gain in development costs and time to market (large number of citizen, able to host increasing number of e-services which we consider as software applications) with a good return on investment. In the sequel, we concentrate on the different design principles proposed by the sample of architectures and we try to classify them and to consider all of them for the design of an e-government architecture based on Service Oriented Product Lines (SOPL) (which is not presented in this paper). The mobility aspect becomes a real requirement that must be present in any e-government system which leads to m-government without forgetting security new issues. Note also that a one global access point with different devices is strongly recommended.

4 Principle Features (Attributes) of an E-Gov Architecture

Our review of the literature of e-government architectures allowed us to list a set of characteristics defining software architecture for E-Government systems. These attributes must be addressed to policymakers, i.e. the people confronted with the real problems of e-government (complexity and scale, i.e. costs; trust and expectations of citizens, i.e. choice and added value; dissemination and impact of technology i.e. change in society), and to users. We group these characteristics into intrinsic characteristics and extrinsic characteristics.

- The intrinsic characteristics of the architecture are characteristics appropriate to the architecture. They concern the specificities of the architecture and its integration with other systems:
  - Interoperability and integration: integration of different governmental information systems in order to cooperate with each other.
  - Security and trust: the security is an essential criterion for the architecture to provide confidence to the users of these systems.
  - Flexibility: indicates the ease with which new components are integrated within the architecture and the possibility of its evolution.
  - Compatibility: indicates that an e-Government architecture has to allow the use and the communication with the existing infrastructures. Thus, such architecture must provide a compatibility layer with existing systems in order to be able to use their data. In a context of E-Government, this compatibility interests especially the integration of the legacy systems and the cohabitation of heterogeneous environments of the various governmental organizations.
  - Traceability: indicates that we can detect for every operation made the steps by which it passed.
  - Symmetry: indicates that any function which is necessary for the system to work correctly, but which is not directly decided according to a mandate of the state should be implemented in a way that it can be provided by an external service provider.
  - Cross-border characteristics: refers to the international communication. In frontiers services, there has an exchange of information, data, or documents between citizens and public administrations (C2G, G2G) in an international context and through administrative borders.
  - Scalability: large number of citizens and able to host increasing number of e-services.
  - Legality: any operation executed on m/e-government system must be legal and respect the users legal and civil rights within a given jurisdiction.
  - Cost considerations: minimization of deployment and operations small organization don't have the same resources in terms of finance and personnel.
  - Limited training of e-government employees.
- The extrinsic characteristics in the architecture: are characteristics that interest rather the user of this architecture:

- Privacy: an E-Government architecture has to offer mechanisms dedicated to the protection of the user's privacy.
- Accessibility: this characteristic results from the fact that the governmental organizations have to cover a maximum number of users. So, it is necessary to propose assistance for the foreign or handicapped citizens.
- Transparency: the operational treatment must be hidden and should not be issued to the end users.
- Mobility: this aspect offers to the citizens the use of governmental services without needing either to move or to respect the administrative schedules. It involves the omnipresence of the governmental services and the facility of access to these services.
- Responsibility: every operation executed on the system must be attributed to a unique identified legal personality. This legal personality is legally responsible for the execution of the considered operation and for all her consequences made public and certified.

We think that all the mentioned architecture attributes are important for designing an e-government architecture. All the presented architectures don't support all of them.

Archi / Criterion	**Archi1**	**Archi2**	**Archi3**	**Archi4**	**Archi5**	**Archi6**	**Archi7**
Interoperability and integration	x	x	x	x	x	x	x
Security and trust	x	x	x	x	x	x	x
flexibility			x	x	x	x	x
Compatibility	x	x	x	x	x	x	x
Traceability	x	x		x			
Symmetry		x					
Cross-border Characteristics	x					x	
Privacy		x		x			x
Accessibility	x			x		x	x
Transparency	x	x	x	x		x	
Mobility	x		x			x	
Responsibility	x	x	x				
Cost considerations				x		x	
Limited Training				x		x	
Scalability				x		x	
Legality		Explicitly defined					

Fig. 6. Comparative table of E-Government architectures attributes
Archi 1: Architecture for mobile government- Archi 2: Geneva state e-government - Archi 3: One-stop government portal architecture - Archi 4: The Jordan electronic Government Architecture Framework - Archi 5: The architecture of an European e-Government Project- Archi 6 : European Commission e-mayor project - Archi 7: The Government of Canada, Federated Architecture

5 Conclusion

E-government involves different stakeholders and disciplines. Successive initiatives address different economic strategies, political aspects, country readiness, citizen connectivity, different objectives according to countries, technological aspects and so on. In this paper, we take the software architecture point of view of an e-government system which among all other considerations promises also the success of the final operational platform. The principle purpose of this work is to study and characterize e-government architectures with multiple views and addressed to different stakeholders where the citizen is at the center of the provided e-governmental services on the web or via mobile devices. To do so, it was fundamental to study the literature about e-government architectures. Even if we investigate a long list of architectures, we present here a sample of them. We have identified their dimension of variance, classify the architectures along these dimensions to better characterize e-government architectures. Even though we know that different design structures exist for different organizations with different objectives, we think that a lot of architecture attributes are similar to all of them. Different services (here software applications that provide services to citizens) and associated processes share similar features and can be thought in terms of product line software reuse. Our final objective is to propose an architecture based on the adoption of a systematic and large scale reuse and specifically the adoption of SOPL [8]. In fact, SOA is largely used by different e-government platforms and SOPL combines SOA and Product line Engineering which promises better time to market and quality of software applications. Our actual work concerns this aspect and we are now developing the main SOPL phases which are domain engineering for the production of reuse based e-government services and application engineering for deriving such services (the underlied software applications) in the context of family of applications. Our long term research objective is to detail the conception of the architecture, to implement it and to evaluate it in the context of the Tunisian PRF project.

References

1. Gugliotta, A., Cabral, L., Dominigue, J., Roberto, V., Rowlatt, M., Davies, R.: A Semantic Web Service-based Architecture for the Interoperability of E-government Services. In: Proceeding of the International Workshop on Web Information Systems Modeling, Sydney, Australia (2005)
2. Gouscos, D., Drossos, D., Marias, G.: A Proposed Architecture For Mobile Government Transactions. In: Proceedings of Euro mGov 2005, 1st European Mobile Government Conference, Brighton, The United Kingdom (2005)
3. Jordan e-Government Architecture Vision, `http://www.jordan.gov.jo/wps/portal/ut/p/c5/`
4. Glassey, O.: One-stop Government Architecture based on the GovML Data Description Language. In: 2nd European Conference on EGovernment (ECEG 2002), St Catherine's College, Oxford (2002)
5. Government of Canada Federated Architecture, `http://www.tbs-sct.gc.ca/inf-inf/documents/iteration/iteration01-eng.asp`

6. Kaliontzoglou, A., Meneklis, B., Polemi, D., Douligeris, C.: A formalized design method for building E-Government Architectures: system developed as part of the European Commission e-mayor project. In: Electronic Government: Concepts, Methodologies, Tools, and Applications. Information Science Reference. Hershey, New York, vol. I, pp. 569–591 (2008)
7. Pohl, K., Bckle, G., Linden, F.: Software Product Line Engineering: Foundation, Principles and Techniques. Springer, Heidelberg (2005)
8. Robert, K., Cohen, S.: Workshop on Service-Oriented Architectures and Software Product Lines - Putting Both Together. In: 12th International Software Product Line Conference, Limerick, Ireland (2008)
9. Sandoz, A.: Design Principles for E-Government Architectures. In: E-Technologies: Innovation in an Open World. Lecture Notes in Business Information Processing., vol. 26, pp. 240–245. Springer, Heidelberg (2009)
10. TOGAF version 9, http://www.opengroup.org/architecture/togaf/, http://www.bizzdesign.com/index.php/ea-and-bpm/togaf
11. The Zachman Framework: A Concise Definition. at Zachman International, http://www.zachmaninternational.com/index.php/the-zachman-framework

Model-Based Engineering of a Managed Process Application Framework

Abel Tegegne and Liam Peyton

School of Information Technology and Engineering,
University of Ottawa, Canada
abel.tegegne@gmail.com, lpeyton@site.uottawa.ca

Abstract. Organizations use managed process applications to improve operational efficiency by monitoring processes and providing performance indicators that can be used to evaluate them. General purpose application development tools and frameworks build applications that are compiled from manually crafted or tool generated source code. These approaches do not support run-time configurability of a managed process nor do they provide specific systematic support for integrated data collection, monitoring and reporting within a managed process. To facilitate configurability and integrated data management, a model-based approach to specifically engineer managed process applications is needed. This approach models all aspects of a managed process application including workflows, roles, entities, events, alerts and performance indicators. In this paper, we describe the engineering of a model-based application framework for managed processes used to implement a palliative care managed process application for severe pain management.

Keywords: managed process, application framework, model-based engineering, performance indicators, run-time configuration, data management.

1 Introduction

There is an increasing expectation on organizations to improve operational efficiency by monitoring processes and providing metrics that can be used to evaluate processes according to the overall goals of the organization. Increasingly, managed process applications are being developed that coordinate linked activities performed by people and systems, and which collect data in order to monitor and measure performance.

Current application development is supported by generic application development tools and frameworks to build applications that are compiled from manually crafted or tool generated source code. These approaches are not suitable for managed process applications because they do not support run-time configurability of a managed process nor do they provide specific systematic support for integrated data collection, monitoring and reporting within a managed process. Every time a managed process application is modified or created, application elements such as web forms, classes, services, database schemas, alerts and reports have to be defined and compiled to create the managed process application. Business Process Management (BPM)

G. Babin, K. Stanoevska-Slabeva, P. Kropf (Eds.): MCETECH 2011, LNBIP 78, pp. 173–188, 2011.

frameworks leverage business process models to provide run-time configurability of workflows but lack an integrated approach to data management. To facilitate configurability and integrated data management, a model-based approach to specifically engineer managed process applications is needed that models all aspects of a managed process application including workflows, roles, entities, events, alerts and performance.

In this paper, we describe the engineering of a model-based application framework for managed processes that enables run-time configuration and monitoring of managed processes based on declarative model definitions. This includes

- Identifying the set of models needed to declaratively define a managed process application.
- Defining an event-driven service-oriented architecture with pre-configured services and components for interpreting and executing model-based definitions of managed processes.
- Implementing a palliative care managed process application for severe pain management to illustrate and evaluate our Managed Process Application Framework (MPAF).

There are two main contributions in this paper. First, it illustrates the potential utility of specializing application frameworks for particular classes of applications. In particular, it identifies a specific class of applications we call managed process applications which are used to collect data and monitor business processes. An application framework allows both the business process and the data collected to monitor it to be flexibly configured and modeled. Second, this paper illustrates the potential advantages of model-based approaches that interpret or execute models, in contrast to model-driven approaches which generate code. The most important advantage being that the application can be maintained solely in terms of business level models of process and data independent of low level code.

2 Background

A managed process application is a process that is controlled by an automated system and contains a set of coordinated activities conducted by both people and systems to accomplish a specific organizational goal [21]. A managed process application is expected to support the ability to collect data and to execute, monitor, manage and facilitate business process optimization in a flexible manner [4].

Traditional code centric application development focuses on application data and its operation. Application data includes database and object model while its operations include application logic in the form of functions, methods and APIs. Traditional code centric application development concentrates on architectural and technical implementation in order to better organize components, layers and services to realize an application. Knowledge about how an organization performs a task and the conditions that must be fulfilled to perform the task are sprinkled across many applications and different layers within a single application in the form of hard coded application logic. In the face of frequently changing business processes, keeping such applications up to date is an expensive and time consuming task because it requires

updating hard-coded application logic and undergoing traditional application development cycles[20][6][24].

Another approach for application development is to put business processes that an organization has to perform, at the center of the application development process. "A business process is a set of activities that are performed in coordination in an organizational and technical environment. These activities jointly realize a business goal. Each business process is enacted by a single organization but it may interact with business processes performed by other organizations" [24]. Business processes enable organizations to better understand who is responsible for doing what, what happens next in a series of steps to perform a task, where decision-making activities are performed, and what rules are used for these decisions.

Putting business processes at the center of the application development effort enables processes to be clearly defined and enables the process definitions to be consumed by an application that can interpret them into business processes. This provides an additional layer of flexibility because it provides the ability to modify a process definition in a controlled manner and the consuming application will reflect these modifications accordingly. It also establishes an environment for configurable business processes that can easily be modified to address new requirements and enables organization to monitor and measure the performance of their business processes [23].

In the following sections, we give the relevant background on our approach to application development for managed process applications, which centers on building a model-driven application framework to support business process management that leverages a pre-configured service oriented architecture and integrated data management for monitoring and reporting.

2.1 Model Driven Engineering and Application Frameworks

Model Driven Architecture is OMG's approach for separating the specification of how a system operates from the details of how it is implemented on a particular technical platform [17]. It uses a higher level modeling language, such as UML, to specify a model of an application's business functionality and behavior, independent of the target platform. These models can then be transformed into another set of models, application code and components that are specific to a particular platform [22][20]. AndroMDA is an extensible open source Model Driven Architecture (MDA) framework that uses plug-in components called cartridges to transform model elements into actual source code [1]. There is no specific support for business processes or workflow, although hard-coded application flow can be specified and generated from UML activity diagrams. A reference model and survey of model driven approaches is provided in [19].

Business Process Execution Language (BPEL) is a proposed standard [16] for explicitly modeling business processes as a "composition, orchestration and coordination" of a set of web services [9]. Such models are typically executed directly by a BPEL engine with the context of a Service Oriented Architecture (SOA). Microsoft [14] provides tools and a framework for defining a service oriented architecture and interpreting workflow models that coordinate processes across both human interaction and web service processing.

2.2 Service Oriented Architecture

Software architecture is "the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them"[2]. It also serves as high-level structural view of a system to provide a common understanding of the components of a system and their interaction [5]. Service Oriented Architecture (SOA) is an application architecture that is composed of reusable services with well-defined interfaces. These services can be distributed, hosted and owned by different organizations across disparate domains of technology [7]. These services can be invoked in a defined sequence to realize business processes [11].

2.3 Integrated Data Management

Integrating data management within a SOA can be challenging. In managed process applications, data is continuously monitored and inspected to provide visibility and control into business processes. An event can be defined as a record of something that happened in the course of executing a business process [13]. Event data is an important aspect of a business process because it is produced only when some type of a business activity is performed [3]. Indicators can be used to measure the effectiveness of a business process based on event data. They can also be used to generate alerts when the performance of a process strays away from the expected KPI values by more than a certain predefined threshold amount [15]. Monitoring of business processes has been proposed through an agent-based architecture to analyze behavior in an SOA [8]. And model-driven approaches have been proposed to generate dashboards for viewing indicators [18].

3 Engineering a Palliative Care Managed Process Application

In our approach, we explicitly modeled all aspects of a managed process application including workflows, roles, entities, events, alerts and performance and have created a run-time engine to execute those models within a pre-configured service oriented architecture that provides integrated data management to support alerting and the reporting of performance indicators.

In order to develop and evaluate our Managed Process Application Framework (MPAF) we used a case study based on an existing palliative care managed process application for severe pain management called PAL-IS (Figure 1). A prototype implementation using MPAF was compared against both the original PAL-IS implementation, and another prototype implementation using AndroMDA (a generic model-driven architecture framework).

Severe pain management is a good example of a managed process application that has well defined workflow, roles, events, entities, alerts and performance indicators where the patient submits pain assessments, nurse provides consultation to assessments and severe pain alerts, case worker monitors the care delivery process and team/unit administrator sets up performance indicators to measure overall quality of the care delivery process.

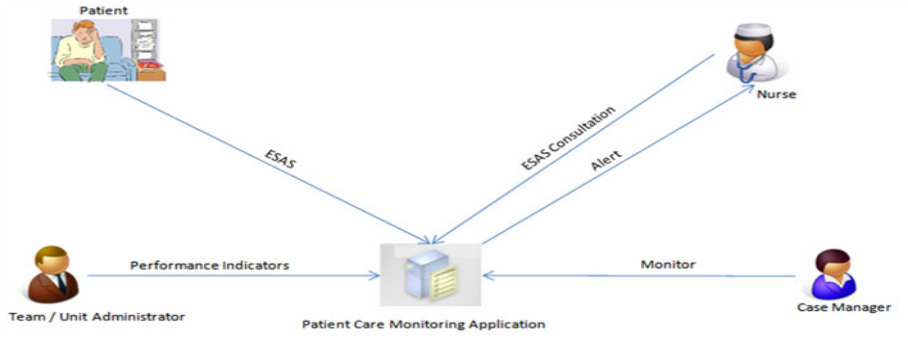


Fig. 1. Example Managed Process Application

As shown in Figure 1 on the previous page, there are four users for the managed process application: the patient, nurse, case manager and the team/unit administrator. The patient submits pain assessments using the Edmonton Symptom Assessment System (ESAS) a few times daily. Pain assessment can be submitted either using a web form that the patient has access to or the patient can call the palliative care nurse and communicate the pain assessment over the phone. After the pain assessment is recorded, the nurse will provide consultation over the pain assessment. Based on the nurse's consultation some action may be taken such as a sending a recommendation for reviewing the patient's prescriptions can be sent to the patient's doctor. Upon receiving the nurse's recommendation, the doctor may issue a new prescription, to alleviate the patient's pain to ultimately improve the patient's quality life. The case manger will also follow up with the patient regularly to ensure that the patient is receiving quality care. The team/unit administrator's responsibility is to ensure that the quality of care is adequate[12].

3.1 PAL-IS Implementation

The original PAL-IS application was implemented as a typical web application using the Microsoft ASP.NET framework. With a simple straightforward architecture where the user interface contains application logic and data access, it is easy to get an overview of the overall source code structure and common user interfaces functionalities. However, it is difficult to understand or even recognize the essential aspects of the managed process application in terms of the business workflow and the integrated data management needed to track alerts and report on indicators. When using this type of implementation, there is often code duplication and it is difficult to isolate a single component for modification as the business rules and data model logic is spread across the application.

3.2 AndroMDA Implementation

Our re-implementation using AndroMDA generated a managed process application for the Microsoft ASP.NET framework similar to the PAL-IS application. However

the architecture of the generated application was more sophisticated and the steps followed were quite different:

1. **Select and/or define the architecture for the application.** We selected SOA from the default application architectures supported by AndroMDA.
2. **Create the platform independent model (PIM).** PIM models are defined using UML and are stored in an XMI format which will later be processed by AndroMDA.
3. **Apply appropriate UML stereotype decoration.** We used "<<Service>>", "<<Entity>>" and "<<Enumeration>>". A class with the stereotype "<<Service>>" becomes a service class exposed as a web service using ASP.NET XML Web. A class decorated with the "<<Entity>>" stereotype generates a set of classes that includes an entity base class, DAO base class and implementation classes.
4. **Generate the code.** AndroMDA generates code that defines the overall structure of the application, with specially designated places where the application developer can fill in or modify code.
5. **Customize the generated code to implement application.** AndroMDA only generates the skeleton source code with implementation for well known and repetitive tasks, such as data access and service interfaces. Once the code is generated, the application developer customizes the generated application to implement and hardcode business logic and workflows. If there is a change in requirements or design, the overall application must be generated and compiled again.

Using AndroMDA, it is not as easy as PAL-IS to get an overview of the user-interface functionalities, but it is quite straightforward, using the UML diagrams, to get an understanding of the application structure and logic even though the architecture is more sophisticated. However, the sophisticated architecture and generated code, makes it perhaps even more difficult to understand or even recognize the essential aspects of the managed process application in terms of the business workflow and the integrated data management.

3.3 MPAF Implementation

In our MPAF implementation, we identified and modeled the essential elements of the managed process application for severe pain management in terms of workflow, roles, events, entities, alerts and performance indicators. And those models were interpreted by a service oriented architecture with pre-configured services. As shown in Figure 2, there was a clear and well understood relationship between separate interfaces for managing tasks created by event and alert-triggered workflows, entity-based data collection that mapped into an event-based data model from which reports on performance indicators could be made.

The service-oriented architecture was as sophisticated as that of AndroMDA, but tuned and pre-configured for managed process applications. This is described in section 4. There was no generated code, no customization of code and in fact no writing of code other than creating the presentation layout for forms and reports. Instead, the application was defined by building the appropriate application specific

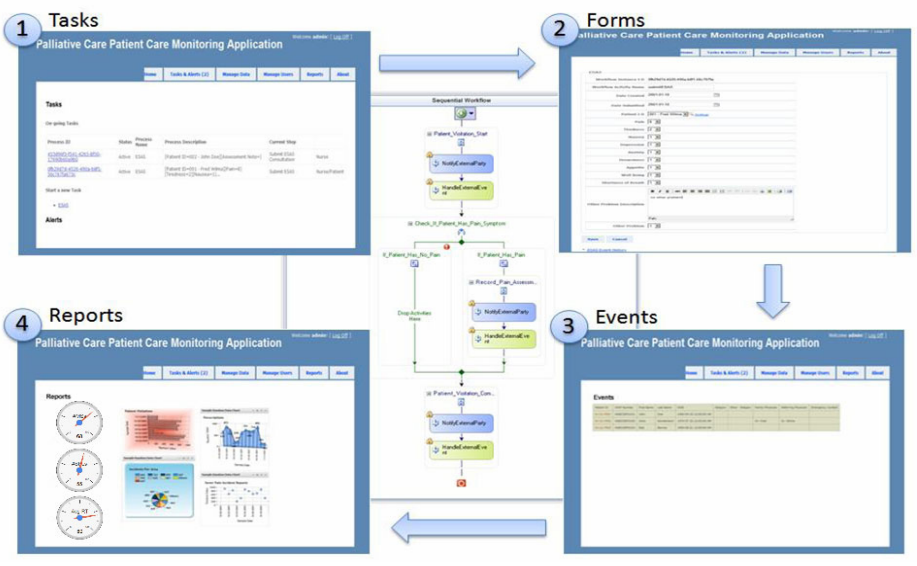


Fig. 2. Managed Process Application Framework

models of workflow, roles, events, entities, alerts and performance indicators. These models were simpler and easier to understand than the UML models used by AndroMDA, as they were specific to the requirements of managed process applications.

4 Managed Process Application Framework

Figure 3 shows our model-based managed process application framework architecture that was implemented in our prototype. The key requirement for implementing an application using MPAF is that there must be well-defined model definitions for workflows, roles, entities and events, alerts and indicators. The managed process application is defined by the execution of these model definitions by the Application Engine in the Application Service Layer which interacts with pre-defined services specific to managed process applications for security, presentation, workflow, persistence, and monitoring. It uses an Adaptive Object Model (AOM) [25] to hold a flexible representation of entities and events and transfer them between the Application Service Layer and the Data Access Layer. The Data Access Layer, supports both the model definitions (written in XML) as well as the persistence storage of the managed process application used for both forms and reports (entities, events and indicators), and the specific instance data for roles, workflow and alerts. The Presentation Layer has predefined user interfaces for tasks and alerts, but the specific forms and reports must be designed by developers to interface with the Application Service Layer.

This can be contrasted with the PAL-IS implementation which had a simplified three tier architecture. The presentation layer (and user experience) of PAL-IS was largely the same as our MPAF implementation. The PAL-IS Data Store was a single

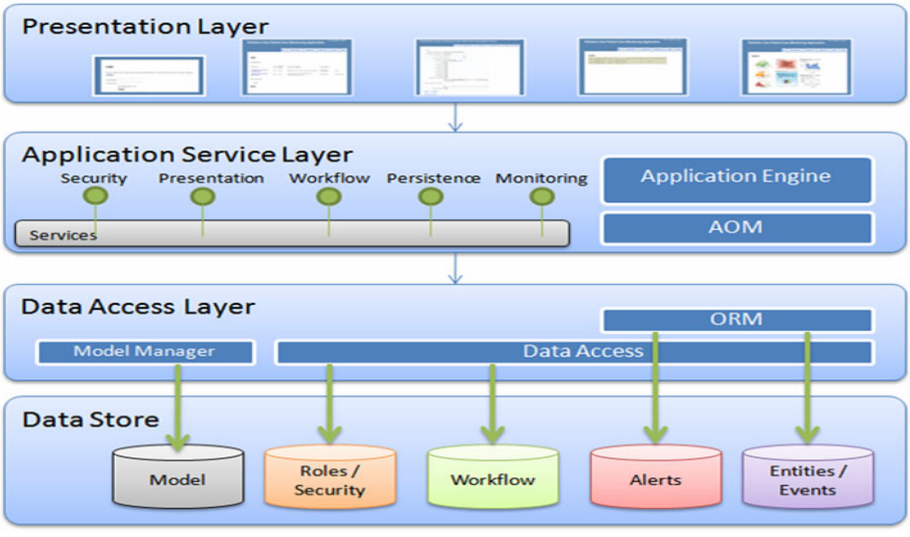


Fig. 3. MPAF Architecture

database schema that did not separate out business models, workflow and alerts. Most significantly, PAL-IS has a single hard coded application layer in ASP.NET instead of a separate data access and services layer in which an Application Engine leveraged an AOM and ORM to coordinated and manage processes based on models. PAL-IS seemed simpler to code initially, but the architecture ensured that the code was difficult to maintain and not easy to reuse.

It can also be contrasted with the AndroMDA implementation which had a very similar architecture. The same four tiers are present in AndroMDA. The difference is the data store was a single database schema that did not separate out business models, workflow and alerts. Instead of a generic Application Engine, a very specific application engine was generated and compiled. If the business models, workflow, or alerts changed, then the application needed to be regenerated. Any problems with the generated code needed to be fixed in the low level code itself. Usually a complete job of generating cannot be done and hand crafted code must be added, which compromises the ability to regenerated code. As a result the code was difficult to maintain and not easy to reuse.

4.1 Application Run-Time Processing

The **application engine** is part of the application run-time environment that manages the in memory AOM data for workflows, roles, entities, events and alerts. The application engine also coordinates calls between services including the presentation and persistence services. For example, when an event is submitted using the workflow service, the application engine makes sure that this event is persisted using the persistence service and sent for monitoring via the monitoring service.

The **workflow service** is responsible for creating a new workflow instance, submitting event data for a workflow activity, retrieving information about the list of

active workflow instances and retrieving information about a particular workflow instance. The workflow service is the interface for interacting with the workflow engine. The workflow engine manages the transition between workflow activities of managed processes.

The **monitoring service** is used for inspecting event data to create an alert for the event, if the event matches an alert trigger condition.

The **persistence service** uses event and entity model definition to perform persistence related actions. The persistence service uses an Object Relational Mapper (ORM), to persist and retrieve entities in a SQL relational database. At run-time, the persistence service, transforms event and entity model definitions into ORM mapping configurations and uses them to process the transformation of entity model and data into an SQL statement which will update, insert or retrieve information from the relational database.

The **presentation service** is responsible for generating html markup that will be used by the application engine to render web forms that users interact with.

The **security service**, among other things, is used for validating user credentials, issuing an authentication token, validating an authentication token and checking user's role membership.

An important element of the application run-time environment is the **dimensional database** for entities and events which provide integrated reporting for the managed process application as well as support for data entry forms. Events and entities are stored in separate tables that are organized using a "star" schema [10]. Event data can make a reference to an entity data but not vice versa. Both the event and entity tables are capable of handling new event and entity model definitions without the need to modify the event or entity table schema. The approach we have taken to handle this flexibility is to have separate tables for storing events and entities and design both the event and entity tables to have several fields for each possible data type that they can potentially handle. That is, several fields for each model data type (short text, medium text, long text, integer, and date time) are created on both event and entity table.

At run-time, the event and entity model definition are inspected by the **ORM** to map properties of the entity to corresponding fields of the event or entity table. The persistence service uses this mapping information to appropriately store and retrieve event and entity information from the dimensional database using the ORM.

4.2 Example Model Definitions

The types of model definitions executed by MPAF are illustrated here with some simple examples from our managed process application for severe pain management that was described in section 3.

Workflow

A workflow can contain and orchestrate different types of activities. Below is the normal routine workflow associated with a patient, or the Nurse who visits the patient, submitting an ESAS pain score, followed by a Nurse making an assessment on the state of the patient based on ESAS scores submitted. Note that additional processing required by doctors, for example to change increase pain medication is not defined in the routine workflow, but rather are defined by separate alert-triggered workflows.

```
<Workflow name="ESASWorkflow" description="ESAS">
   <CustomSequentialWorkflowActivity name="ESASWorkflow">
   <CollectEventDataActivity displayName="Submit ESAS" name="submitESAS"
                              eventTypeName="ESAS" roles="Nurse;Patient" />
   <CollectEventDataActivity displayName="Submit ESAS Consultation"
          name="submitESASConsultation" eventTypeName="ESAS_Consultation"
          roles="Nurse" />
   </CustomSequentialWorkflowActivity>
</Workflow>
```

Roles

The MPAF model definition can have a list of role names that are used in the application. These role names are used in other model elements to identify the roles that have access rights to perform an action that is represented by the particular model element. For example, in the workflow model definition, a particular workflow activity can have a list of roles that are allowed to execute the activity in the "roles" attribute of the workflow activity model definition.

```
   <Roles>
        <Role name="Patient"/>
        <Role name="CaseManager"/>
      <Role name="Nurse"/>
        <Role name="TeamAdministrator"/>
      </Roles>
```

Entities

Entities are constructs that are used to hold application data. They are used to describe business entities of an organization that exist and are involved in a business activity. An entity model definition can have one or more properties. Below is a model definition for a "Patient" entity as specified in the "name" XML attribute of the "EntityType" XML element. The "Patient" entity has property definitions such as "Patient ID", "First Name", "Last Name", date of birth ("DOB") and many others. The data type is also specified using the "type" attribute. Typical values for the "type" attribute in the model definition include, "System.DateTime" for date or time valued properties and "System.String" for text valued properties. The "Gender" property is a "pick list" ("isPickList="True""), it can only assume values that are described in the "pickList" attribute. That is the "Gender" property can only assume one of the following values "", "Male", "Female".

```
<EntityType name="Patient">
   <PropertyType name="DateCreated" type="System.DateTime"/>
   <PropertyType name="Patient ID" type="System.String" />
   <PropertyType name="OHIP Number" type="System.String" />
   <PropertyType name="First Name" type="System.String" />
   <PropertyType name="Last Name" type="System.String" />
   <PropertyType name="Gender" type="System.String" isPickList="True"
                                                  pickList=";Male;Female"/>
   <PropertyType name="DOB" type="System.DateTime" />
   <PropertyType name="Family Physician" type="System.String" />
   <PropertyType name="Referring Physician" type="System.String" />
   <PropertyType name="Emergency Contact" type="System.String" />
</EntityType>
```

Events

Events are used to record actual facts or measures in a managed process and are the point of interest for data collection during the execution of a managed process. Event data are only recorded in the context of performing a workflow activity within a managed process. The following is an event for patients submitting their ESAS score as a measurement of their pain.

```
<EventType name="ESAS">
   <PropertyType name="WorkflowInstanceID" type="System.String"/>
   <PropertyType name="WorkflowActivityName" type="System.String"/>
   <PropertyType name="DateCreated" type="System.DateTime"/>
   <PropertyType name="DateSubmitted" type="System.DateTime"/>
   PropertyType name="Patient ID" type="System.Int32" isLookup="True"
                                    value="System.User.[Patient ID]"/>
   <PropertyType name="Pain" type="System.Int32" isPickList="True"
                                    pickList="1;2;3;4;5;6;7;8;9;10" />
</EventType>
```

Events always contain property definition for “WorkflowInstanceID”, “WorkflowActivityName”, “DateCreated” and “DateSubmitted”. These properties are used to capture information about the id of the workflow instance that is used to capture the event data, the name of the workflow activity that is used to capture the event data, the date the event data is created and submitted. Event data can also make a reference to an entity data. In the above event model definition, the “Patient ID” property is used for making a reference to the “ID” of a “Patient” entity. The “Pain” property is also defined for this event with a possible set of values ranging from 1 to 10.

Alerts

An Alert model definition holds information about the name of the alert (“name” attribute), the event entity type name (“eventTypeName” attribute) that will be checked for an alert trigger condition and the alert trigger condition (“triggerCondition” attribute) for the event that is related to the alert. This alert model is used by the monitoring service of to check if events match an alert condition. The following is an example alert that is used to create severe pain alerts for “ESAS” assessments with a pain score level of eight or more on a scale of ten. Note that the alert trigger condition is described in terms of properties of the event that is related to the alert.

```
<AlertNotifications>
   <AlertNotification name="Severe Pain Alert" eventTypeName="ESAS"
           triggerCondition="ESAS.[Pain]>=8" triggerWorkflow="" roles="">
   </AlertNotification>
</AlertNotifications>
```

Performance Indicators

A performance indicator defines the name of the performance indicator (“name” attribute), the formula that will be used to calculate its value (“value” attribute), and the criteria that will be used to filter the set of events that are used to calculate the performance indicator value (“filter” attribute). This formula defined in the “value” attribute is described in terms of the properties of the events that are related to the performance indicator (“eventTypeNames” attribute) and the query capabilities of

the run-time environment's persistence service. For example, to calculate a performance indicator that measures the average duration between two event data submissions in hours, a formula similar to "avg(hour(Event_1.DateSubmitted) – hour(Event_2.DateSubmitted)) can be used. The "eventTypeNames" attribute is also used to identify the event types that are related to the performance indicator. Performance indicator model also defines color coded range of values for the performance indicator. The ranges that are color coded as "green" or "yellow" are assumed to be acceptable ranges. However, if the performance indicator value falls in the range of values that are coded as "orange" or "red" an alert notification will be created for the performance indicator. Performance indicators are checked every time an event is submitted to the monitoring service of the run-time environment.

```
<PerformanceIndicators>
   <PerformanceIndicator name="Daily Average Severe Pain Response In Hours"
   eventTypeNames="ESAS;                                          ESAS_Consultation"
   value="avg(hour(ESAS_Consultation.DateSubmitted) - hour(ESAS.DateSubmitted))"
   filter="ESAS.DateSubmitted between current_date() and current_date() + 1" >
   <InidcatorRange min="0" max="2" status="green"/>
   <InidcatorRange min="2" max="4" status="yellow"/>
   <InidcatorRange min="4" max="6" status="orange"/>
   <InidcatorRange min="6" max="" status="red"/>
   </PerformanceIndicator>
</PerformanceIndicators>.
```

5 Evaluation

We compare the three implementations of the managed process application for severe pain management in terms of models, run-time environment and engineering effort.

In MPAF, we have used business level models specific to the requirements of managed process applications whereas AndroMDA uses generic UML based models. The original PAL-IS system, on the other hand, has no business level models but has very limited database schema and implicit object model. As a result, all of the application source code in PAL-IS is manually crafted and is not generated or interpreted from a model. Using business level models lowers application development complexity and allows domain experts to easily get involved in the application development without the need for domain experts to understand the technical or code level details of the managed application framework.

Table 1. Models Comparison

Models			
Criteria	**PAL-IS**	**AndroMDA**	**MPAF**
Business Level Models workflow, roles, entities, events, alerts, indicators	No. Hand-crafted DB schema + implicit object model	Possible, mixed with code level elements (UML activity diagram + object model)	Yes

Table 2. Application Run-time Environment Comparison

Application Run-time Environment			
Criteria	**PAL-IS**	**AndroMDA**	**MPAF**
Business Process Workflow	No. Hard-coded workflow.	No. Hard-coded workflow generated based on activity diagram.	Yes
SOA	Not really. Hard coded services.	Generated from a Model. Hard-code invocation	Yes. Pre-configured.
Configuration	Possible, but ad hoc.	Possible, but ad hoc.	Yes
Alerting and Monitoring	Possible, but hard-coded.	Possible, but hard coded.	Yes. Pre-configured SOA driven by Event & Alert models.
Data Models & Reporting	Possible, hard-coded to hand-crafted database schema.	Possible, hard-coded to generated database schema.	Yes. Pre-configured event-based dimensional models.

Table 3. Engineering Effort Comparison

Engineering Effort			
Criteria	**PAL-IS**	**AndroMDA**	**MPAF**
Avoid recompile for business changes	No, Always need to code.	No. Always recompile, sometime code.	Yes, no code for workflow, roles, entities, events, alerts, indicators
Coding Complexity	High, manually coded	High, generate from models & manual coding of behavior	Low, configure engine with models
Model Abstraction level	C# Objects (Low) DB Schema (Low)	Full UML (Medium)	Business Elements (High)
Tool Support	Visual Studio	AndroMDA framework + UML Case tool (MagicDraw)	XML Editor to edit XML files.
Learning Curve	Programming Language	Programming Language, UML AndroMDA Framework	XML Business Model for workflow, roles, entities, events, alerts, indicators
Code reuse for similar applications	Low. Ad hoc	Medium. Model-structured code	High Engine based. Reuse Models.

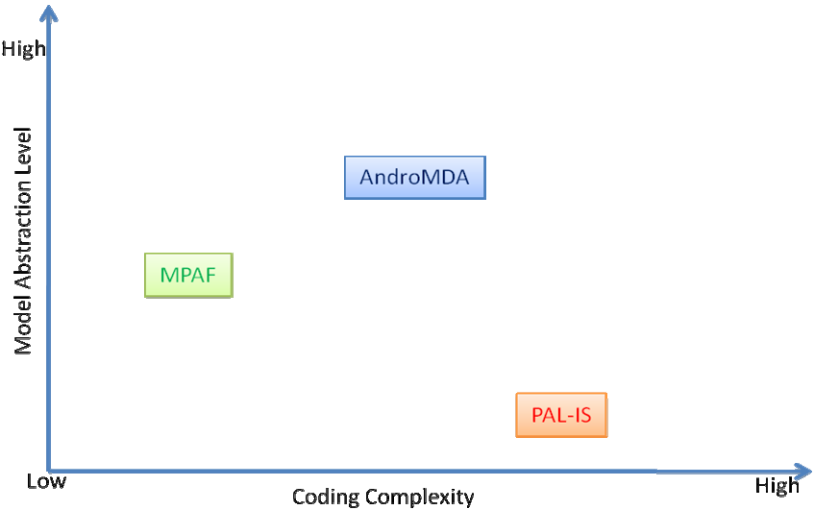


Fig. 4. Complexity comparison

PAL-IS is a totally customized run-time environment, whereas AndroMDA and MPAF have pre-configured environments based on their model-driven approach. However, a pre-configured environment is generic and requires extensive coding to support a specific application. MPAF, on the other hand, is pre-configured to execute managed process applications defined by models. PAL-IS and AndroMDA allow system level configuration while MPAF allows business/domain level configuration.

The most critical element of this model-based configurability is that MPAF provides integrated data management that links the execution of a business process in the run-time environment to the built-in mechanisms for data collection, alerting and performance indicator reporting. These three elements are perhaps the essential aspects of a managed process application.

The essential difference between MPAF and the other two implementations is that the abstracted architecture and model definitions are optimized and specialized for a particular class of application: managed process applications. As a result both the model complexity and the development complexity are simplified which makes the learning curve, code reuse and code maintenance easier. Tool support is an issue because it was not a focus of our research. We simply used text editors to edit model definitions. One area for future work would be the development of better user interfaces for building models.

As shown in figure 4, both the modeling and code complexity of MPAF is less than that of AndroMDA. The original PAL-IS has a very low model complexity because it does not use models at all but the development complexity for PAL-IS is high. MPAF has a medium level of model complexity as there are only a few model elements (workflow, roles, entities, events, alerts, performance indicators) that need to be configured. MPAF models are designed and targeted for managed process applications and this gives it a high model specificity value.

6 Conclusions

We have illustrated the potential utility of specializing application frameworks for particular classes of applications, as well as the potential advantages of model-based approaches that interpret or execute models, in contrast to model-driven approaches which generate code. A more systematic and complete set of case studies which evaluates the approach in practice in industry is needed to fully validate and quantify the potential benefits.

In theory, our framework should be applicable to any type of managed process application that has the requirement to enable incremental process improvements to easily innovating new processes and support constantly changing data and process requirements. This would include applications in other areas such as insurance and government, as well as other applications in healthcare. However, at this time we have a prototyped framework that was used to build a proof of concept managed process application for a palliative care severe pain patient monitoring scenario. More comprehensive case studies should be performed to further validate our approach and to identify an exhaustive list of criteria that can be used for evaluation. Using our framework in other domains may also indicate additional requirements and components that our framework should address.

The model definition that we developed in our scenario is relatively simple in that it can be edited by a simple text editor. However, the model for real scenarios will be much larger in scope and will require a visual modeling tool.

Acknowledgements

This work was supported by a Collaborative Health Research Project grant from CIHR and NSERC (Canada) on Performance Management at the Point of Care: Secure Data Delivery to Drive Clinical Decision Making Processes for Hospital Quality Control.

References

1. AndroMDA, `http://www.andromda.org` (retrieved November 2010)
2. Bass, L., Clements, P., Kazman, R.: Software Architecture in Practice, 2nd edn. Addison-Wesley, Reading (2003)
3. Eze, B., Kuziemsky, C., Peyton, L., Middleton, G., Moutthам, A.: Policy-based Data Integration for e-Health Monitoring Processes in a B2B Environment: Experiences from Canada. Journal of Theoretical and Applied Electronic Commerce Research, 56–70 (2010)
4. Forrester Report: Enabling Dynamic Business Processes with BPM and SOA. Forrester (2008)
5. Fowler, M.: Patterns of Enterprise Application Architecture. Addison-Wesley, Reading (2003)
6. Hailpern, B., Tarr, P.: Model-driven development: the good, the bad, and the ugly. IBM Systems Journal, 451–461 (2006)
7. Huhns, M.N., Singh, M.P.: Service-Oriented Computing: Key Concepts and Principles. IEEE Internet Computing 9(1), 75–81 (2005)

8. Jeng, J., Schiefer, J., Chang, H.: An Agent-based Architecture for Analyzing Business Processes of Real-Time Enterprises. In: Seventh International Enterprise Distributed Object Computing Conference (EDOC 2003), Brisbane, Australia (2003)
9. Jurič, M.B., Mathew, B., Sarang, P.: Business Process Execution Language for Web Services (2006)
10. Kimball, R.: The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling, 2nd edn. John Wiley & Sons, Chichester (2002)
11. Leymann, F., Roller, D., Schmidt, M.-T.: Web services and business process management. IBM Systems Journal 41(2), 198–211 (2002)
12. Liu, X., Peyton, L., Kuziemsky, C.: A Requirement Engineering Framework for Electronic Data Sharing of Health Care Data Between Organizations. In: Babin, G., Kropf, P., Weiss, M. (eds.) E-Technologies: Innovation in an Open World. Lecture Notes in Business Information Processing, vol. 26, pp. 279–289. Springer, Heidelberg (2009)
13. Michelson, B.M.: Event-Driven Architecture Overview Event-Driven SOA Is Just Part of the EDA Story, from Object Management Group, http://www.omg.org/soa/Uploaded%20Docs/EDA/bda2-2-06cc.pdf (retrieved November 2010)
14. Microsoft Application Architecture Guide 2nd Edition (Patterns & Practices), Microsoft Corporation (2009)
15. Middleton, G., Peyton, L., Kuziemsky, C., Eze, B.: A Framework for Continuous Compliance Monitoring of eHealth Processes. World Congress on Privacy, Security, Trust and Management of eBusiness (2009)
16. OASIS BPEL, from OASIS, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel (retrieved November 2010)
17. OMG Model Driven Architecture, from OMG - Object Management Group, http://www.omg.org/mda/ (retrieved November 2010)
18. Palpanas, T., Chowdhary, P., Mihaila, G., Pinel, F.: Integrated model-driven dashboard development. Information Systems Frontiers 9(2-3), 195–208 (2007)
19. Saraiva, J.S., Silva, A.R.: A Reference Model for the Analysis and Comparison of MDE Approaches for Web-Application Development. J. Software Engineering & Applications 3, 419–425 (2010)
20. Schmidt, D.C.: Model Driven Engineering. IEEE Computer (2006)
21. Smith, H., Peter, F.: Business Process Management: The Third Wave. Meghan-Kiffer Press, Tampa (2003)
22. Truyen, F.: The Fast Guide to Model Driven Architecture - The Basics of Model Driven Architecture, from OMG-Object Management Group, http://www.omg.org/mda/mda_files/Cephas_MDA_Fast_Guide.pdf (retrieved November 2010)
23. van der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, M.: Business process management: A survey. In: van der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, M. (eds.) BPM 2003. LNCS, vol. 2678, pp. 1–12. Springer, Heidelberg (2003)
24. Weske, M.: Business Process Management: Concepts, Languages, Architectures. Springer, Heidelberg (2007)
25. Yoder, J., Johnson, R.: The Adaptive Object-Model Architectural Style. In: Proceeding of the Working IEEE/IFIP Conference on Software Architecture (2002)

Harnessing Enterprise 2.0 Technologies: The Midnight Projects

Lee Schlenker

EMLYON Business School
Lyon, France

Abstract. The following contribution explores the relevance of the vision of Enterprise 2.0 in the light of the past efforts and future plans of a technology company's pre-sales team in Europe, the Mideast and Asia. The discussion begins with a review of varying conceptions of Enterprise 2.0 to identify the potential impact, value levers, and challenges of these applications in sales and marketing. This conceptual framework is then applied to explore how a major technology supplier is executing 2.0 strategies in their own organization. The case study describes the business challenges the company that has justified the focus on Enterprise 2.0, the individual skills that facilitated their initial progress in this area, and the roadmap and metrics used to guide and evaluate these experiments. A short discussion section will close the case by exploring the value proposition for the organization and the industry as a whole.

Keywords: Enterprise 2.0, collaborative technologies, social media, sales and marketing processes.

1 Enterprise 2.0

"Enterprise 2.0" evokes a vision of using the web as a platform for enterprise applications to facilitate the production and aggregation of user-generated content. Though practitioners have increasingly adopted Web 2.0 technologies as part of their on-line activities, there is notable debate in the research community over the specificity of "2.0".[1] What does this vision mean to marketing and sales consultants and how do they integrate 2.0 strategies into the way they work? Are such investments delivering on the promise of providing a new source of creativity, influence and empowerment?[1]

Enterprise 2.0 generally refers to changes in the way that software is designed, distributed and consumed over the Internet rather than a set of distinct technical specifications. In introducing the term Enterprise 2.0, Andrew McAfee (2006) argued that such applications should include technical functionalities for search, links, authorship, tags, extensions and signals.[2] Constantinides and Fountain (2007) proposed that 2.0 technologies refer to software used on-line, whereas social media focuses on the organizational impact of the use of these applications.[3] Enterprise 2.0 technologies,

[1] According to the Social Network Practitioner Consensus Survey of May 2007 more than 50 per cent of professionals participate already in social networks.

G. Babin, K. Stanoevska-Slabeva, P. Kropf (Eds.): MCETECH 2011, LNBIP 78, pp. 189–198, 2011.

including blogs, tagging, social networks, online forums and mashups, can be viewed as a corporate implementation of web-based technologies to extend organizational capabilities for collaboration, discovery and integration.

Mayfield (2007), among others, has suggested that these applications can potentially spawn participation, conversation, and connectedness inside organizations and in their interactions with their business communities.[4] Hinchcliffe (2007) has since insisted that such "social "technologies are better understood in exploring their organizational goals: promoting transparency, diversifying content, harnessing the wisdom of the crowds, and facilitating appropriation by management and employees alike.[5] In subscribing to this vision in which value propositions are based on network effects, IT suppliers and distributors alike have been challenged to harness these technologies for their own organizations, for their business partners, and for their external customers.

If the potential impact of Enterprise 2.0 goes beyond accessing a collection of on-line applications, we need to understand how these technologies influence managerial and employee behavior. The popular press has concluded that the Web 2.0 does indeed transform individual and group performance by impacting both the way people communicate and the power structures between vendors and consumers.[6] In the IT industry, the value proposition has focused on challenges facing strategists and marketers in managing customer behavioral change.[7] Although marketers are increasingly engaging Enterprise 2.0 campaigns as part of their marketing platforms, the question of how these applications actually shape managerial behavior remains an open question.[8] Because the technical and business perspectives of Enterprise 2.0 are intimately interrelated, the identification of the causal factors influencing organizational productivity is difficult to gauge.

From a business point of view, the potential value propositions of Enterprise 2.0 are built around aggregating management and employee experiences, building trust, networking inside the organization and out, and co-creating value directly with the customers. On one level, these technologies can be used to actively "listen" to the customer's voice. Sales and marketing managers can potentially use these approaches to develop personalized one-to-one marketing campaigns. On another level, these technologies promote distributed information systems and in doing provide a common forum for discussion within organizations. Ideally, the result would be one of deeper personal engagements to the organization, its products and services. The bottom line is a vision of Enterprise 2.0 which enhances collaboration and cooperation as a sustainable source of competitive advantage.

This vision is none-the-less tainted by a number of questions that challenge long established organizational practices of command and control. In relying on user generated content, Enterprise 2.0 technologies go beyond the controllable on-line presence of the corporate Web site to on-line experiences that lie largely outside managerial control. Influenced by blogs, chats, and instant messaging, individual preferences and decisions are increasingly based on inputs provided by sources beyond management's chain of command. The wealth of information and opinion at our fingertips has "complexified" the decision-making process, management has discovered that influencing employee and/or customer behavior by traditional communication platforms appears more challenging than in the past.

How can organizations best harness enterprise technologies to enhance efforts in sales and marketing? If such technologies are intended to impact business practice, how are they best implemented in an organizational setting? Which incentives will

actually push individual managers to privilege on participation, conversation, community and connectivity? If the value of Enterprise 2.0 technologies depends on their actual use in the organization, what metrics best measure their eventual success? Perhaps a good place to start would be in exploring how one of the leaders of the IT industry is attempting to put theory to practice.

2 The Context of the Midnight Projects

Oracle's Enterprise 2.0 strategy, as outlined by the company's president Charles Phillips in early 2008, comprises fusing 2.0 capabilities throughout its product offering, delivering Enterprise 2.0-enable Oracle applications, and promoting its WebCenter portal platform.[9] To support this vision, the company has embarked on an ambitious acquisition of 2.0 technologies including those of Sun Microsystems, Golden Gate Software, HyperRoll, Silver Creek, AmberPoint and Convergin.[2] These investments haven't been accompanied to date by suggested guidelines for the internal use of these technologies, nor explicit propositions of how 2.0 strategies should impact its own production, sales, and service processes. In working to fill these gaps, Alfonso Di'Ianni, Senior Vice President, European Enlargement and Commonwealth of Independent Stages Region, initiated a number of workshops and discussions in early 2010 to explore the potential value propositions.

Oracles pre-sales regional presales team didn't wait for managerial direction in elaborating their personal projects using 2.0 platforms. Several members of the team are today recognized as 2.0 specialists in their own countries, one member is the author of a leading reference book on the subject.[10] Referred to as "the Midnight Projects", many members of the team have produced personal blogs, while several of the subs are now producing and distributing video clips. Alain Ozan, Vice President for the Technology business in the region and head of the pre-sales team, proposed that the midnight projects constitute one of the three experiments to promote common practices and encourage capitalization of existing efforts.

Three specific goals established for the Midnight Projects included:

- Enhancing the visibility of these projects both internally to upper management and externally to the company's customer base;
- Enriching the impact of the experiments on the personal brand of the presales consultants:
- Strengthening the channels for information exchange, reuse and evaluation among the team as a whole.

2.1 Developing the Skills for Successful Enterprise 2.0 Strategies

An initial discussion with the project team failed to establish exactly what blend and level of skills were critical to successful Enterprise 2.0 strategies. Were the project

[2] Founded in 1977 by Larry Ellison, Oracle is a multinational computer technology corporation that develops and markets enterprise software systems. The organization employs today more than 115,000 people worldwide in more than 145 countries around the globe. In terms of revenue, Oracle is the third-largest player in the software market behind Microsoft and IBM.

managers simply trying to showcase existing skills or to encourage the consultants to develop complementary skills through "Enterprise 2.0"? Was the project's intention to focus on the skills of the managers themselves or those of their customers in working with Oracle? Should the focus of skill development be on traditional business competencies or to the contrary, should the experiment encourage the consultants to focus on a different skill set in growing their business?

Fig. 1. Cloud representation of the pre-sales team responses to the question "What type of skills do you believe Enterprise 2.0 technologies should develop?"

The pre-sales consultants themselves tended to underline the development of business critical skills that defined the substance of their current assignments. Knowledge about Oracle products and methodologies was seen critical to developing communication platforms. Many consultants suggested that demonstrating both knowledge and skills in business critical areas, such as security and business intelligence, were powerful levers in developing traction and credibility with the target audience. Other managers felt that more room should be given to the softer skills: presentation skills, strategic thinking, and knowledge management in developing the value of their projects. From their point of view, Enterprise 2.0 was just another technology platform and as such did not a fundamental change in the way they did business.

If Enterprise 2.0 was just a question of developing business critical skills, given the abundance of skills already in place, the team could have concluded that there was little need for the project at all. The concept of Enterprise 2.0, none-the-less, suggested a different approach to using technology to connect with customers. Whereas in traditional marketing strategies information technology has been deployed as a one way communications platform, Enterprise 2.0 gurus suggest that information technology's role now is to design a space to facilitate conversation between customers, business partners and the organization. Historically, corporate communication is grounded in corporate messaging, whereas Enterprise 2.0 is based on encouraging user-generated content. In traditional marketing campaigns, companies work to "control" their brand, in an Enterprise 2.0 mindset the consumer that defines the nature and the extent conversation. For sales driven organizations, the repeatability of the message is the overarching concern; in 2.0 personalizing customer stories is the over-riding imperative.

Beyond considerations tied to the platform itself, Alain Ozan suggested that the project was exploring a skill set that went beyond that of the traditional presales consultants. Given the voluntary nature of the Midnight projects, he did not wish to impose any one solution or best practice on the group as a whole. He suggested to his team that Enterprise 2.0 might be best understood as an approach to infrastructure rather than mastering any one particular technology. In light of the current customer challenges and the relative novelty of the 2.0 methodologies, he felt that crowd sourcing, rather than relying on industry experts provided a much more realistic approach for moving forward. Skills associated with benchmarking and reuse took on a new meaning here for probing, filtering and experimentation were to be actively encouraged.

In sum, trying to understand what skills were targeted in the Midnight Projects will have a strong bearing on how the team would evaluate the project's success. Several discussion questions came to mind:

- Are the presales consultants traditional skills set a hindrance or an advantage in building a 2.0 strategy?
- Which applications are best suited for two way conversations?
- To what extent can crowd sourcing provide a lever for initiating the conversation?
- To what extent should the project metrics measure skill development rather than the technology itself?

2.2 Building a Roadmap to Link Enterprise 2.0 Technologies to Strategy

The project roadmap provides the link between organizational strategies, project tactics and technological supports. A viable roadmap should outline the current state of affairs, middle and long objectives, and the steps needed to move the project forward. For the Midnight Projects, a team workshop was held in June 2010 to focus on fostering a viable 2.0 strategy for each participant, as well as the team as a whole.[3] To meet this goal, the workshop addressed in turn the current personal projects, explored the business objectives, and deployed a form of crowd sourcing to identify the priorities for the months to come.

Members of the presales team have developed various platforms of social media for a variety of objectives over the past couple of years. In most cases the technological platforms used to deploy these social technologies were implemented outside the corporate firewall to avoid technical and organization concerns of the IT department. Several have created personal blogs on subjects ranging from security to business intelligence in which they privilege the localization of corporate messaging and their own expertise.[4] Video casts have proven quite popular for other members as they attempt to reuse content put together for sales calls, conferences, and workshops.[5] Video clips, including Oracle Austria YouTube channel [11], have been designed to

[3] Referred internally to as the Athens Workshop, the conference reunited the top performers from Oracle's pre-sales team in EMEA to review the past experience and future direction of the group's Enterprise 2.0 projects.

[4] One notable example among others from Andrey Pivovarov, `http://oraclebi.ru`

[5] See for example Lajos Sárecz' videocast, `http://tv.computerworld.hu/video/az-adattarolas-alapjai`

create interest and a buzz around Oracle products, services and concepts In the Athens workshop, examples of each medium were discussed among the team using Sollis' Conversation Prism [12] to explore the panorama of social media and the different forms of customer conversations.

All of the projects identified for development within the framework of the Midnight projects met the minimal definition set out by McAfee (2006) with the acronym "Slates": each incorporated the functions of search, links, authorship, tags, extensions and signals. As a whole the projects also met some or most of the criteria set out by Mayfield (2007) and Hinchcliffe (2007) for social media: they promoting transparency within the sales team, they helped diversifying and localize content, they sought to incorporate the wisdom of the crowds, and they were by definition appropriated to various degrees by management and employees alike. Going beyond the basic definitions, the Midnight projects set out to more ambitious goals: tying the each project to longer term business objectives, extending the impact of each initiative to amplify the reach and visibility of the consultants' efforts, and identifying appropriate metrics to feedback to the team and the organization the corresponding return on investment.

The link between the use of Enterprise 2.0 technologies and strategy was evaluated using an adaptation of Tom Peter's propositions on personal branding.[13] In the context of the Midnight Projects, the consultants were asked to explore their own beliefs concerning their personal strengths, why their customers' trust tem, and how their use of Enterprise 2.0 technologies could make a difference. The notion of Social Impact was introduced and examined using the application Webmii to explore where the team's social footprints could be found on the Web and which business concepts were commonly associated with each consultant.[6] The notion of Social Reach was also presented and explored using examples from the application HowSociable to map out how each consultant's efforts were taken up on different types of social media, including Facebook, Twitter, and the blogging community.[7] Following these discussions, the consultants where challenged as a group to analyze how they could adapt or extend these types of applications to get a more valid image of the strength of their personal brands.

An integral part of the Athens Workshop was the idea that there is no right solution, nor even "one best way", in implementing Enterprise 2.0 applications. In line with Snowdon and Boone's approach (2007) to complex problem solving, group discussion and analysis was integrated into the workshop to encourage the participants to probe, sense, and respond.[14] The first of three workshop deliverables involved probing the differing objectives of the consultants, and then asking them by group to rank the individual and group importance of each objective as well as which forms of Enterprise 2.0 technologies might be used to work towards that objective. A second deliverable involved a roundtable discussion of the strengths and weaknesses of the different types of applications available, as well as which could be included in a group toolbox. Finally, the third deliverable explored a potential implementation plan, and included mapping the customer's value chain, the communications patterns,

[6] Webmii is an online people search engine that provides aggregate scores of social presence on the Web, `http//webmii.com`

[7] HowSociable is an online application designed to measure brand visibility on the social web, `http//howsociable.com`

and what touch points each 2.0 application would cover. Taken as a whole, the resulting road map was designed to access the relative maturity of the consultants' approaches to 2.0, their personal objectives, and collectively which next steps could help each move their projects forward.

The future success of the Midnight Projects will depend as much on the coherence of the consultant's personal brands as it does with the relevance of their technological supports. Several discussion questions came to mind:

- Should personal brands be modeled after "best practices" or personal strengths and convictions?
- How important are the notions of Social Reach and Social Impact in understanding the impact of personal and team branding in a community or market?
- To what extent do you and your team understand the variety of customer conversations and potential technological supports?
- To what extent should the project metrics measure the coherence of vision rather than the technology itself?

2.3 Establishing the Metrics to Judge Success

The notion of developing evaluation metrics for the Midnight projects, introduced during the Athens workshop, provoked lively debate both during the conference and in the weeks that followed. Key points of contention included the degree to which metrics are critical to developing the long term success of Enterprise 2.0, whether it is possible to capture the quality of customer conversations, and whether the introduction of metrics might lead the project participants to focus more on the numbers than on developing customer intimacy. Let's quickly look at the arguments for and against using metrics in the project today.

Although hard metrics are part of the DNA in any sales driven organization, several factors influenced perceptions of their importance for these initiatives. On one side, the team was in general agreement that the metrics should measure the improvement in customer relationships rather than be tied to the technology itself. Some consultants argued that the best way to improve Midnight Projects was to define benchmarks that could be used to compare the initiatives. Other argued that as the Midnight Projects gained visibility within the organization, they would be evaluated by management with or without specific metrics. They insisted that the team should take the initiative in proposing operational evaluation metrics before their managers did it for them.

On the other side, many of the participants feared that setting hard metrics might well stifle the creativity, passion and commitment that had initially driven the initiatives. Several argued that there were no hard, direct measures of individual or team effectiveness in Enterprise 2.0, and that any future constructs would be at best poor approximations of the perceived value of the project. Others feared that once the metrics were established, the consultants would focus more on the evaluation criteria rather than on improving the working relationships with their customers. Finally, some participants offered that the metrics would be of marginal value at best and needed to be considered in relation to whether they would facilitate or restrict the consultants' scope of action.

The workshop facilitators suggested that in any case the group's work would be evaluated by either explicit or implicit metrics. In driving their projects forward, the project sponsors suggested that the group focus on which metrics might encourage the passion and creativity that provided the initial focus for the projects. They also asked the consultants to consider that their projects were essentially forms of virtual communication with management, business partners and customers where signs of referred and/or situation trust are particularly important. Metrics might not be "game stoppers", but the lack of metrics might well hinder the possibility of the consultants communicating the value of the stories they had to tell.

After discussion, the team identified a number of potential metrics for the Midnight projects. These included the number of items posted or received for discussion, the number of prospect or customer links to posted items, the number of unique users (audience), the take-up by mainstream media coverage, the number of leads qualified for the pre-sales team. How these metrics can be specified, as well as how they can be captured automatically, is still under discussion.

As the question of metrics remains open within the Midnight Projects, a number of further discussion questions come to mind:

- Will the implementation of metrics facilitate or restrict the development of Enterprise 2.0 projects?
- Which metrics can best measure the quality of customer relationships?
- At what point in time should metrics be suggested or imposed in a project of this nature?
- Are customer testimony and other "soft" metrics sufficient to obtain sponsor support?

3 The Impact of Enterprise 2.0 on Business Practice

Enterprise 2.0 technologies have been presented here as corporate implementations of web-based technologies to extend organizational capabilities for collaboration, discovery and integration. This discussion began by reviewing varying conceptions of Enterprise 2.0 to gauge the potential impact, value levers, and challenges of these applications in sales and marketing. This reference points provided a framework of analysis to explore how a major technology supplier promotes Enterprise 2.0 internally to influence the presales team's activities in Europe, the Mideast and Asia. In guise of a conclusion, let's quickly review what we have learned about these technologies influence on business practice.

What does the vision of Enterprise 2.0 mean to marketing and sales consultants and how do they integrate 2.0 strategies into the way they work? Enterprise 2.0 technologies suggest a user-focused approach to social technologies within the organization rather than define a specific set of technical specifications. The corresponding project mindset did not lend itself well to classical corporate approaches to IT implementations. In a process centric sales culture similar to Oracle's, the origins and the motivations for innovative approaches to social technologies come from the consultants themselves rather than from upper management or the IT department. The Midnight projects were bred outside formal processes by the consultants themselves, and then nurtured through internal discussion and benchmarking. Such discussion and experimentation appears critical in processes in which there are no best practices.

Are such investments delivering on the promise of providing a new source of creativity, influence and empowerment? The objectives of 2.0 technologies extend far beyond considerations with platform and applications; they are implemented to influence managerial behavior. The IT department's desire to promote corporate mandated software proved counterproductive in this case, discouraging individual initiatives and limiting the available options. Beyond the early innovators that invested in the Midnight projects out of personal conviction, the other consultants wished to clearly understand the organizational and personal value propositions of 2.0 technologies before committing to their development. These experiments do not appear, in their opinion, to be directly tied to the identifiable organizational objectives for either sales or logistics. The active demonstration of the intrinsic value of Enterprise 2.0 use scenarios doesn't seem convincing either, for their use seems to compete with better established metrics of lowering cost and time recognized throughout the organization. Better visibility of how they directly improve individual or organizational performance is needed here.

Finally, at least in this case, the question of how to best evaluate the value proposition of Enterprise 2.O technologies remains an open question. If the significance of such "social" technologies is in their impact on the organization, the jury is still out on which metrics might best measure their eventual success. On one hand, since this experiment invites the organization to look beyond the narrow definitions of operational efficiency, the potential range of pertinent metrics is almost endless. On the other hand, the consultants argued with conviction that the choice of metrics would both structure and limit the potential outcomes. In both cases, given the personal investments needed to put these projects in place, careful consideration of what constitutes the personal return on investment seems critical to the long term success of such "Midnight" projects.

Although this one case can hardly be considered representative of the all markets and industries, there are several reasons to believe that these initial conclusions can help provide focus for future research and debate. Since the principal actors in the IT industry play a fundamental role in the design and deployment of Enterprise 2.0, the experience of its sales and marketing teams will have a large impact on other organizations. The experience in Oracle EMEA underlines that this vision is not dependent on technology alone, but conditioned by the context and experience of client organizations. This project also confirms the suggestion that the promise of Enterprise 2.0 tests the precepts of managerial models based on command and control. Finally, the case seems to underline the importance of accounting for objectives, incentives, and return on investment in gauging successful implementations.

References

1. Gillin, P.: The New Influencers. In: A Marketer's Guide to the New Social Media. Quill Driver Books Word Dancer Press, Inc., CA, USA (2007)
2. McAfee, A.: Enterprise 2.0: The Dawn of Emergent Collaboration. MIT Sloan Management Review 47(3) (Spring 2006)
3. Constantinides, E., Fountain, S.: Web 2.0: Conceptual foundations and marketing issues. Journal of Direct, Data and Digital Marketing Practice 9, 231–244 (2008)
4. Mayfield, A.: What is social media? Spannerworks 1.4 (2007)

5. Hinchcliffe, D.: The state of Enterprise 2.0. ZDNET.com, London (October 22, 2007)
6. Markillie, P.: A survey of consumer power. Crowned at last, The Economist (April 2, 2005), http://www.economist.com/node/3785166?story_id=3785166
7. McKinsey: How business are using Web 2.0: A McKinsey global survey. The McKinsey Quarterly (2007), http://www.mckinseyquarterly.com/Marketing
8. Forrester: The ROI Of Social Media Marketing (2010), http://blogs.forrester.com/augie_ray/10-07-19-roi_social_media_marketing_more_dollars_and_cents
9. Phllips, C.: OAUG Collaborate 2008 (2008), reported on http://www.ondemandbeat.com/2008/04/16/charles-phillips-outlines-oracles-enterprise-20-strategy/
10. Weckerle, P., et al.: Reshaping Your Business with Web 2. McGraw-Hill, New York (2009)
11. Oracle Austria, http://www.youtube.com/oracleaustria
12. Sollis, B.: The Conversation Prism, http://www.briansolis.com/2009/03/conversation-prism-v20/
13. Peters, T.: The Brand Called You, FastCompany, http://www.fastcompany.com/magazine/10/brandyou.html
14. Snowden, D.J., Boone, M.: A Leader's Framework for Decision Making, Harvard Business Review, 69–76 (November 2007)

Following the Conversation: A More Meaningful Expression of Engagement

Cate Huston, Michael Weiss, and Morad Benyoucef

University of Ottawa, Carleton University
chust056@site.uottawa.ca, weiss@sce.carleton.ca,
benyoucef@telfer.uottawa.ca

Abstract. Twitter is a relatively recent phenomenon, and the common metric of success is number of followers. Because people use Twitter in a myriad different ways, and the presence of spammers, it is necessary to discover new ways of quantifying success. In this paper, we explore the nature of engagement on Twitter and find the traditional follower/following network to be meaningless in this regard. Building on previous research, we define engagement in terms of interactions using the @ notation, and visualize this as a graph. We then apply clique finding techniques to this graph, to extract a sub-graph of the most important connections in a user's immediate network.

Keywords: Twitter, influence, conversation networks, cliques, visualization, micro-blogging.

1 Introduction

Micro-blogging is a simple but versatile concept that involves sending and receiving short messages; one of the most popular micro-blogging services is Twitter. This may be because "[c]ompared to regular blogging, micro-blogging fulfills a need for a faster and more immediate mode of communication" [1]. The value of status messages is not technical; it is created through user participation - users participating both as authors and as readers created an active community, and is the reason for widespread adoption [2].

Popular social networks such as Facebook, and MySpace support the concept of "friends" – reciprocated relationships. On Facebook, not only are relationships reciprocal, but for 65% of users all interactions are reciprocated [3]. A 2008 study found users had an average of 124 friends [4], as of February 2010 this was 130 [5]. By contrast, Twitter allows one-way relationships in the form of "followers". A minority of usersprotect(private to only those authorized) their updates,for most anyone can subscribe to their stream either by "following" or by subscribing to the RSS feed [6]. This option of one-way relationships is also true on Flickr, where around (68%) of links are reciprocal [7]. We do not have the data to compare this to Twitter.

The @username notation allows for conversation between two users (who do not need to follow each other), and stems from older IRC practice [8]. The exchange shows up only in the feeds of those people who follow both the person sending the message and the person it is directed to, making it non-intrusive. The communication is available to anyone looking at the sender's Twitter page, or via RSS.

G. Babin, K. Stanoevska-Slabeva, P. Kropf (Eds.): MCETECH 2011, LNBIP 78, pp. 199–210, 2011.

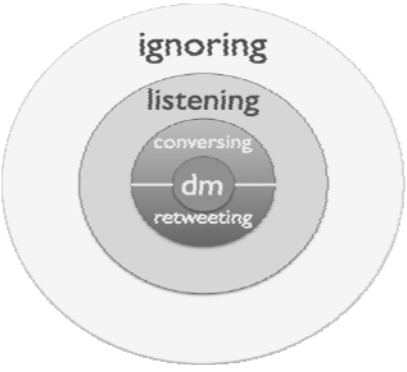


Fig. 1. Levels of Interaction on Twitter

Thus there are various levels of interaction. The most private (intimate) is the direct message, which we cannot track through the API without authentication (and then only for an individual user). We have engagement through conversation and retweets (reposting other people's tweets), passive listening (also known as lurking) and ignoring. These are expressed in Figure 1. Missing from the diagram is the possibility of something between passive listening and lurking, i.e. quickly skimming looking out for some specific things that are of interest. It is important to consider that a user might participate at all levels with another user [9]. Only spammers, interested only in pushing their content, will remain always at the outside – ignoring, not consuming.

Lurking has been described as a "tragedy of the commons", suggesting that lurkers do not benefit the overall ecosystem. This is inaccurate – content is no less because someone chooses to passively consume and not contribute. Lurking may be an important initial step as a new user tries to understand the service [9]. The majority of users in an online space has always been lurkers; given the volume of tweets in our stream in any given day, we are almost certainly all lurking with respect to at least some of them. Lurking is really more akin to listening – we all move between the states of listening and disclosing online, and both are forms of participation [9].

Can we measure this group of passive consumers? A rough estimate can be produced using data from link shorteners (such as [10]) (18.96% of regular tweets contained a link, and (56.69%) ReTweets contain links [11]). One popular link from the first author had 51 tweets (tracked by [12]) and 441 clicks (tracked via [10]) - this does not measure people who read the post were not sufficiently interested to click on it. We can infer that the number of passive consumers is potentially very large.

In this paper we argue that attempting to characterize the interaction of users in terms of a directed network graph of follower-following relationships has serious limitations. We propose a new way to characterize user engagement. We believe that this is important as judging a user's influence on the basis of their number of followers is prevalent, and yet misleading (e.g. spammers with over 1000 followers). To this end, we have created a tool that shows a user's network and allows them to identify close outer connections they may be interested in interacting with. Our preliminary finding is that there is significant interest in this approach (over 1000 hits over a weekend when the initial network graphs were released on the first author's website,

and over 45 requests for the latest release) and that these users are finding utility in the visualization of their network. Further, we are finding that the number of cliques a user is a part of varies greatly, and that this is likely a result of their varying use of Twitter.

2 Related Work

Number of followers is often used as the metric for influence on Twitter. In a proposed "dynamical theory of opinion formation", people with a large number of connections have huge influence [13]. Similarly, the "social choice model for evolutionary dynamics of behavior in social networks" also models the contagion of ideas [14]. The question we ask is, what constitutes a connection? Potentially, high levels of interaction with a diffuse group of others who have high levels of interaction will result in being more influential. Klout[15] considers followers, but people who have "klout" do not necessarily have a huge number of followers – it is the right kind of followers that are important; those who are influential themselves, that are consuming the information, acting on it, and passing it on to their followers.

When investigating the structure of the web, a different picture emerges depending on the directionality of the links [16]. It would be fascinating to replicate this work with Twitter's Follower/Following network, but this is currently infeasible given the restrictions of the API. One interpretation of a website's PageRank is that it quantifies how easy it is to find a page through browsing [16]. If we compare this to conversations (or retweets) it's a measure of how easy it is to discover another user because they are retweeted, or having a conversation with someone else you know. The widespread connectivity on the internet is not the result of a few highly connected nodes [16], is this the case with Twitter? Applying clique discovery to a user's social graph may pull out the communities in which they participate [17].

Around 25.4% of posts are directed, indicating that there is a sparse and hidden network of connections within the declared set of friends/followers; the networks that matter are "made out of the pattern of interactions that people have with their friends or acquaintances" [18]. The number of people a user will communicate with saturates as a function of the number of followers; although people may follow a large number of people, they cannot tune it to the volume of posts produced. Most importantly, the study found that whilst the follower-following graph is dense, the network created by interactions is sparse. The power law exponent on the Twitter follower graph was about -2.4, similar to the value of the web and the blogosphere [1], however "researchers in psychology and sociology have repeatedly cast doubt on the practice of inferring meaningful relationships from social network connections alone" [3], which begs the question – how can we infer a meaningful relationship? The proposed solution, "interaction graphs", result in graphs that are different from the social connections graph, displaying significantly lower levels of "small world" properties, and fewer highly connected "supernodes". Human interactions are limited by time, who you make time to respond to or retweet is an indication of who, in your network, is most important or most informative to you.

Messaging has been found to be a better measurement of engagement than friend relationships on Facebook [19], often users have no interaction with up to 50% of

their Facebook friends [3]. There are obvious similarities in this to Twitter. When we consider reciprocity, we must take into account the ratio of giving to and taking from the community [20]. The rate of user activity on Twitter is influenced by social connectivity and it has been found that social relationships determined if users would remain active using a blogging system [1]. This makes sense; without a community, status updating on Twitter can seem like talking to oneself in public.

Research determining clique sizes on email found that whilst e-mail is more egalitarian (more inclusive of additional contributors) than face-to-face conversations, the number of people involved in the conversation was no larger [21]. Twitter is hard to compare to email, because whilst e-mail is often sent to just one individual, an undirected tweet is received by an unknown subset of the people who follow a user and possibly some who do not. Email is useful for getting information from weak ties and we believe that Twitter is also useful in this respect. Dunbar found conversational cliques to average 2.7 members (and rarely to exceed 4) whilst groups would be much bigger – up to 15 – he concludes that a maximum clique size of 4 is an inherent property in human speech [22]. Translating to Twitter, a clique would be the people who @ each other, whilst the group would be those people for whom the conversation would show up in their stream. Whilst the authors of [22] hypothesized that clique size would be bigger on email, this was found not to be the case. Due to the space limitation of 140 characters on Twitter, we might expect the clique size for a conversation to be smaller. However this is only in the case of people *explicitly* involved in the conversation, and so is hard to determine. E.g. person A may say something, person B may remark on it, and person C may respond to person B but not include person A's handle in the tweet. The "/cc @username" notation is designed for this kind of situation, but is not yet in wide use. Or, A makes a comment, B responds and C responds, but B and C do not follow each other; it is a stretch to say that B and C are in the same conversation as it is likely that neither will see what the other has said. Rather, A is carrying on two separate conversations with B and C, concurrently, on the same topic.

3 Background

PageRankinfers the popularity of a web page through measuring inbound links [23]. The similarity between link structures in following/followers is similar to the web and thus susceptible to the same kind of spam [24]. After analyzing spam behavior from both network and social perspectives, it was concluded that behavioral analysis might be more helpful to detect spammers than content analysis [24]. The full cycle of a meme started on twitter, #robotpickuplines (4 days, 17,803 tweets, 8,616 users) was analyzed; 14% of tweets were spam, (spam lagged slightly behind the meme trend). Spam accounts similar in age to legitimate accounts (spam accounts are/were not being suspended), spammers tweet more than legitimate accounts, and have a similar number of @ replies. What isn't included is the volume of incoming @ mentions – an important omission; it is easy to @ random users, it is extremely difficult to generate any significant number of responses to spam. Spammers had three times the total number of followers/following than legitimate users. Outgoing links from a typical user would lead to a celebrity, or other high-profile user; "popularity and legitimacy are indicated by high in degree and spam is indicated by high out degree."

However it is usually easy for a human to determine if a user is a spammer; it is clear that their engagement is low. People are not talking to them, and their content is not being retweeted. Thus, perhaps a better measure of a user's influence, is how much they are being talked to or retweeted. Like the web where "the prominence of authoritative pages is typically due to the endorsement of many relevant pages that are not, in themselves, prominent" [17], – a reply or a retweet could well be construed as a measure of endorsement. This approach reduces the number of links per person, API limits as of February 2010 allow around 10 days or the 100 most recent @ mentions and the last 200 tweets from a given user for one API call each. The HITS algorithm [25][26]has been used to identify implicit online communities by analysis of the links between web pages This required a vast amount of data and aggressive pruning, we think it is possible to find communities that a user is a part of on Twitter with a similar strategy. It has been proposed that the big difference of a Social Networks is the *clustering* – a result of communities within the network [13].

In graph theory, cliques in an undirected graph are subsets of its vertices such that there is a connection between every pair of vertices in the subset. Whilst our initial graph is directed, for the purpose of extracting cliques we use an undirected graph.

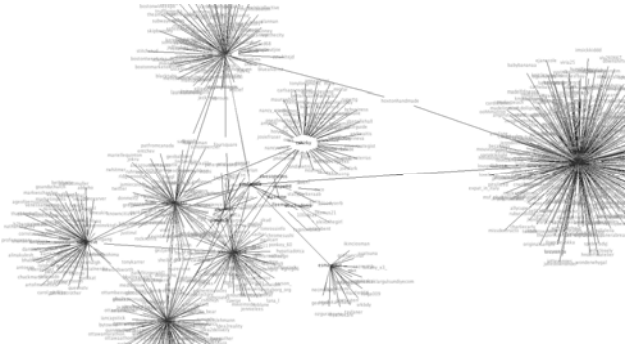


Fig. 2. Conversation Network for a Light User (@emdaniels)

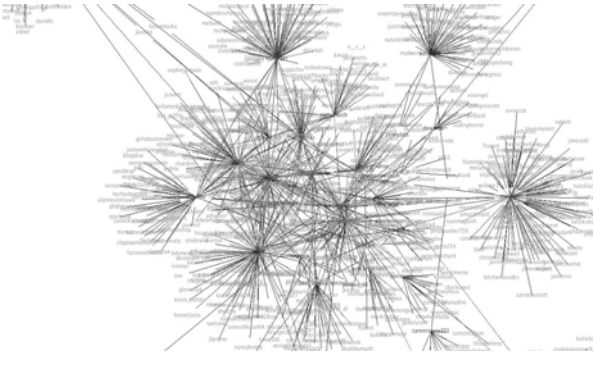


Fig. 3. Conversation Network for a Moderate User (@kittenthebad)

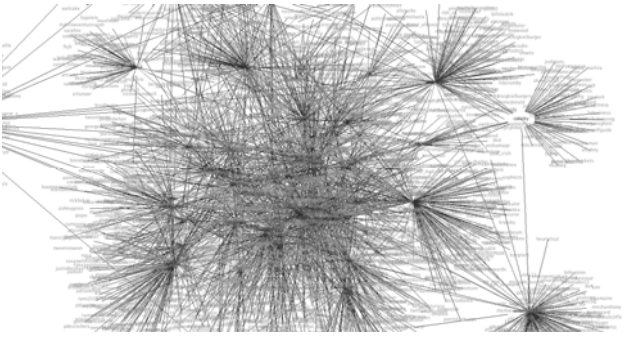


Fig. 4. Conversation Network for a Heavy User (@krusk)

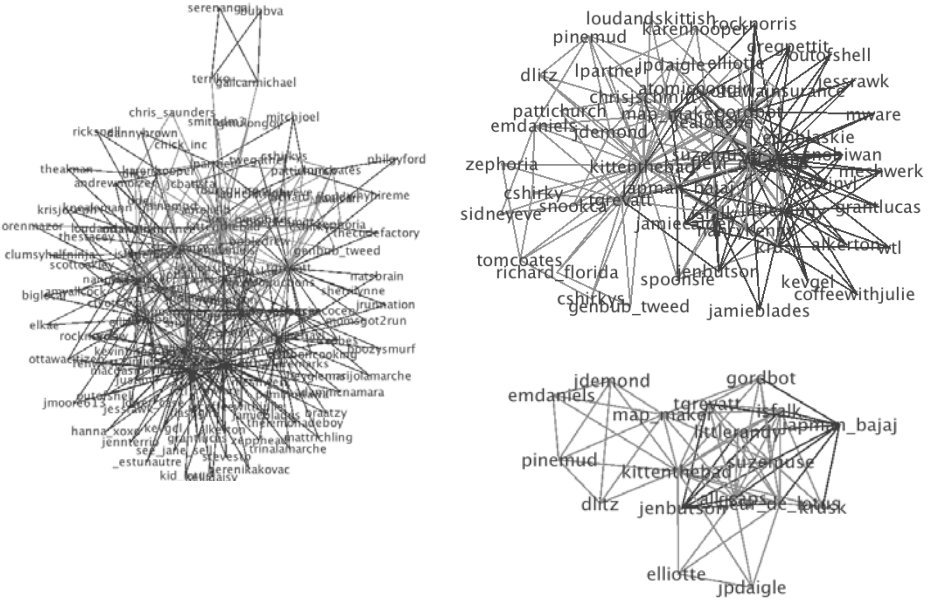


Fig.5. Cliques Size 3+, 4+ and 5+ (clockwise) for a Moderate User (@kittenthebad)

Fig. 6. Cliques Size 6+ for a Heavy User (@anitaborg_org)

In social network analysis, we should consider the homogeneity of users [27], any observation of the Twitter network tells us people use Twitter in a myriad different ways. We contend that it is more interesting and fruitful to build a picture user by user, to better characterize them on scales of conversation, community, and information diffusion.

Creating these networks allows us to identify cliques. Once identified, we can potentially identify what the clique is talking about – are they sharing what they're eating, events they're attending, or the stories of their day? Or, grouping around a specific topic? Homophily has been found on Instant Messaging networks – people who chat with each other, are more likely to share interests. This is also true, to a lesser extent, in a "two-hop" network – i.e. if A talks to B, and B talks to C, A and C are more to search for similar things[28]. Extrapolating from this, we can expect that people who talk to each other on Twitter are interested in similar information. This is the basis of a retweet - I'm interested in something and hope that my followers will be, too. A further area of research is determining if those we have conversations with are more likely to retweet our content.

4 Data and Methodology

The data comes from the Twitter public profiles of 54 users who either expressed an interest in having their network created or were requested by another researcher due to them being prominent figures in the discussion of the future of journalism. Everything was accessed through the public API, without need for authentication. There is an element of self-selection bias (all graphs visualized using the collected data are available at http://catehuston.com/prefuse_twitter).

The data is accessed through the Twitter API and generates GraphML (an XML based format for graphs), then visualized as a "Conversation Network" graph. A light user is shown in Figure 2, moderate user in Figure 4, and heavy user in Figure 5. Different shades of the connections represent the different types of interaction (outgoing/incoming/reciprocal). Data is collected only up to a depth of two due to API limits (i.e. every one the central user talks to/about and is mentioned by, and the same for every one of those people). We collected the last 200 tweets from the user and the last 100 mentions (or as many as are returned by the search API, which is typically limited to 10 days), and then adjust to cover the same date period for both. We then collect the same data from every person the central user has been mentioned by, and/or spoken to. This does mean that if a user is very popular, the length of time gathered for their followers is potentially quite short (around 2 days in the case of @cshirky – Clay Shirky). For a more typical user in our dataset, the time frame is a week to 10 days.

A simple maximal clique algorithm such as those described in [29-32] is run on the graph. Although the initial graphs are quite large, by deleting nodes with fewer connections than clique size (3+) we are able to dramatically reduce the search space and performance is reasonable (runtime near-instant using Haskell on a MacBook Pro). We then generate the graph (defined in GraphML) from the resulting cliques to produce graphs like those shown for a moderate user in Figure 5 and for a heavy user in Figure 6 (that user has up to cliques of size 7). Lighter lines mean both nodes are connected to the central user. It is interesting to note that a user might be

characterized as heavy from their conversation network, however be fairly light on cliques. This is exemplified in the graphs for one heavy user, shown in Figure 7; whilst that user has a large (conversation) network, it is more disconnected and the maximum clique size is just 4.The sample is not representative of the global Twitter network, users have a range of connection sizes (Figure 8).

Fig. 7. Conversation Network and Cliques Size 4+ for a Heavy User (@snookca)

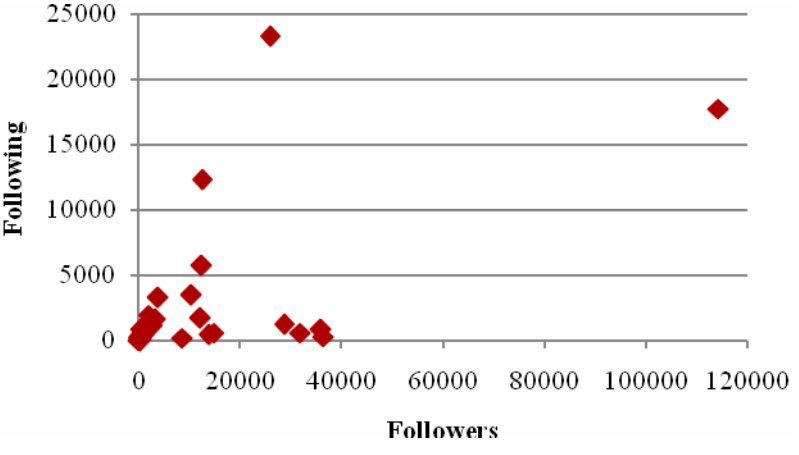


Fig. 8. Number of Followers vs. Number of Following

5 Results

The number of people a user converses with saturates as a number of followers [18]. This may means that the number of communities they are part of saturates as well. Figure 8 shows the relation of following to followers from our dataset, we can see that the relationship between number of followers and the number of people a user follows is not closely related, but in general, users tend to follow fewer than 2000 people. This is different from the results of [1], likely as the result of differences in the dataset. Our dataset is small, and not representative due to self-selection bias and the use of prominent thought-leaders on the future of journalism.

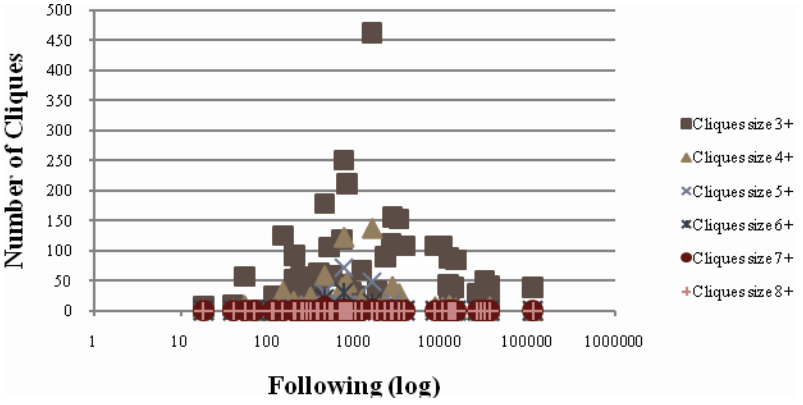


Fig. 9. Number of Cliques vs. Number of Following (log scale)

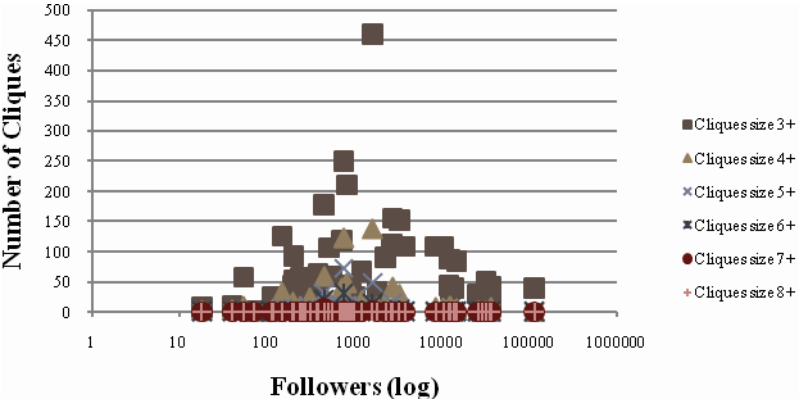


Fig. 10. Number of Cliques vs. Number of Followers (log scale)

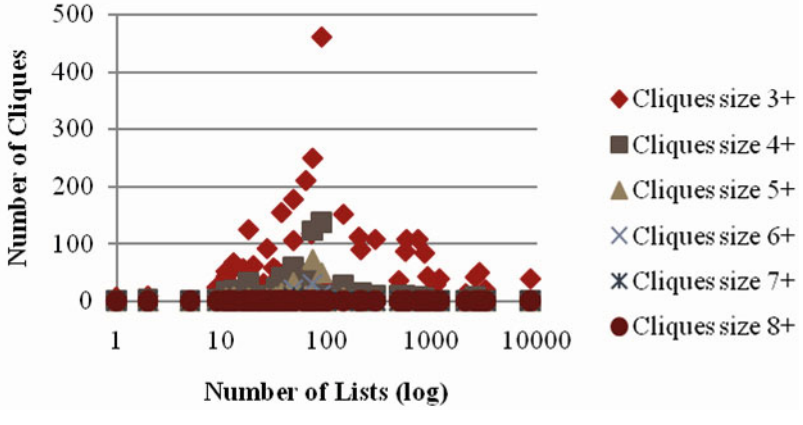


Fig. 11. Number of Cliques vs. Number of Lists (log scale)

Table 1. Results Overview

	Following	*Followers*	*Lists*	*Cliques*		
				size 3+	*size 4+*	*size 5+*
Mean	1750	7222	509	70	18	12
Median	539	782	45	43	9	5
Mode	264	198	37	31	9	1

For the number of cliques compared to people followed (Figure 9), we see an upward trend (for cliques of size 3+ at least), which tails off after a certain number of following is reached. Either after a certain point, following a large number of people means a user's network has become less cohesive or that they are sufficiently popular that the data covered too short a time period. We see a similar picture emerging when we compare the number of cliques to the number of people a user is followed by (Figure 10) and the number of lists a user is on (Figure 11).

An overview of the results is in Table 1, the mean/median/mode indicate a skewed distribution. Interestingly, the users skewing the clique distribution graph are different than those skewing the follower/following/list graphs. This suggests that users can be influential amongst the communities they are a part of, even if they have little influence outside that circle. The number of cliques could be construed as an indication of how connected the community of a user is. A user with a large conversation network, but few cliques has a more disconnected network than a user with a similar sized (or smaller) conversation network but a large number of cliques, particularly cliques of bigger sizes – small cliques can occur by chance, or due to a retweet.

6 Conclusion

In this paper we use @ mentions and retweets to quantify engagement. This is an improvement over following counts in a number of ways: firstly, this is much harder for spammers to game. Secondly, this is much more current picture of a user's engagement and allows for the capturing of changes in interaction patterns and partners over time. Thirdly, by characterizing a user's engagement, we think this can be built upon in order to characterize a user's level and type of influence. We use visualization to explore user's communities.

Our initial findings suggest that by pulling out the cliques a user is a part of we can visualize the communities they are a part of. Because of privacy concerns around making public more public, graphs were created by request and users have reported finding value in knowing the outer circle of connections exposed by the clique visualizations; those who are strongly connected to those the central user is strongly connected to. We believe there is the potential to build a recommendation system on top of this, in a similar way to how Amazon recommends purchases and Facebook recommends friends. This has already been investigated in academic networks [33]. One possible avenue is determining if these networks are predictive - if user A is in a clique with users B, C and D, and users B, C and D are in a clique with user E, is it likely that eventually user A will end up interacting with user E? If this is the case, there is the potential to create a recommendation engine we hasten that process. This is the origin of the idea of cliques within social networks, the potential interest in two of one's acquaintances in connecting.

Further, this representation of graphs is closer to the way that a user remembers their network. In [34], it is noted that we are most likely remember out connections through connections in common, i.e. one might think of "Uncle Bob", and from him remember "Aunty Ann" and their daughter… thus we can view the core hub of the clique network as those connections a user is most likely to remember, as there are multiple paths to reach them. This is based on the idea that these connections are stronger ties, which they may not be as the directionality of the link is not considered. If we restrict our graphs to *reciprocal* links (mutual interest), or *outgoing* links (connections a user has demonstrated an interest in), or both, we may get a better recommendation set. Additionally, if presenting the recommended connections as a list as well, potential new connections could be ordered by the number of cliques they are a part of.

We also think there is significant potential for characterizing individual users based on the results of the clique finding algorithm, for instance distinguishing between what are commonly described as connectors, broadcasters, celebrities, mavens etc. This will require classification of the different categories and much more data to ensure that it is representative. We can also restrict relationships to reciprocal ones only, for example. For a broadcaster (someone who gets re-tweeted a lot but interacts little) this will likely be illuminating.

References

[1] Java, A., Song, X., Finin, T., Tseng, B.: Why We Twitter: An Analysis of a Microblogging Community. Designing Privacy Enhancing Technologies, 118–138 (2009)

[2] Ryan, W., Hazlewood, W.R., Makice, K.: Twitterspace: A co-developed display using Twitter to enhance community awareness. In: PDC 2008, pp. 230–234 (2008)

[3] Wilson, C., Boe, B., Sala, A., Puttaswamy, K.P.N., Zhao, B.Y.: User interactions in social networks and their implications. In: Proceedings of the Fourth ACM European Conference on Computer Systems, pp. 205–218. ACM, Nuremberg (2009)

[4] Joinson, A.N.: Looking at, looking up or keeping up with people? motives and use of facebook. In: Proceeding of the Twenty-Sixth Annual SIGCHI Conference on Human Factors in Computing Systems, ACM, Florence (2008)

[5] Statistics | Facebook, `http://www.facebook.com/press/info.php?statistics`

[6] O'Reilly, T., Milstein, S.: The Twitter Book. O'Reilly Media, Inc., Sebastopol (2009)

[7] Cha, M., Mislove, A., Gummadi, K.P.: A measurement-driven analysis of information propagation in the flickr social network. In: Proceedings of the 18th International Conference on World wide web, pp. 721–730. ACM, Madrid (2009)

[8] Boyd, D.: Tweet, tweet,retweet: Conversational aspects of retweetingontwitter. In: Proceedings of HICSS, vol. 43 (2010)

[9] Crawford, K.: Following you: Disciplines of listening in social media. Continuum: Journal of Media &# 38; Cultural Studies 23, 525–535 (2009)

[10] bit.ly, a simple urlshortener, `http://bit.ly/`

[11] The Science of ReTweets Report | Dan Zarrella, `http://danzarrella.com/the-science-of-retweets-report.html#`

[12] Topsy - A search engine powered by tweets, `http://topsy.com/`

[13] Newman, M.E.J., Park, J.: Why social networks are different from other types of networks. Physical Review E 68, 36122 (2003)

[14] Olfati-Saber, R.: Evolutionary dynamics of behavior in social networks. In: 2007 46th IEEE Conference on Decision and Control, pp. 4051–4056 (2007)

[15] Klout - Twitter Analytics - Measuring Influence Across The Social Web, http://www.klout.net/
[16] Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A., Wiener, J.: Graph structure in the web. Computer Networks 33, 309–320 (2000)
[17] Kleinberg, J.M., Kumar, R., Raghavan, P., Rajagopalan, S., Tomkins, A.S.: The web as a graph: Measurements, models, and methods. In: Asano, T., Imai, H., Lee, D.T., Nakano, S.-i., Tokuyama, T. (eds.) COCOON 1999. LNCS, vol. 1627, pp. 1–17. Springer, Heidelberg (1999)
[18] Huberman, B.A., Romero, D.M., Wu, F.: Social Networks That Matter: Twitter Under the Microscope. In: SSRN eLibrary (December 2008)
[19] Golder, S.A., Wilkinson, D.M., Huberman, B.A.: Rhythms of social interaction: Messaging within a massive online network. In: Communities and Technologies 2007: Proceedings of the Third Communities and Technologies Conference, p. 41. Michigan State University (2007)
[20] Preece, J.: Sociability and usability in online communities: determining and measuring success. Behaviour& Information Technology 20, 347–356 (2001)
[21] Loch, C.H., Tyler, J.R., Lukose, R.: Conversational Structure in Email and Face to Face Communication. Draft, submitted to Organization Science
[22] Zhou, W., Sornette, D., Hill, R., Dunbar, R.: Discrete hierarchical organization of social group sizes. Proc. R. Soc. B, 439–444 (2005)
[23] Brin, S., Page, L.: The anatomy of a large-scale hypertextual Web search engine. Computer Networks and ISDN Systems 30, 107–117 (1998)
[24] S. Yardi, D. Romero, G. Schoenebeck, and danahboyd, "Yardi," Detecting Spam in a Twitter Network, http://www.uic.edu/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/2793/2431
[25] Gibson, D., Kleinberg, J., Raghavan, P.: Inferring web communities from link topology
[26] Kumar, R., Raghavan, P., Rajagopalan, S., Tomkins, A.: Trawling the Web for emerging cyber-communities. Computer Networks 31, 1481–1493 (1999)
[27] Hogg, T., Szabo, G.: Dynamics and diversity of online community activities. EPL (Europhysics Letters) 86, 38003 (2009)
[28] Singla, P., Richardson, M.: Yes, there is a correlation: - from social networks to personal behavior on the web. In: Proceeding of the 17th International Conference on World Wide Web, ACM, Beijing (2008)
[29] Österg, P.R.J.: A fast algorithm for the maximum clique problem. Discrete Applied Mathematics 120, 197–207 (2002)
[30] Pardalos, P.M., Xue, J.: The maximum clique problem. Journal of Global Optimization 4, 301–328 (1994)
[31] Masuda, S., Nakajima, K., Kashiwabara, T., Fujisawa, T.: Efficient Algorithms for Finding Maximum Cliques of an Overlap Graph (1986)
[32] Babel, L.: Finding maximum cliques in arbitrary and in special graphs. Computing 46, 321–341 (1991)
[33] Liben-Nowell, D., Kleinberg, J.: The link prediction problem for social networks. In: Proceedings of the Twelfth International Conference on Information and Knowledge Management, pp. 556–559 (2003)
[34] B.J. Hogan, Networking in everyday life, University of Toronto (2009)
[35] What will (2011), bring for journalism? Clay Shirky predicts widespread disruptions for syndication » Nieman Journalism Lab, http://www.niemanlab.org/2010/12/what-will-2011-bring-for-journalism-clay-shirky-predicts-widespread-disruptions-for-syndication/

The Design, Development and Application of a Proxy Credential Auditing Infrastructure for Collaborative Research

Christopher Bayliss[1], Richard O. Sinnott[2], Wei Jie[3], and Junaid Arshad[4]

[1] National e-Science Centre, University of Glasgow, United Kingdom
c.bayliss@nesc.gla.ac.uk
[2] Melbourne eResearch Group, University of Melbourne, Australia
rsinnott@unimelb.edu.au
[3] School of Computing, Thames Valley University, United Kingdom
wei.jie@tvu.ac.uk
[4] School of Computing, University of Leeds, United Kingdom
sc06ja@leeds.ac.uk

Abstract. Single sign-on and delegation of privileges are fundamental tenets upon which e-Infrastructures and Grid-based research more generally have been based. The realisation of single sign-on and delegation of privileges in accessing resources such as the UK e-Science National Grid Service is typically facilitated by X.509-based Public Key Infrastructures (PKI) and exploitation of proxy certificates. This model can be categorised by authentication-oriented access and usage of resources. It is the case however that proxy certificates, can potentially be obtained and abused by a malicious third party without the knowledge of the holder. In this paper we describe a novel proxy auditing solution that addresses this issue directly. We describe the design and implementation of this solution and illustrate its application in widely distributed and heterogeneous research environments.

Keywords: grid computing, grid security, user authentication, public key infrastructure, proxy certificate.

1 Introduction

In current e-Infrastructure environments, authentication and authorisation of users when accessing resources are essential functionalities that need to be supported. Authentication to facilities such as the UK e-Science National Grid Service (NGS, http://www.ngs.ac.uk) is predominantly achieved through public key infrastructure (PKI) and use of X.509 [1] certificates issued by the UK e-Science Certificate Authority (CA) (http://www.grid-support.ac.uk/ca). Whilst other authentication models have also been explored including federated authentication models of access and usage based upon the Internet2 Shibboleth technology [2] in UK Joint Information Systems Committee (JISC, http://www.jisc.ac.uk) funded projects such as SHEBANGS [3], ShibGrid

G. Babin, K. Stanoevska-Slabeva, P. Kropf (Eds.): MCETECH 2011, LNBIP 78, pp. 211–226, 2011.

[4] and SARONGS [5], the primary and most commonly adopted authentication model by the research community is still based upon X.509 PKI-based authentication where users acquire and maintain their own X.509 certificates and use them to create proxy credentials when submitted jobs or accessing data on resources such as the NGS.We note also that the UK e-Science CA also issues host certificates that can be used for similar purposes. Proxy certificates are commonly used to create a certificate with a minimal subset of the capabilities of the parent certificate, most commonly period of validity, making a certificate that is safer to delegate.

The primary middleware that is deployed on the NGS is the Globus Toolkit [6]. Globus has implemented a model of authentication based upon the Grid Security Infrastructure (GSI) [7]. GSI incorporates essential features to support single sign-on (SSO) and delegation of privileges (also often referred to as delegation of rights). In SSO, access to multiple distributed and autonomous resources, e.g. different NGS HPC clusters, is achieved with a single authentication, i.e. without repeated authentication challenge/responses from each cluster. With delegation of privileges, users are able to make their credentials available to Grid resources to act on their behalf. In realising this SSO and delegation of privileges, GSI relies on proxy certificates. In contrast to end user or host certificates which in the UK e-Science community are signed by the UK e-Science CA directly identity management to a local registration authority) , proxy certificates are signed using the private key of the user or host certificate itself. Proxy certificates can also be derived from other proxy certificates using the certificates corresponding private key for signing. By signing each certificate with a predecessors private key, a connection between derived proxy certificates is established that allows Grid resources to resolve the certificate chain up to the user/host certificate and eventually to the issuing CA. Such a chain is shown in Fig. 1. Establishing this chain of trust ensures that proxy certificates are trustworthy, i.e. ultimately that they have been issued by the UK e-Science CA whose processes for issuance and revocation of certificates, for management of the underlying PKI etc are accepted both by the Grid users and the resource providers themselves.

Whilst proxy certificates allow for SSO and delegation of privileges to be achieved, they are also a potential danger to the overall security of the Grid infrastructure itself and to the disparate end users themselves. Thus whilst the private key of a user credential is normally encrypted and requires a strong password to use, private keys of proxy credentials are generally unencrypted and stored on the local file system of the Grid resource protected only by file permissions. This model is not a design mistake, but a key requirement that is used to support SSO and delegation of privileges, i.e. since SSO implicitly demands that users only enter their password once and not every time that their proxy credential is used or delegated. To minimise the threats of proxy credentials, most Grid middleware (including Globus and GSI) set a default proxy certificate validity to a much shorter time-span than the life of the X.509 credential itself the default for proxy credential lifetimes is set to 12 hours.

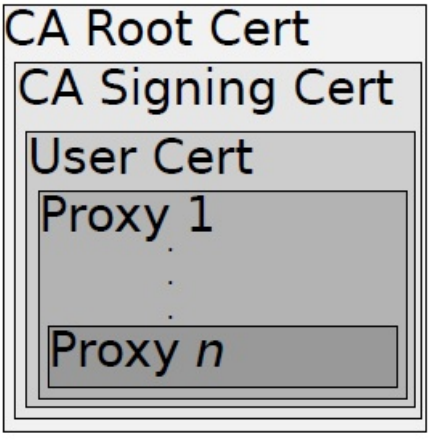


Fig. 1. A certificate chain for a proxy certificate of depth n

We note that while attempting to create a proxy with a lifetime beyond that of its parent should render it invalid, a suitably prepared attacker may only need a few minutes by using approaches described in [8].

When selecting a lifetime for a proxy certificate it is important to ensure it will remain valid throughout its usage. When submitting a job to a queue of indeterminate length this can be problematic and results in users setting lifetimes significantly longer than required to compensate for unpredictable latency. Therefore, proxies may well have a significant lifetime remaining at the end of the task they were created to perform increasing the window of opportunity an attacker has for using a stolen certificate. While [9] suggests that proxies be invalidated by adding a CRL distribution point to their proxy certificates this is not a viable option in a Grid environment where most certificates are managed by individual users for two reasons. Firstly, most, if not all, the proxy certificate generating tools available do not offer CRL generation as an option and, secondly, CRLs are not checked regularly enough to ensure the revocation was distributed before the proxy expired.

A further challenge is with delegation of privileges an essential component for successful e-Infrastructures. To understand this, consider the follow representative scenario of Grid usage. A user submits a computationally intensive job to run on an NGS compute cluster but their associated input data exists on a different NGS data cluster. The results themselves are required to be written to a local campus Grid resource associated with the NGS, e.g. in the case of Glasgow this might be the ScotGrid resource (`http://www.scotgrid.ac.uk`) which itself is a full partner of the NGS. To support SSO and delegation of privileges, the initial NGS compute cluster resource may be presented with a proxy certificate from the user who uses a command such as grid-proxy-init, voms-proxy-init, or exploits a credential repository such as MyProxy. Irrespective of how the proxy credential is created, it is subsequently made available to a particular cluster

worker node through a Grid mapfile mapping to a local HPC account. As part of the job execution, this proxy credential can then be used to create a further proxy credential used to access and securely copy data from the NGS data cluster, e.g. through gridFTP. Once this data is returned and job execution proceeds and completes, a final proxy credential can be created that is used to return the final resultant data sets to the local campus Grid resource, e.g. ScotGrid.

As seen in this scenario, resource-oriented delegation of privileges of user credentials is supported that allow jobs to act on behalf of the end user (represented by their original proxy credential). The main issue with this model however is that multiple proxy credentials now exist on multiple distributed clusters. Should one of these clusters become compromised then the proxy credential can subsequently be used to create further proxy credentials and used to access other remote resources, masquerading as the original user. This whole process of masquerading as the user can occur without any knowledge of the user themselves who created the initial proxy credential when submitting their job. They may well (quite rightly!) assume that local NGS and/or ScotGrid garbage collection activities take place after running their jobs which will automatically remove proxy credentials and/or temporary files that have been created in executing the compute/dataoriented tasks. This assumption may well be naive however, and as a result security threats and dangers on wider use of their proxy credentials may well exist.

It is emphasised that the proxy credential SSO and delegation of privileges model is especially open to the weakest link security paradigm. That is, should any resource in the Grid be compromised by a malicious third party and they manage to gain elevated operating system privileges on that resource, they also gain access to all proxy credentials that are delegated to that resource at the time of the attack and can subsequently attack other resources under the guise of a valid user with valid proxy credentials, in so doing garnering further delegated proxy credentials. These can then be used to explore and exploit potential system vulnerabilities on other resources and to launch distributed denial of service attacks (amongst other worst case scenarios).

Even more complex arrangements are possible when a credential management service such as the SARONGS system is used. This allows for the dynamic creation of short lived, low assurance X.509 certificates to allow users without a certificate to access the NGS via translation of SAML assertions from Internet2 Shibboleth-based Identity Providers.

Obviously many of these issues are caused by allowing e-Researchers access to resources such as the NGS to compile and run or simply execute arbitrary code based solely upon authentication through GSI. In the GSI model, a locally maintained Grid mapfile is used to map the distinguished name (DN) of the user as included in their X.509 certificate with a local user account on the Grid resource as shown in Fig. 2.

The GSI-based model provides users with flexibility and is relatively easy to implement and support for resource administrators, but also represents a clear danger in that users can execute arbitrary codes. A more secure model would

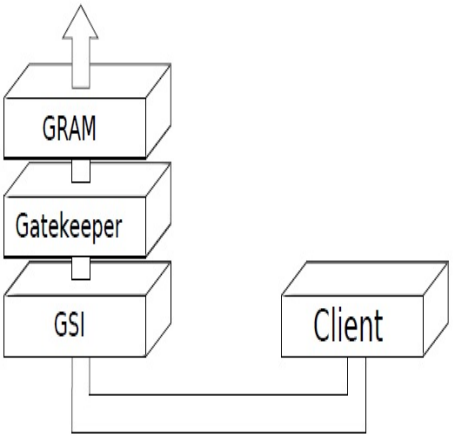


Fig. 2. A simple Globus stack

be to support finer grained authorisation where the users themselves do not get access to local accounts to do stuff, but access to services that are fixed and targeted at their needs. Thus for example the NGS is currently exploring GT4 based hosting of services, e.g. a BLAST service for biologists, a Gaussian service for chemists, as well as support for portals where a predefined set of applications is made available. However it is still the case that the vast majority of people accessing and using the NGS are using GSI models of access and usage (including GSISSH) and this seems likely to remain the case for the foreseeable future.

In this context, it is thus highly desirable to reduce the overall risk associated with proxy credentials ideally through transparent extensions to mainstream Grid middleware as deployed on national facilities such as the NGS, i.e. Globus and its use of GSI. We note that it is not realistic to simply deprecate use of authentication-only models since for many researchers, this ability to compile and execute their own simulation codes is essential. To tackle this , one model which has been put forward in the JISC funded Proxy Credential Auditing Infrastructure for the UK e-Science National Grid Service (PCA, `http://pca.nesc.gla.ac.uk`) and described in this paper, is to extend the Grid infrastructu capabilities of the NGS and similar resource providers with monitoring and auditing services for proxy credential usage and tracking. The primary requirements for this facility were that it should allow audit-enabled GSI resources to automatically capture proxy credential usage and send this and related information to one or more targeted secure, online services for tracking proxy credential usage. These services should allow proxy credential usage information at the:

- Individual user level — so that individual users are generally aware of their credential usage and can identify when their credentials are potentially being misused;

- Virtual Organisation (VO) usage level — so that VO administrators and the members involved are able to track the access to and usage patterns associated with their VO and by its members. Thus it might be the case that a particular VO has been set up to only use particular resources, e.g. particular NGS nodes. When a proxy credential is used to access other resources not identified as part of the VO, then this might highlight that a potential misuse of a proxy credential is taking place (or has taken place).
- Resource provider level — so that service providers can themselves monitor the usage of their own facilities by their user communities and detect as early as possible any abuses or misuses of credentials, and where appropriate revoke proxy credentials; update certificate revocation lists and update Grid mapfiles, e.g. remove the DN and account information for compromised certificates.

In this work we acknowledge that the approach we are taking represents a pragmatic and realistic model for improving overall security rather than a guarantee of overall security. We recognise that GSI-based access and usage is likely to continue to be the mainstream approach in accessing resources such as the NGS and similar international facilities for some time to come. Our aim is thus to provide a mechanism for rapid detection of credential abuse that address key stakeholders demands.

Furthermore, this work also underpins the areas of granularity in supporting n-tier based approaches. In terms of granularity of access, finer-grained authorisation approaches such as those based upon Role Based Access Control (RBAC) depend upon authentication. Knowing the identity of an individual requesting access to a particular resource is the first step in deciding what roles this person might have which can subsequently be used for finer-grained authorisation decisions, e.g. using technologies such as XACML, OAuthZ or PERMIS. If a proxy credential has been compromised, then a masquerader attempting to access a remote resource may well be indistinguishable from a legitimate user since the Policy Enforcement Point (PEP) — Policy Decision Point (PDP) that might well support finer-grained RBAC models, may well be configured to pull further attribute certificates from a remote attribute authority to make a local access control decision. Similarly, n-tier systems function primarily based upon passing of credentials for authentication and authorisation. Compromised authentication tokens given as proxy certificates, are indistinguishable between tiers unless other challenge/responses are demanded, which violate the intrinsic model and benefits of SSO. In short, if authentication systems are compromised then more granular n-tiered authorisation systems may well be redundant!

Initial work on proxy credential auditing was described in [10] and [11]. A Globus incubator project has been established to support enhancements and refinements to this work. This work was explored in the course of the PCA project indeed it formed the initial starting point for the work, however a different architecture and system design has since been undertaken for reasons discussed below.

More generally a body of work has been undertaken on auditing of stack based systems that support message passing paradigms. Typically in this model, an

incoming message can pass through several different application layers as it traverses the stack making monitoring individual calls problematic since calls are often logged independently between them making identifying events caused by a specific call problematic. Systems like DTrace [12] and SystemTap [13] have been developed to address run-time logging information. In the absence of a modified application these systems use kernel level services to monitor the target. Monitors are then bound to components throughout the system which generate events when triggered which are subsequently sent to a central monitoring system which filters and processes them in accordance with a script supplied by the user. However, these systems were designed to monitor single systems and have access to robust methods of associating events with their cause via process identifiers. Furthermore approaches such as XTrace [14] have also been developed to supporting logging of network applications more generally but would have required the introduction of a modified stack which is discussed later. None of these address the fundamental problem of proxy credential usage in collaborative and loosely coupled research environments such as Grids.

2 PCA Software Architecture

The basic model of authentication to an NGS resource through the Globus software stack is illustrated in Fig. 2. The initial work on the PCA project explored the prototyping work described in [3] and [11] where the focus was upon implementation of an audit-enabled enhancement to GSI, i.e. replacing the lower level of Fig. 2 completely. Whilst supporting the basic auditing capabilities, the work described there had issues in its widespread deployment. Most importantly, it required development and roll-out of a new version of GSI to resource providers such as the NGS. There are numerous pragmatic aspects which make this non-trivial to achieve and other models were thus required.

An improved model of auditing is to provide a transparent auditing layer to the GSI software stack as shown in Fig. 3. This is the approach that has been taken in this work. This architecture was adopted for several reasons. Firstly, obviously and most importantly the basic requirements of the system were to allow appropriate parties to observe activity associated with a proxy credential in order to allow both appropriate and inappropriate activity to be identified. Another design requirement was due to the understandable reticence of system administrators of facilities such as the NGS to want to install applications which may worsen the performance, complexity, stability or security of their resources. Therefore, there was the need to integrate with existing software stacks with as little impact as possible. These requirements ultimately precluded several of the methods used by other similar systems as they require the modification of key components which have performance degradation consequences and/or issues of complexity in system-wide deployment.

The original design of [10] was based upon embedding the sink of the auditing information, i.e. a secure web service that should be notified when the certificate was used or more precisely an event was raised when the proxy certificate

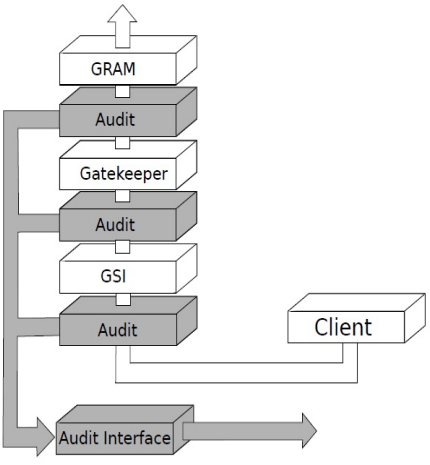


Fig. 3. A simple Globus stack

(or certificates derived from the proxy certificate) was received by a GSI-audit enabled service. This single sink of information model had several limitations especially when propagating data to third parties. A different model is for data to be collected at the resource on all requests without requiring special modifications of the credentials. This eliminates a problem whereby nonaudit aware certificates are hidden from the system resulting in incomplete data or, worse, a mechanism for malicious users to disable tracking of their actions. Instead events are published at the resource using information from the certificate the user cannot alter and interested parties can then acquire events as needed.

This design decision introduces a new problem however as it uses a pull instead of a push since it requires consumers to know which sites have data they are interested in. Should a proxy cert be used on an unexpected site the user has no way of discovering this unless the attempt raises an event which includes details of the new system. However, as we associate events with the certificate used to invoke them the user can locate them by searching. Thus given a proxy certificate such as the one shown in Fig. 1 all proxy certificates generated share the same sequence of parent DNs from the root of the issuing CA to the end entity. More generally, each entity that requires access to the data discovers events using data known to them. Users, as mentioned above, can use details of their certificate to obtain associated event information.

Altering the model to collate events at the service also changes the security model of the system. In work such as [11], events are forwarded to a logging service whose URL was embedded in the proxy certificate. This provides two potential security problems. Firstly, it is possible for a malicious party to embed the URL of a third party in a proxy certificate and then use it to access services in order to generate traffic as part of a Denial of Service (DoS) attack. Secondly, it would be possible for an attacker to direct services to forward events to a server which maintained the HTTP connection for as long as possible opening

the possibility of submitting sufficient jobs to exhaust the servers supply of ports again causing a DoS.

In environments such as the NGS, it is highly desirable to refine individual level auditing information of resource usage. One mainstream way that this is achieved is through establishment and support of Virtual Organisations (VO). In this model, technologies such as VOMS [15] are used to establish the VO structure including the roles and privileges assigned to individuals in that VO. This information when included in a proxy credential (as an extension to the X.509 credential itself) is subsequently used by solutions such as LCMAPS and LCAS to transparently map VO-specific resource requests onto Grid resources targeted to the needs of the VO itself. Often this is to map VO user requests onto pooled accounts established on resources such as the NGS for the purposes of that VO. With this model, a VO administrator is typically tasked with establishing the software environment on the Grid resources, i.e. configuring the software and data resources associated with that VO. As with individual level usage tracking through the DN and the hierarchy of parent certificates outlined in Fig. 1 VO specific usage can be logged and auditing through extracting the associated VOMS attribute certificates and the DN of the users embedded into the certificates. Key to this solution is to recognise that the same individuals can belong to multiple VOs and thus want/need access to multiple auditing services.

By separating the event processing between the service, user and applicable third parties the system can support many different groups with differing requirements. The type of questions the administrator of a site may wish to answer will likely be based around who and how their resources are being used. Clients will likely only wish information relating to when their credentials were used and if they accessed further sites. Third parties may similarly attach to a filtered feed appropriate to their needs and potentially emit events from their own interface for consumption by other entities.

2.1 Auditing Events

Based on the previous discussions, the PCA system has been designed to offer a wider range of event options instead of simply credential acceptance and logging / auditing information capture. A key requirement was how to associate multiple events, potentially taking place across multiple machines, with their causal predecessors. We currently solve this problem by associating events with a connection object which represents a specific TCP connection between a client and local service. We create the connection object after the TCP connection is established but before any security context is established in the SSL or GSI layer. Should the context fail to be established we emit a connection failure event if successful we emit a connection succeeded event. In both cases if a client credential is provided it is associated with the connection by a further event.

We assume that we can insert extra data into a request as it passes up the services stack. This allows us to insert an identifier that event emitters can use to associate events with the correct connection. Our current use cases use both

HTTP messages and / or shell scripts to propagate through the stack which are trivial to modify in this way.

The system uses an event logging service at each resource which is responsible for collecting event data from a specific site. It is then responsible for providing this data to external entities in a secure and useful manner.

2.2 Auditing Event Emitters

The original service described in [3] used a modified GSI library which performed a SOAP call to push the event to the logging service. In addition to the previous limitations, this design had a major drawbacks in that the SOAP call introduced latency into the stack i.e. the GSI handshake would not complete until the call had completed. As outlined above, the general design pattern for an audit/usage event emitter was based on the proxy pattern. This was selected as it allows emitters to be inserted between the layers of the existing stack without requiring any modifications to software forming important parts of the system. As noted previously, we felt that this minimum impact approach was most likely to be accepted by site administrators who would be hesitant to install modified replacements to vital components such as the GSI security library or the GRAM job manager.

2.3 Credentials

The original design used in [3] required an SSLv3 extensions mechanism to embed a field containing the URL of the event logging service. This requirement was dropped in the PCA solution as it allowed hostile users to simply omit the extension from their certificates to avoid their actions being monitored. Instead events are always captured at the service site with the option of events being sent to further sites if requested. This design allows events to be forwarded to potentially many third party sites allowing usage to be monitored.

2.4 External Interface

As the PCA design no longer pushed events directly to a location accessible by the user there is a need to provide an external interface for clients to acquire auditing and usage information. As we are no longer simply forwarding events as they occur it becomes possible to offer more sophisticated services. For example, a consumer may not simply wish to receive a stream of all events generated but to filter them into, potentially several, different streams matching specific patterns. Such patterns could capture undesired behaviour, such as proxy rejection, or potentially malicious behaviour, such as a the usage of a sub-proxy certificate not generated by the user.

2.5 Analysis

Simply collecting logs of events is in of itself not directly useful. There is a need for the events to be analysed to identify patterns of usage both, permitted and

forbidden, so that useful information can be extracted from the raw data. Again, the PCA design permits the different entities in the system to focus on processing data for their own needs.

3 Implementation

The entire project is currently written in the Ruby programming language. The event logging service is a Ruby on Rails application providing a RESTful API for event emitters to access.

3.1 Events

There was a need for some format with which to publish generated events. As we commonly produce a time series the Atom Syndication format [16] presents itself as an obvious candidate being an IETF standardised XML format for publishing data in reverse chronological order. Adopting an existing format allowed us to exploit the resources developed for it and reduce our development time. Events themselves are are stored in a MongoDB document store [17]. This was selected for the scalability, functionality and overall compatibility with the event information that is captured. That is, given that events are modelled as a bag of name value pairs associated with a connection a document store offers a close semantic match and allows for highly performant key-value stores look-ups.

3.2 Event Emitters

Development of the complete set of event emitters is currently on-going. Initially we focused on the development of an SSL server which emits connection and security events and allowed for security information to be captured and logged by an event emitter. Following this a primitive GRAM job manager event emitter was developed which emits events when it receives a Globus based job (either through globusjob-submit or globus-job-run). The event emitters wrap these services and are completely transparent to GRAM itself and thus not intrusive into the overall Globus software. This software has been developed and tested on an NGS-like cluster at the National e-Science Centre at the University of Glasgow, i.e. a cluster with the same job submission software stack as currently exists on the NGS.

3.3 Public Interface

As mention previously the public interface is implemented as a Ruby on Rails application that provides an to the secure web service interface to the event store. It provides both a HTML and programmatic atom based access to the data. Currently, the secure web service front end supports a simple query interface which permits querying the data store using URL encoded queries. Queriable attributes include the DN of the subject one of the members of the certificate

chain of a credential and the name of an action as a string using regular expressions. It is also possible to specify maximum and minimum points in time within which an events occurred or when a connection was established. By encoding the query in the URL instead of the HTTP request body queries can be treated as first class entities by the system and provided with the same services as a connection and events group is allowing users to create a filtered view of events that suits their needs.

4 DAMES Case Study

To demonstrate the application of the PCA proxy credential auditing infrastructure we have identified a portfolio of projects using the NGS and related resources. The ESRC funded DAMES project [18] is a prime candidate for proxy credential usage tracking. The DAMES project is focused upon the challenges of data management facing the social sciences. It is the case, as with many other domains, that the social sciences are facing unprecedented challenges in the volume and heterogeneity of data sets from an increasingly diverse portfolio of data providers. When deal with issues around e-Health for example, it is often necessary to leverage data resources crossing the social, clinical and geospatial domain where individual and autonomous data providers are extremely aware of, and bound by criteria associated with information governance on data access and usage. The DAMES project has four primary areas of data management in the social science domain and is developing a family of specialist data environments to tackle the challenges that arise. These include:

- Grid-Enabled Educational Data Environment (GEEDE) — where researchers are able to access and analyse national and international data resources associated with education and associated qualifications;
- Grid-Enabled Occupational Data Environment (GEODE) — where researchers are able to access and analyse data resources associated with occupational classifications and associated coding systems;
- Grid-Enabled Minority Data Environment (GEMDE) — where researchers are able to access and analyse a variety of data resources associated with minority and ethnicity;
- Grid-Enabled Health Data Environment (GEHDE) — where researchers are able to access and analyse a variety of data resources associated with clinical and health related resources;

The GEODE and GEHDE related work is described in [19] and [20] respectively.

All of these data environments require access to and usage of statistical analysis tools. A variety of such solutions are widely used in this space including SAS, STATA and R. In the on-going work in DAMES we have supported exploitation of R on large scale HPC facilities. This was due in part to R being open source and already deployed on facilities such as the NGS and the expertise of the social science community. R itself can be used for a variety of statistical analysis.

Of particular relevance to DAMES is the coding, recoding and subsequent statistical analysis of social science data sets. Many of these data resources are large and can require significant processing, especially when re-purposing is needed. One example from GEMDE is the classification of UK Census data from the UK Data Archive related to the ethnic classification of the UK population over the past 40 years. Researchers may want to analyse this data from a variety of perspectives. Considering white/British and Others; considering white British; black African; Indian; Pakistani/Bangladeshi; as a unit group, and Chinese and Others as another group etc and comparing this data at a regional levels versus a national level and indeed comparing across regions. Different coding and classification schemes are used for this purpose, and re-purposing of the data is often needed.

To support this, the DAMES project intends to exploit the UK e-Science NGS however our work here is based on a representative cluster in Glasgow, i.e. using the same software stack. The social scientists involved in the DAMES project exploit a portal developed using the LifeRay framework itself and accessible through the UK Access Management Federation (`http://www.ukfederation.org.uk`). Details of how the SAML assertions provided by the UK Access Management Identity Providers are used to configure the portal contents, and subsequently used to create user X.509 proxy credentials are described in [21]. At present users are able to create their own X.509 proxy credentials through a targeted portlet that exploits a dedicated MyProxy service. Other approaches also exist for this purpose as outlined previously, e.g. SARoNGS.

The auditing information that was obtained in the execution of the R script, i.e. to illustrate where these R jobs were, when and by whom these jobs were run is shown in Fig. 4, Fig. 5 and Fig. 6. The result shown is the browsable HTML

Fig. 4. The overview page of the PCA interface showing recent activity

Fig. 5. The PCA interface showing details of a certificate

Fig. 6. The PCA interface showing details of an event

interface displaying a summary of discrete connections. Each connection shows the end point addresses and links to detailed information on the credential used and events associated with this credential.

5 Future Work

The work described here has demonstrated the proof of concept in auditing of proxy credential usage and its application in the DAMES project. The work is far from complete however. There are numerous avenues and case studies that

remain to be explored as part of the PCA project. However it is the case that the software has reached a stage where we can begin to deploy it at test NGS sites. A workshop is scheduled with the NGS technical support staff to demonstrate the solutions put forward in August 2010. Case studies also exist as part of the PCA project to demonstrate this software when used in an international context, e.g. in supporting access to and use of the NGS nodes, ScotGrid and international HPC facilities including TeraGrid in the US and D-Grid in Germany. Such international auditing efforts represent a key requirement in establishing global Grid infrastructures.

As the auditing work in PCA continues we expect to explore a variety of other research areas in auditing and usage of Grid facilities. In particular, once basic auditing capabilities exist, it will be possible to explore research avenues in the area of identifying irregular patterns of usage. Our focus here is on training algorithms to predict potentially suspicious proxy credential usage. We intend to apply Bayesian Neural Networks in this regard. However given the often sporadic access to and usage of Grid facilities by user communities, we expect that this in turn will be a challenge in itself.

Acknowledgement. The PCA work described here is funded by the Joint Information Systems Committee (JISC) in the UK. The DAMES project is funded by the Economic and Social Sciences Research Council (ESRC) in the UK. The authors gratefully acknowledge this support.

References

1. Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., Polk, W.: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (crl) Profile. RFC 5280 (Proposed Standard), Internet Engineering Task Force (2008)
2. Shibboleth Project, http://shibboleth.internet2.edu
3. Sinnott, R.O., Jiang, J., Watt, D.J., Ajayi, O.: Shibboleth-based Access to and Usage of Grid Resources. In: 7th IEEE/ACM International Conference on Grid Computing, pp. 136–143. IEEE Computer Society Press, Los Alamitos (2006)
4. Spence, D., Geddes, N., Jensen, J., Richards, A., Viljoen, M., Martin, A., Dovey, M., Norman, M., Tang, K., Trefethen, A., Wallom, D.: Shibgrid: Shibboleth Access for the UK National Grid Service. In: 2nd IEEE International Conference on e-Science and Grid Computing, p. 75. IEEE Computer Society Press, Los Alamitos (2006)
5. Wang, X.D., Jones, M., Jensen, J., Richards, A., Wallom, D., Ma, T., Frank, R., Spence, D., Young, S., Devereux, C., Geddes, N.: Shibboleth Access for Resources on the National Grid Service (sarongs). In: International Symposium on Information Assurance and Security, vol. 2, pp. 338–341 (2009)
6. Foster, I.: Globus Toolkit Version 4: Software for Service-Oriented Systems. In: Jin, H., Reed, D., Jiang, W. (eds.) NPC 2005. LNCS, vol. 3779, pp. 2–13. Springer, Heidelberg (2005)
7. Butler, R., Welch, V., Engert, D., Foster, I., Tuecke, S., Volmer, J., Kesselman, C.: A National-Scale Authentication Infrastructure. Computer 33(12), 60–66 (2000)

8. Staniford, S., Paxson, V., Weaver, N.: How to own the Internet in Your Spare Time. In: Proceedings of the USENIX Security Symposium (2002)
9. Tuecke, S., Welch, V., Engert, D., Pearlman, L., Thompson, M.: Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile. RFC 3820 (Proposed Standard), Internet Engineering Task Force, http://www.ietf.org/rfc/rfc3820.txt
10. Kunz, C., Szongott, C., Wiebelitz, J., Grimm, C.: Design and Implementation of a Grid Proxy Auditing Infrastructure. In: 5th IEEE International Conference on e-Science, pp. 11–18 (2009)
11. Szongott, C.: Websevice-based Auditing for Grid Proxy Credentials. Masters thesis. Gottfried Wilhelm Leibniz Universitat Hannover (2009)
12. DTrace Project, http://opensolaris.org/os/community/dtrace/
13. SystemTap Project, http://sourceware.org/systemtap/
14. Fonseca, R., Porter, G., Katz, R.H., Shenker, S., Stoica, I.: X-trace: A Pervasive Network Tracing Framework. In: 4th USENIX Symposium on Networked Systems Design and Implementation (2007)
15. Alfieri, R., Cecchini, R., Ciaschini, V., Frohner, A., Gianoli, A., Lrentey, K., Spataro, F., Firenze, I.: An Authorization System for Virtual Organizations. In: 1st European Across Grids Conference, pp. 13–14 (2003)
16. Nottingham, M., Sayre, R.: The Atom Syndication Format. Internet Engineering Task Force (2005), http://www.ietf.org/rfc/rfc4287.txt
17. MongoDB project, http://www.mongodb.org/
18. DAMES project, http://www.dames.org.uk
19. Lambert, P.S., Tan, K.L., Turner, K.J., Gayle, V., Prandy, K., Sinnott, R.O.: Utilising a Grid Enabled Occupational Data Environment. In: 16th World Congress of the International Sociological Association (2006)
20. McCafferty, S., Doherty, T., Sinnott, R.O., Watt, J.: Supporting Research into Depression, Self-harm and Suicide across Scotland. Journal of the Philosophical Transactions of the Royal Society, Series A, 3845–3858 (2010)
21. Watt, J., Sinnott, R.O., Doherty, T., Jiang, J.: Portal-based Access to Advanced Security Infrastructures. In: UK e-Science All Hands Meeting (2008)

Towards Detecting Influential Users in Social Networks

Amir Afrasiabi Rad and Morad Benyoucef

University of Ottawa, 55 Laurier Ave. East
Ottawa, Ontario, Canada K1N 6N5
a.afrasiabi@uOttawa.ca, benyoucef@Telfer.uOttawa.ca

Abstract. One of online social networks' best marketing strategies is viral advertisement. The influence of users on their friends can increase or decrease sales, so businesses are interested in finding influential people and encouraging them to create positive influence. Models and techniques have been proposed to facilitate finding influential people, however most fail to address common online social network problems such as fake friends, spammers and inactive users. We propose a method that uses interaction between social network users to detect the most influential among them. We calculate the relationship strength and influence by capturing the frequency of interactions between users. We tested our model in a simulated social network of 150 users. Results show that our model succeeds in excluding spammers and inactive users from the calculation and in handling fake friendships.

Keywords: Social network, Influence, Viral advertising, Word-of-mouth.

1 Introduction

In recent years, advances in internet technologies, security, and payment systems increased the importance and role of internet as a commercial tool and marketing channel. As a result, businesses increased their presence and activities on the internet in order to take advantage of a lower cost business channel, and attract more customers. On the other hand, emerge of web 2.0 technologies and introduction of advantageous internet tools such as blogs, wikis, instant communication, and social networks revolutionized the web collaboration structure which resulted in empowering and sophisticating both on-line businesses and customers. These new technologies and features have transformed the concept of web content contribution, and driven the web to be more social and interconnected.

The reason for popularity of Web 2.0 technologies revolves around user contribution and user content generation, so the users can easily present themselves on the online environment. Web 2.0 changed the concept of web contribution in a way that end users can become active web contributors, so the border between content providers as business owners and end users is diminishing. This is very important for online marketers, since increased user content generation created a new phenomenon called open innovation, which gives new opportunities to businesses so customers actively participate in the product design, customer service, and specially marketing and sale processes [1-3].

G. Babin, K. Stanoevska-Slabeva, P. Kropf (Eds.): MCETECH 2011, LNBIP 78, pp. 227–240, 2011.

Social network is one of the most successful Web 2.0 technologies that bases its grounds on user interactions and user content generation. Social networks have attracted more attention since they sit on the core of the web 2.0 technologies, and provide the main means for building interconnected web of users. A deep look into statistics shows that only in US social networks attracted more than 90% of all teenagers and young adults [4]. Facebook by itself has more than 400 million active users with the average of 130 friends, and users spend 22 billion minutes on Facebook every day. More than 60 million status updates are posted by users each day, and users upload more than 3 billion photos every month. Businesses are also active of Facebook, and created 1.5 million fan pages [5]. However, Facebook is only one of the hundreds of online social networks. All these facts were enough for Marketing Science Institute to announce "New Media" (i.e. social networking sites, blogs, mobile, and others) as its top research priority for 2008-2010 [6].

The importance of social networks for businesses revolves around their user base and the level of user activeness and contribution on content generation as an electronic social network is the main application that facilitates content distribution [1]. The generated contents and their ability to be widely distributed on the social networks turned them into a tool that shapes the behavior of users on the web. Thus, social networks provided users with the power to communicate their ideas in a broad range. This power that helps users to make other people agree with their opinion, or do what they want to do is referred as influence. The interconnectivity of social network allows that influence cascade through the whole social network. 52% of online social network users are influential, and statistics show that only 12% of users with negative influence could decrease the sales by 15% [7]. Hence businesses try to detect influential social network members in order to: (1) affect their influence and make it positive; (2) avoid spending their advertisement budget on those who are easily influenced by others; and (3) spread the word faster. To do so, businesses should be able to identify influential people. Then, businesses and social networks should create opportunities for users to benefit from product and service reviews provided by those influential people. As a result, customers benefit from better decisions, and businesses benefit from an increase in sales. Targeting a few influential consumers can trigger a cascade of influence throughout a social network or community [8, 9].

The question of who influences others in social networks has been engaging researchers for many years. Many techniques, in the areas of social science, mathematics, and computer science, have been devised to measure influence and identify influential users in social networks. Most techniques focus on connections between users. However, one fact is often forgotten, if there is no communication, there is no influence. In offline world, the connection usually means communication, but in online world, people are often connected without communicating with each other. In online social networks, communication (referred to as message, or interaction throughout this paper) can take a form of status updates, which functions similar to a broadcast message, commenting on activity, or direct message, and it is always associated with generating or re-generating new content. Therefore, communicative and active users, who are more influential than passive users, usually create more content, or distribute pre-generated content.

To our best knowledge, both categories of techniques that focus on connections or communication are rather successful in detecting influential people, but they still

show limitations in accuracy. The limitations are mainly observable in connection-based techniques, and their main limitation revolves around the specific issues that only exist in online social networks, i.e. fake friendship and spammers. These techniques also fail to differentiate active and passive users. Due to the explored limitations, we believe a more accurate methodology for identifying influential users in online social networks is needed. We developed a novel method to eliminate abovementioned limiting factors in measuring influence. Our model, despite formerly developed techniques, focuses on interactions between users rather than only relying on connections between users. Our method focuses on quantity of user interactions, omitting their type, or scent. Therefore, it is worth mentioning that our model identifies influential people no matter they create positive or negative influence.

The paper is organized as follows. Section 2 introduces different influence measurement techniques. Section 3 details our proposed model for detecting influential people. Section 4 is devoted to the implementation of our model. We conclude the paper in Section 5.

2 Influence Measurement Techniques

Different influence measurement models have been proposed in different academic sectors from social science, psychology, and marketing to computer science, and mathematics. In this section we review currently in use models.

Most existing social science and psychology studies on social influence adopt a survey approach since they believe that the decision to buy an item depends on user-specific behavioral characteristics [10], and surveys provide a comprehensive method for collecting user behavioral information considering all user-specific characteristics. A survey approach (self designation) to identify influential people, addresses the influential power of the participant himself and they include questions like (1) "Have you recently tried to convince anyone of your political ideas?" and (2) "Has anyone recently asked you for your advice on a political question?". A similar survey-based technique is the key informant method. This method, in spite of the self designation technique, questions about the influential power of friends. These methods provide a scale for categorizing people leading to the detection of the most influential individuals or groups among them. Although questionnaires may work well for small groups, for an online community with millions of members, surveys are problematic. Surveying a large group of people requires a long time for planning and processing. Moreover, reliability cannot be easily controlled in a large group of people. The third problem results from the dynamic nature of social networks. People change fast, and their networks change faster. Surveys are costly and cannot be done in close periods of time, so the survey results can only be valid for a short period of time. Finally, when survey questions address some personal qualities, the participants unwillingly over-describe or under-describe themselves depending on their personality [10].

Since the survey techniques are hardly to work for large scale social networks, such as those exist on the web, new techniques should be devised. Fortunately, in computer-mediated environments, users' online activities can be tracked and recorded, and some models have been developed to utilize user activity tracking feature. Activity duration method [7] infers site usage influence from secondary data on

member login activity. When users log in to the site, they mostly leave traces, so some facts about the level of activeness of the user can be inferred from that information. Users spending more time on social networks when they expect that there is likely to be new content to view, so a member's site usage level at each point is driven by his/her expectation about the volume and update frequency of relevant new content created by friends. In another word people spend more time on social network if there is more content to consume. This is still true when the user is a content generator, since more generated content by friends intensifies the competition for generating new content. Therefore, higher number of logins per day is taken to be a sign of higher usage.

Although this method as presented, acts on user logged in duration, it can be easily extended to other online activity measures such as the amount of time spent on the web site and number of messages sent. However, the problem of identifying influential users with site activity data turns out to be difficult because the data are typically sparse relative to the number of effects that need to be evaluated. Moreover, the number of logins does not necessarily reflect the influence. People spend more time on social networks when they are free while being busy tops people from more logins. Plus, as this method states the login pattern in a number of members should change to detect the influential, so in a cluster, where there are many interconnections with high number of shared friendship exists, it is difficult to determine who was the source of the influence.

Mathematical techniques, such as sociometric techniques, are the most popular among social network analysts [11, 12]. They are especially useful for analyzing online social networks since they can leverage large sets of customer data. They rely on network centrality scores to detect influence and influential people. They are more accurate than other techniques since they do not rely on answers from users, and the analysis can be repeated as changes happen in the network. Sociometric techniques rely on network measures such as the number of connections, networking purpose, demographics, and group membership [4, 13-22], although these measurements only reveal partial facts about a given social network. They use the social network graph to calculate influence based on the graph's centrality values.

There are two interpretations of social network graphs among social network analysts. Some envision the social networks as directed graphs, while others argue that they are undirected [23]. Most older sociometric techniques intended to measure influence in social networks focus on using undirected graphs as they see a social network as a two-way friendship environment [24], but newer techniques argue that influence in online social networks are mostly directed [25-27]. Not considering their approach about graph structure, sociometric techniques focus on node degree centralities by analyzing the network connections and friendships [28, 29]. But since they rely on social networks' static features, sociometric techniques miss users' dynamic activities. For this reason, they fail to differentiate between active and passive behaviour. In fact, they detect both content generators and content consumers as influential members. Moreover, because online social networks contain millions of members, their graphs would be very large, and crawling large, highly interconnected graphs is always costly [30]. With their inability to detect user activity, sociometric methods cannot differentiate fake from actual friendships. It is obvious that fake friends do not influence each other, but their connection value is calculated as a metric for influence.

Plus, spammers can (and usually do) engage with online social networks. They try to increase their influence by increasing their indegree, outdegree, and clustering coefficients, in an effort to fool existing models into thinking they are indeed influential.

3 Model for Interaction-Based Influence Measurement

To address the limitations of existing sociometric techniques, we propose a solution revolving around user interactions. Newman and Park [31] argue that online and offline social networks have different characteristics. They claim that in online social networks, interactions always have a direction (from initiator to receiver), which is not the case for offline social networks. That means that, in offline social networks, interactions (i.e., conversations) are mutually happening between two or more members as a sequence of send and receive activities. Since these activities are happening at the same time, they create a two-way influence on both (or all) participants in the conversation. Therefore, because the influence travels in both directions at an almost equal pace, the existing sociometric solutions were successful in determining influence in offline social networks. However, we believe that since influence in online social networks is happening because there is a message transferred from one member to another, the interaction feature should be part of the online social network analysis. Including interactions is important knowing that the initiated interaction may or may not be responded, which is rare in offline social networks.

To cover interactions in online social network analysis, we propose to generate a directed graph of interactions where arc direction is indicated by the direction of messages traveling on the social network. To do so, all interactions from every node are captured and integrated in the graph. Note that there can be several interactions between two nodes. Moreover, if there is a node in the social network that does not initiate any interaction, it will not have any outgoing edge, in the same way, if a node does not receive any messages, it will not have any incoming edge. It is worth mentioning that older interactions can be removed from the graph after a certain period of time since interactions, and consequently influences, usually (but not always) happen over a specific topic. Removing old interactions keeps the graph to a reasonable size.

To calculate the influence of social network users based on their interactions, we need a new set of metrics that can be used for analyzing directed graphs. The following section introduces these metrics.

3.1 Indegree and Outdegree Centrality

The indegree centrality factor correlates to the number of incoming interactions between node v and its neighbours. Indegree is the most basic metric for analysing a node in a network [32, 33]. High indegree can represent a lack of activity by the member or a tendency towards being a content consumer not a content generator. If the user with a high indegree value has a very low or zero outdegree value (defined later in this section) it is an indication of a complete inactivity or fake friendship. A higher number of received messages indicate a higher probability of adopting friends' behaviour, so the recent (i.e., not outdated) indegree value of nodes represent their

willingness to or tendency to get ideas from or be influenced by others. The indegree centrality of v is calculated by:

$$D_i(v) = \sum_{w \in S} \vec{e}(w,v) \ . \tag{1}$$

In the formula 1 and throughout this paper, w and v are graph node representatives, and e represents a directed graph edge. S represents a set of neighbours of v.

The outdegree factor correlates to the number of outgoing interactions between v and its neighbours. Outdegree represents the number of messages sent by a user in a certain period of time. It reflects the proclivity of a member to interact with his/her friends. A member with high outdegree has more opportunities to influence others by his behaviour because this kind of user creates more content in the social network.

The combination of outdegree and indegree centrality values helps in detecting spammers and inactive users. A user with zero indegree and high outdegree may be a spammer. Outdegree is calculated by:

$$D_o(v) = \sum_{w \in S} \vec{e}(v,w) \ . \tag{2}$$

3.2 Link Strength

Measuring link strength is probably one of the most important factors that should be considered in the analysis of social networks in order to find influential users. If the relationship between friends is strong, they can be influenced even with the lowest effort from the influencer. Therefore, users who have some close friends (i.e., characterized by strong friendship bonds) are more likely to influence or be influenced by them than those who have many friends but almost no close friends; provided that the two groups create comparable amounts of content.

We define the strength of a friendship as the average number of two-way interactions between a node and its adjacent nodes. For nodes w and v; Link Strength is positively related to the number of two-way interactions between w and v, so the probability that an individual influences a friend increases as a function of their link strength.

Considering friendship strength, we can add that users with low indegree and low outdegree may be considered fake friends because low indegree shows that the node either lacks intimate friends, or its friends are inactive. Note that since the outdegree value is low, the user is not considered a spammer. The link strength of w and v is calculated by:

$$R(w,v) = \frac{\left|\vec{e}(w,v)\right| + \left|\vec{e}(v,w)\right|}{D_o(w) + D_o(v)} \ . \tag{3}$$

3.3 Incoming and Outgoing Clustering Value

The clustering value is defined as the closeness of a node to a cluster of highly interconnected nodes [34, 35]. The incoming clustering value represents the number of

messages transferred to the node in question from its adjacent nodes that are also part of the cluster. A higher incoming clustering value means that the node is connected by more clusters, so it has more opportunities to be involved in and informed about different discussions, and consequently means a higher chance of adopting other members' behaviours. The incoming clustering value is calculated by:

$$C_i(v) = \frac{\sum_{w \in S} D_i(w)}{D_i(v) * \left(D_i(v) - 1\right)} \ . \tag{4}$$

The outgoing clustering value is defined as the number of messages transmitted from a user to adjacent clusters [34, 35]. As clusters are highly interrelated communities with a high level of interactions, they can cascade the user-generated content both inside and outside the cluster, so being related to a member of a cluster can increase the chances that your voice is distributed in the network. Therefore, a member with a higher outgoing clustering value has a higher chance of being influential. The outgoing clustering value is calculated by:

$$C_o(v) = \frac{\sum_{w \in S} D_o(w)}{D_o(v) * \left(D_o(v) - 1\right)} \ . \tag{5}$$

3.4 Calculating Influence

Considering the abovementioned values, the users with high outdegree and outgoing clustering value who have stronger relationships with others are influential. However, different factors do not affect influence equally as one factor may have a larger impact than the other. For instance, although proximity to clusters has a larger effect on the distribution of influence, if a user is not connected to any cluster but generates content, s/he is still influential to some extent. Therefore, each factor is weighted (α_1 and α_2) depending on its effects on influence. Adding weights to the calculation of influence contributes to its accuracy. However, we cannot estimate the values of these weights until we use the model on data from a real social network and validate the results with a survey of members of that network.

Note that since the graph is directional and spammers, who have no indegree value, are not crawled, they are automatically eliminated from the calculation. Moreover, in order to assign a zero value to all inactive nodes, we multiply the hyperbolic tangent of outgoing degree by the whole value. Therefore, if the outdegree is equal (or close) to zero, the final influence value will be equal (or close) to zero.

$$Influencer(v) = \lceil \tanh(D_o(v)) \rceil * \left(\alpha_1 D_o(v) + \alpha_2 C_o(v)\right) * \sum_{t \in S} R(w, v) \ . \tag{6}$$

The interaction-based method has one more advantage over similar sociometric measures as it can detect users with a high probability of being influenced by others. Since those users do not usually participate in the creation or transfer of influence, and follow opinion leaders in their decisions, marketers are interested in identifying

them in order to advertise to them trough the network of influencers rather than directly to them. Users with higher indegree value and intense friendships are considered to have a tendency to be influenced by others. However, those users should not be mistaken with inactive users. We eliminate users who have zero outdegree value, so the users who never generate any content (i.e., inactive users) are removed from our calculation. Therefore, the equation uses a hyperbolic tangent function to translate the outdegree value of v to either zero or one.

$$Influenced(v) = \lceil \tanh(D_o(v)) \rceil * (\alpha_3 D_i(v) + \alpha_4 C_i(v)) * \sum_{t \in S} R(w,v) \ . \quad (7)$$

4 Implementation

To test our model, we opted for the simulation methodology. For that we generated a random directed social graph of 150 nodes the density of which is visually presented in Figure 1. Since our model is based on interactions, there could be more than two links, in both directions, between any adjacent nodes. We based our directed graph generation on modified versions of random graph generation algorithms (available at http://www.yworks.com) suggested by Van Horn et al. [36], and we developed the simulation environment using Java and Java Universal Network/Graph Framework (JUNG) [37]. The random graph provides us with the opportunity to simulate situations close to the reality of social networks [38]. However, Newman et al. [38] argue that not all random graphs provide similar features to social networks as some nodes in the social networks have a skewed degree that makes them specific and different from random graphs.

The first algorithm generates a graph with a certain number of nodes. It then assigns a random degree to each node. In the next step, the node is connected with other nodes until a predefined number of edges are generated. This method creates a randomly distributed graph, but the generated graph differs from actual social network graphs. In the random graph, almost all nodes belong to one cluster which is not typically the case in social networks. The latter algorithm generates random trees, but they can easily be converted into graphs if we randomly change the direction of some edges, which will result in creation of clusters, and makes it similar to social networks. Therefore, we used the second algorithm as the main algorithm, and partially used the first one to increase the density of our graph and create nodes with special characteristics as can be seen in Figure 1. Plus, this algorithm distributes the degree in a way that a few skewed degree nodes will appear in the graph, which makes it similar to social networks. In our random graph, we embedded three classes of nodes, namely regular nodes, spammer nodes and inactive nodes. In Figure 1, we identified spammers and inactive users in rectangles and circles respectively. As mentioned earlier, the indegree of spammer nodes as well as the outdegree of inactive nodes are usually very low. Table 1 presents the specifications of our sample social graph. Please note that, we limited the number of specific nodes in our graph in order to develop a smaller graph to facilitate the presentation.

In order to gather the required information for our calculation, we need to traverse the graph from a starting point. We observed that since we are dealing with a directed

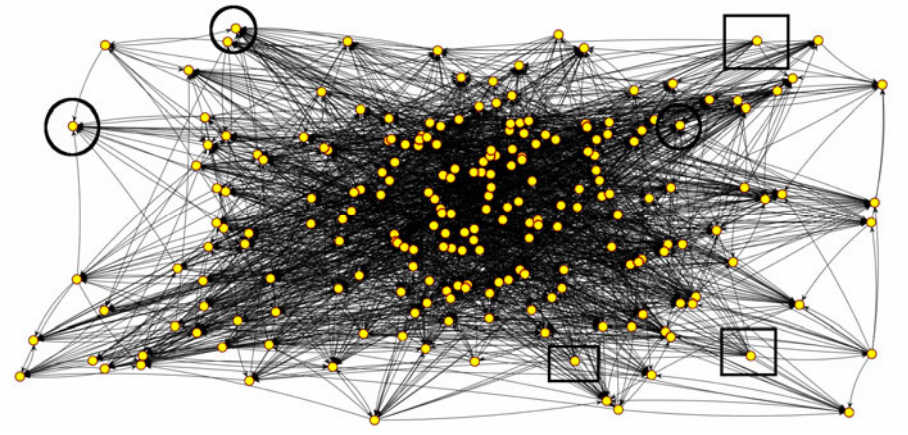


Fig. 1. Visual representation of our social graph

Table 1. Specifications of our Sample Social Graph

Indegree	Outdegree	Number of Nodes	Specification
>0	>0	144	Ordinary
>0	~0	3	Inactive
~0	>0	3	Spammer

graph, identifying the best starting point is essential for the efficiency of our technique. Because spammers and some inactive nodes do not have any incoming edge, they are never traversed and they are out of the calculation. As a general rule in our model, whenever the graph crawling algorithm encounters a node with zero incoming edges as a starting node, it should select a different starting node. Moreover, the larger the number of outgoing edges is, the better the node is as a starting point. Not selecting a node with zero incoming edges eliminates it from being traversed.

To traverse the graph, we use the breadth-first-search (BFS) algorithm since, according to the snowball method [39], it can traverse an acceptable portion of the graph instead of traversing all of the large graphs. Although our sample graph is small compared to an actual social network graph, we provide at least a sample node for each possible characteristic. The inaccuracies of the snowball sampling method do not affect our results given that the inclusion of clustering coefficient makes up for the moderate inaccuracy of the degree values.

As can be seen in Figure 2-a, we generated a moderately dense social graph of 50 users in order to elaborate on the role of clusters and specific nodes in our methodology. It is apparent in the figure that the most influential member has both outgoing and ingoing communication (interactions) with all three clusters in the network (clusters are circled in the figure). In other words, the influential node has strong connections with most or all clusters, and at the same time generates more content since it has many outgoing communications. The result of our analysis (Table 2) shows that if the connection-based sociometric solutions had been applied to our simulated graph,

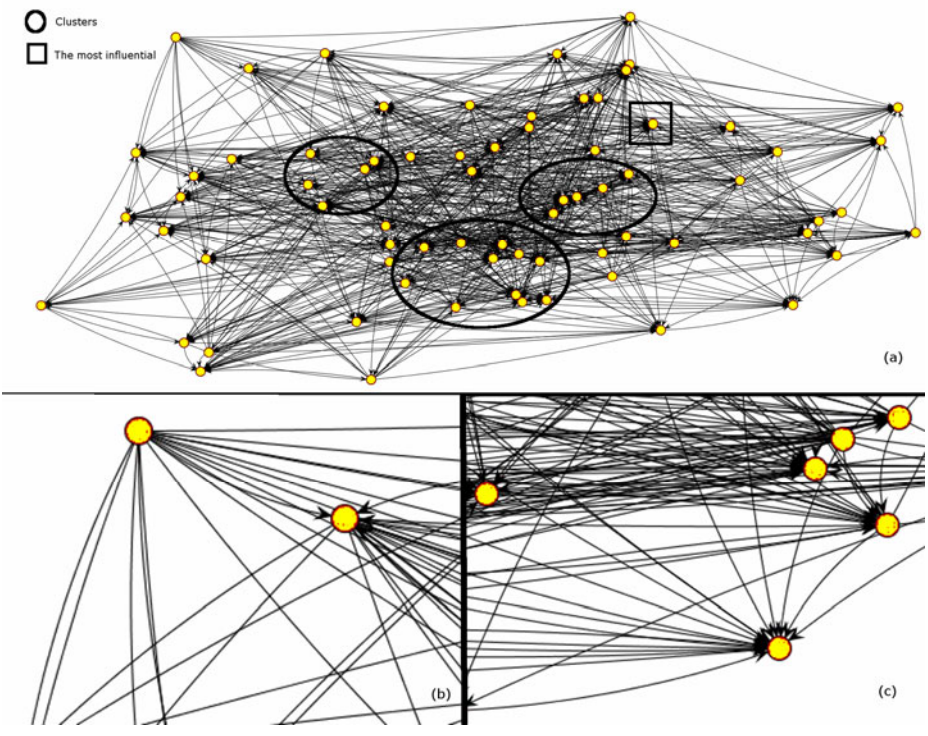


Fig. 2. Clusters and Specific Nodes

the spammer node as well as the inactive nodes would have been considered among the influencer nodes as their centrality values are larger than average. However, the interaction-based technique eliminates the possibility of those nodes being considered influential. As it appears in Figure 2-b, the spammer node does not have incoming communication, so its chance to be traversed in the graph is close to zero. On the other hand, the inactive node (Figure 2-c), does not generate any outgoing message, therefore its connection strength is too low to allow the influence value to grow. Our results are close to those obtained by applying the connection-based sociometric technique, but they are not exactly the same. The differences are the result of the inclusion of spammer nodes, inactive nodes, and fake friends in the calculation of the influence. In the case of a spammer node, it is considered to have a high influence in connection-based methods, but it is eliminated in our method. Plus, the most influential person (node) in our method comes third in the connection-based method because the strength of connections is not a factor, but the only important factor is the number of connections.

5 Conclusion

Social networks are completely reliant on their users and their correlations given that users are the main content generators. Since it differs in terms of frequency, volume,

type, and quality, not all user-generated content can derive traffic to the social network in the same way. Detecting users who generate attractive content for the social network paves the way for directing advertisement to the right users.

Accurately detecting social influence provides multiple benefits for online business such as the effectiveness of viral advertisement, and a better user involvement in product design (also called crowd-sourcing). Many techniques have been developed to detect influential social network users in social sciences, marketing, and recently mathematics and computer science research. Social science and marketing techniques, for instance, are more useful for the offline world and small groups while mathematical solutions better fit online social networks. Online social networks are different from offline social networks in a sense that the interactions are not always two-way, and users may create directed influence rather than the mutual influence which exists in offline social networks. Meanwhile, spammers as well as fake friends can only exist in online social networks, but sociometric methods for identifying influential people do not handle these issues effectively and may actually return inaccurate result.

In this paper we proposed an interaction based model to overcome the limitations of sociometric methods in dealing with special issues pertaining to online social networks. We generated a directed graph of users and interactions equivalent to an actual social network and calculated the influence of users based on their network's equivalent graph values. To calculate the influence, we measured each node's incoming and outgoing degrees in addition to our devised friendship strength metric. Based on our metrics (indegree, outdegree, clustering coefficient, and relationship strength), users who generate more content, are close to clusters, and have an average strong relationship with their friends are more influential. Our method provides a solution to the spammer and fake friendship problems and removes them from the list of influential people.

Table 2. Analysis Result on a Random Graph of a Social Network

Indegree	Outdegree	Incoming Clustering Value	Outgoing Clustering Value	Connection Strength	Influencer Value	Degree	Specification
57.57	57.58	1.57	1.55	0.17	231.73	117.31	Network Average
77	71	2.51	2.63	0.51	277.51	131	Highest
117	0	0.47	∞	0.06	~0	117	Inactive Average
0	117	∞	0.47	0.03	~0	117	Spammer Average

The applicability of our proposed technique is vividly visible in user buying behaviour is social commerce environments. Focusing on user decision making processes, the social commerce model consists of six stages namely need recognition, product brokerage, merchant brokerage, purchase decision, purchase, and evaluation [39]. Influence is an important part of almost all stages, even though it is not as important in the purchase decision and purchase phases. Identifying influential people can improve brand recognition. Users can also influence each other in purchasing different products. The generated positive influence affects the decisions in a way that a user

may be tempted or recognizes a need to purchase a product. This illustrates the effect of advertising products to influential people, so they influence their communities in purchasing those products.

There are some limitations associated with our model. The cost of detecting influential users increases exponentially by adding a highly connected node to the graph. Nevertheless, the model decreases the effect of high density by naturally excluding nodes that have low incoming edges. Moreover, since our model does not deal with any behavioural aspect, it is hard to say if the detected influential users create positive or negative influence. Moreover, as we mentioned earlier, more content generation implies higher influence, but redistribution of content is also important in cascading the influence. However, it is one of the limitations of our model that does not measure cascade of influence, by actions such as re-tweeting.

An evaluation of our proposed method is possible by applying the model to a real social network and at the same time surveying members of the social network to validate the result of the calculation. In our future work, we plan to do just that, plus we would like to perform statistical tests to improve the influence calculation equation. We would also like to integrate content analysis features in order to add behavioural analysis to our model while keeping the process cost low.

References

1. Huberman, B.A.: Crowdsourcing and Attention. Computer 41, 103–105 (2008)
2. Howe, J.: The Rise of Crowdsourcing, http://www.wired.com/wired/archive/14.06/crowds.html
3. Huberman, B.A., Romero, D.M., Wu, F.: Social networks that matter: Twitter under the microscope. 0812.1045 (2008)
4. Trusov, M., Bodapati, A.V., Bucklin, R.E.: Determining Influential Users in Internet Social Networks. SSRN eLibrary (2009)
5. Statistics | Facebook, http://www.facebook.com/press/info.php?statistics
6. MSI - 2008-2010 Research Priorities, http://www.msi.org/research/index.cfm?id=43
7. Iyengar, R., Han, S., Gupta, S.: Do Friends Influence Purchases in a Social Network? SSRN eLibrary (2009)
8. Kempe, D., Kleinberg, J., Tardos, É.: Maximizing the spread of influence through a social network. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 137–146. ACM, Washington (2003)
9. Provost, S.H.F., Volinsky, C.: Network-Based Marketing: Identifying Likely Adopters via Consumer Networks. Statistical Science 21, 256–276 (2006)
10. Rogers, E., Cartano, D.: Methods of Measuring Opinion Leadership. The Public Opinion Quarterly 26, 441, 435 (1962)
11. Valente, T.W., Hoffman, B.R., Ritt-Olson, A., Lichtman, K., Johnson, C.A.: Effects of a Social-Network Method for Group Assignment Strategies on Peer-Led Tobacco Prevention Programs in Schools. Am J. Public Health 93, 1837–1843 (2003)
12. Coleman, J.S., Katz, E., Menzel, H.: Medical Innovation: A Diffusion Study. Bobbs-Merrill Co (1966)
13. Ariely, D., Levav, J.: Sequential Choice in Group Settings: Taking the Road Less Traveled and Less Enjoyed. Journal of Consumer Research 27, 279–290 (2000)

14. Tantipathananandh, C., Berger-Wolf, T., Kempe, D.: A framework for community identification in dynamic social networks. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 717–726. ACM, San Jose (2007)
15. Faloutsos, C., McCurley, K.S., Tomkins, A.: Fast discovery of connection subgraphs. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 118–127. ACM, Seattle (2004)
16. Evans, D.C.: Beyond Influencers: Social Network Properties and Viral Marketing. Psychster Inc (2009)
17. Singh, S.: Social Networks And Group Formation - Boxes and Arrows: The design behind the design, http://www.boxesandarrows.com/view/social-networks
18. Preece, J., Maloney-Krichmar, D.: Online communities: focusing on sociability and usability. In: The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications, pp. 596-620L. Erlbaum Associates Inc., Mahwah (2003)
19. Preece, J.: Sociability and usability in online communities: determining and measuring success. Behaviour & Information Technology 20, 347 (2001)
20. Kim, Y.A., Srivastava, J.: Impact of social influence in e-commerce decision making. In: Proceedings of the Ninth International Conference on Electronic Commerce, pp. 293–302. ACM, Minneapolis (2007)
21. Social network influence or similar demographics driving product adoption-Part 1 » Improving profits using network analysis, http://www.sonamine.com/home/index.php?option=com_wordpress&p=332&Itemid=70
22. Liu, H.: Social Network Profiles as Taste Performances. Journal of Computer-Mediated Communication 13, 252–275 (2007)
23. Wasserman, S., Faust, K.: Social network analysis: methods and applications. Cambridge University Press, Cambridge (1995)
24. Tichy, N.M., Tushman, M.L., Fombrun, C.: Social Network Analysis for Organizations. The Academy of Management Review 4, 507–519 (1979)
25. Freeman, L.C.: Centrality in social networks conceptual clarification. Social Networks 1, 215–239 (1978)
26. Newman, M.E.J.: Random graph models of social networks. Proceedings of the National Academy of Sciences 99, 2566–2572 (2002)
27. Kimura, M., Saito, K., Nakano, R.: Extracting influential nodes for information diffusion on a social network. In: Proceedings of the 22nd National Conference on Artificial Intelligence, vol. 2, pp. 1371–1376. AAAI Press, Vancouver (2007)
28. Katona, Z., Sarvary, M.: Network Formation and the Structure of the Commercial World Wide Web. MARKETING SCIENCE 27, 764–778 (2008)
29. Katona, Z., Zubcsek, P.P., Sarvary, M.: Network Effects and Personal Influences: Diffusion of an Online Social Network. Haas School of Business, UC Berkeley (2009)
30. Mislove, A., Marcon, M., Gummadi, K.P., Druschel, P., Bhattacharjee, B.: Measurement and analysis of online social networks. In: Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement, pp. 29–42. ACM, San Diego (2007)
31. Newman, M.E.J., Park, J.: Why social networks are different from other types of networks. Physical Review E 68 (2003)
32. Wasserman, S., Faust, K.: Social Network Analysis: Methods and Applications. Cambridge University Press, Cambridge (1994)
33. Scott, J.: Social Network Analysis. Sociology 22, 109–127 (1988)

34. Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world' networks. Nature 393, 440–442 (1998)
35. Zhou, H.: Scaling exponents and clustering coefficients of a growing random network. Physical Review E 66 (2002)
36. Van Horn, M., Richter, A., Lopez, D.: A random graph generator. In: Presented at the 36th Annual Midwest Instruction and Computing Symposium, Duluth, MN,
37. JUNG - Java Universal Network/Graph Framework, http://jung.sourceforge.net/
38. Newman, M., Watts, D., Strogatz, S.: Random graph models of social networks. Proceedings of the National Academy of Sciences of the United States of America 99, 2572, 2566 (2002)
39. Lee, S., Kim, P., Jeong, H.: Statistical properties of sampled networks. Physical Review E 73 (2006)

Intelligent Monitoring System for Online Listing and Auctioning

Farid Seifi and Mohammad Rastgoo

School of Information Technology and Engineering
University of Ottawa
Ottawa Ontario Canada, K1N 6N5
{fseif050,mrast074}@uottawa.ca

Abstract. As the online auctioning sites grew, it became necessary to restrict or forbid auctions for various items. For this purpose, online auctioning companies assign special personnel, a large team of monitoring experts, to monitor the items posted on the web to ensure a safe and healthy online trading atmosphere. This process costs a lot for such companies and also takes a lot of time. In this research we propose a solution to this problem as an automated intelligent monitoring system which uses machine learning and data mining algorithms, in particular document classification, to monitor new items. Our results show that this approach is reliable and it reduces the monitoring cost and time.

Keywords: online auctioning and listing, monitoring, Machine Learning, document classification, Naive Bayes classifier.

1 Introduction

"eBay Inc. is an American Internet company that manages eBay.com, an online auction and shopping website in which people and businesses buy and sell a broad variety of goods and services worldwide. In its earliest days, eBay was essentially unregulated. However, as the site grew, it became necessary to restrict or forbid auctions for various items." [6] For this purpose, online auctioning and listing companies assign special personnel, a large team of monitoring experts, to monitor the online transactions to ensure a safe and healthy online trading atmosphere. This process costs a lot for such companies and also is time consuming. Due to these problems some companies do not monitor all items posted on the web and just monitor some items randomly.

These problems motivated us to use decision-making and machine learning algorithms to monitor new listings and auctions using special keywords which might be used together in most illegal transactions. A wide range of such illegal keywords and items are available at 'eBay prohibited and restricted items'[4] and also 'offensive material policies'[3]. Those keywords can be used to maintain and update a decision-making algorithm. In addition to above mentioned keywords, we have proposed to aggregate sufficient amount of advertises and auctions as

G. Babin, K. Stanoevska-Slabeva, P. Kropf (Eds.): MCETECH 2011, LNBIP 78, pp. 241–252, 2011.

training data, classified as *legal* or *illegal*, along with text classification learning methods to classify and diagnose future transactions.

A very strict cost-sensitive classifier[11] is used to find items which are even a little bit likely to be illegal, in order to avoid from slipping any illegal item. The number of items classified as *illegal* is much smaller than the total number of items, consequently, a small group of monitoring experts could be hired to revise the classified items as *illegal*. This solution does not only reduce the cost of hiring monitoring personnel but also it reduces the monitoring time and it can increase its accuracy.

2 Motivation

The amount of transactions in online auctioning and listing websites has increased to a large number in recent years. Considering the fact that both *legal* and *illegal* proceedings fall among these transactions, *illegal* postings though with low occurrence rate, have great impact on the marketplace. This will account for all the listed items by pranksters or unlawful traders. Consequently the marketplace will suffer from a considerable risk. So it is clear that there is a strong need for a monitoring system for such websites to restrict or forbid illegal auctions and listings posted online. For such purpose, these companies hire a large team of monitoring experts. But, since there are a huge number of auctions and listings, the required monitoring staff is more than it is expected and consequently, this monitoring strategy would cost a lot. Noteworthy, an eBay spokesman said (at Feb 09 2009)[2]:

"The company invests more than £6 million every year in developing the best technology possible to prevent anything from slipping through the net. Safety is our number one priority and we recognize we need to do more to protect our members. We need sophisticated technology to help us identify high risk or illegal items."

In this study, we propose an automated intelligent monitoring system which uses machine learning and decision making algorithms to facilitate the monitoring process.

2.1 History of Illegal Auctions

If we look at the history of illegal auctions and listings posted on websites we will find out there are a lot of well-known bothering stories about illegal auctions which became very famous and even appeared on news headlines. As an example Reuters[5] said 'Online Bidders Offer Millions For Human Kidney'. In another story Associated Press[1] mentioned that 'eBay takes down offers of babies for sale'. In these stories pranksters offered up two human kidneys and three infant children to the highest bidders. The bogus (and illegal) items were quickly deleted, but not before fetching bids of $5.7 million for a kidney and $109,100 for a yet-to-be-born male child. These were by no means the first incidents of their kind, but for some reason they attracted worldwide media attention and sent eBay executives scrambling to do damage control.

3 Intelligent Monitoring System

The problem which is defined before and also the related works which is done to solve a similar problem, spam filtering, motivated us to use decision making and machine learning algorithms to monitor new auctions and postings using special keywords which will be used together in most illegal transactions. So we use document classification in order to classify new auctions and postings to two classes of *legal* and *illegal*. We use a strict cost-sensitive classifier which will classify all risky items to the class of *illegal*. Then we send those items which are classified as *illegal* to a small team of monitoring experts which will revise the decision made by machine learning algorithms.

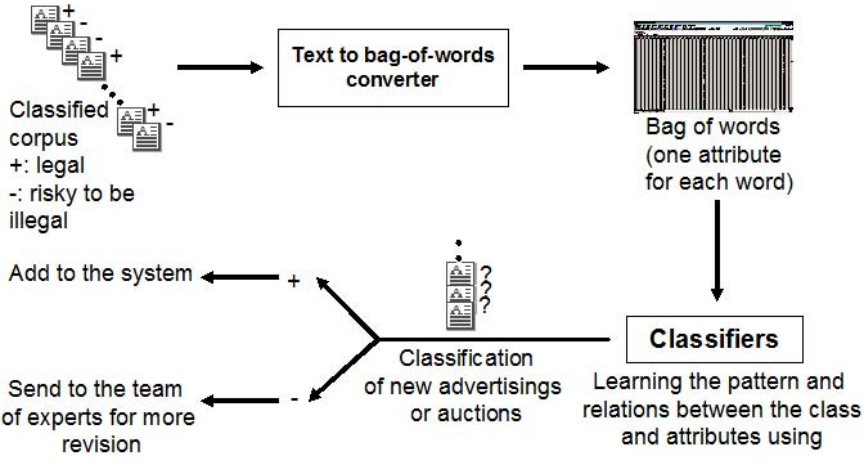


Fig. 1. Proposed model for intelligent monitoring system of online listing and auctioning

3.1 Data Gathering and Generation

The ideal data is the data gathered and classified, as *legal* or *illegal*, by the team of monitoring experts. But we only have access to the legal items posted on online auctioning and listing websites, since they do not post items classified as *illegal* on the web. To simulate the process we use the keywords provided by eBay as prohibited and restricted item categories and also available history of illegal auctions in news headlines and then we generate some illegal samples. Some of these categories are as follows:

Adult only category, human body tissues, alcohol, animals and wildlife products, art, drugs and drug paraphernalia, firearms, weapons, and knives, government documents, IDs, and licenses.

So the data gathering and generation of our model in order to simulate the problem and proposed solution is a four step process as following:

• **Step1:** extracting a number of available auctions on eBay as legal items.
• **Step2:** converting those HTML files to plain text files using HTML to text converters by removing html tags and also removing unrelated text.
• **Step3:** generating a number of text documents as illegal items using prohibited and restricted item categories and history of such auctions.
• **Step4:** extracting bag-of-Words from the text documents.

After this four step process our bag-of-words contains instances belonging to both classes of *legal* and *illegal* and is ready to feed a classification algorithm.

3.2 Bag-of-Words Model

As we see in figure 1 bag-of-words model is used in our proposed model to feed a classifier. In text classification we have to do some pre-processing on the input data to make it ready to feed a classifier. In a bag-of-words, a text (such as a sentence or a document) is represented as an unordered collection of words, disregarding grammar and even word order. There are three different types of bag of words which can provide us with different type and amount of information.

• **Binary bag of words:**
This type of bag of words only includes information about the words which appeared in each text document.
• **Frequency based bag of words:**
This type of bag of words indicates the number of appearance of each word in each text document.
• **Tf-idf (term frequency- inverse document frequency) bag of words:**
In this type, the importance of a word in a text document increases proportionally to the number of times the word appears in the document but offsets by the frequency of the word in the corpus (in document classification, we call all the text documents that we have in a problem, the corpus).

We use tf-idf bag of words in all our experiments.

3.3 Naive Bayes Classifier

A classifier should be used as part of our model. This is a document classification task and some classification methods are much more efficient in such applications. The most common and simple classification method which is used for document classification is Naive Bayes[13]. We consider an assumption in this classification method. In simple terms, a Naive Bayes classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature. In our problem this means that the appearance of a certain word in a text document would be considered to be unrelated to the appearance of any other words in that text document. For example, a fruit may be considered to be an apple if it is red, round, and about 4" in diameter. Even if these features depend on each other or upon the existence of the other features, a Naive Bayes classifier considers all of these properties to independently contribute to the probability that this fruit is an apple.

The reason why we are interested in classification tasks, is to compute the posterior probability for any value of the class given a new unclassified data sample. Considering F in the following formulas as feature, the posterior probability is $p(Class|F_1, F_2, F_3, ..., F_n)$. In our problem the class can have two values, *legal* and *illegal*, and the features are the words that we have in the dictionary of our corpus and based on the type of bag of words model used they could have different values for any given text document. So if we can compute the probability of *legal* and also *illegal* class for a given sample, as a feature vector, then we can compare these two probabilities and decide to classify the given sample into the most probable class. So now the problem is to compute the posterior probability for the given data sample using the bag-of-words model extracted from corpus.

We can reformulate the posterior probability as following based on the Bayes rule:

$$p(Class|F_1, F_2, F_3, ..., F_n) = \frac{p(Class)p(F_1, F_2, F_3, ..., F_n|Class)}{p(F_1, F_2, F_3, ..., F_n)} \quad (1)$$

which different parts of this formula are as following:

$$Posterior = \frac{Prior \times Likelihood}{Evidence} \quad (2)$$

The denominator of this formula is not dependent on the class value so when we want to compute the posterior probability for different values of the class we don't need to compute the evidence in denominator. So we can simply compute the numerator of this formula. The numerator is actually the joint probability of the class and features, $p(Class, F_1, F_2, F_3, ..., F_n)$. We use the following Naive conditional independence assumption between features so we can extend this joint probability of the class and attributes.

$$p(F_i|Class, F_j) = p(F_i|Class) \quad (3)$$

This assumption considers that the appearance of a value in a feature is not dependent on the values of other features in a given sample. Considering this assumption we can reformulate the joint probability as following:

$$p(Class, F_1, F_2, F_3, ..., F_n) = p(Class)p(F_1|Class)p(F_2|Class)p(F_3|Class)... \quad (4)$$

Computing different parts of this formula given the bag-of-words is very easy. Simply, for each class we can count the frequency of a given feature in documents of that class and normalize it by the number of all documents of that class to compute the probability of that feature given that certain class. The probability of the class is also easy to compute by dividing the number of documents of that class to the number of all documents. Computing these marginal probabilities and priors, we can use Naive Bayes classifier to compute the posterior probability and then do the inference and classify the new text documents.

Naive Bayes classifier is a probabilistic classifier. It means that, for a new given sample, it provides us with the probability of belonging to any class. So we can compare these probabilities and decide to classify the given sample to the most probable class. Since, in this problem, it is very important for us not to classify an illegal sample as *legal*, we can use a strict strategy and instead of just finding the most probable class we can use some threshold or weights[17] to do the inference. In that case we would classify any risky sample to *illegal* class and then we need to revise the decision made by the algorithm, using monitoring experts. But since the number of samples classified as *illegal* is small, many samples are classified as *legal*, a very small team of monitoring experts can handle this task. Therefore using this method we assert that we can reduce the cost and time of monitoring items with a very good performance. The result of our experiments using this model is shown in experimental results section.

4 Experimental Results

4.1 Data Set

As mentioned before we don't have access to a complete training data set for this problem because the items are posted on auctioning and listing websites are all those which are classified as *legal*. So in order to simulate the data set of this problem we saved 131 auctions from eBay as legal samples. The saved HTML files are converted to text using HTML to text converters and the text which is related to the items extracted. Then we created 17 samples as illegal items using restricted and prohibited items and categories provided by eBay. The number of samples may be considered to be very small but it makes the problem much more challenging; that is it would be harder for machine learning algorithms to extract the patterns of the illegal items using a small training data set. More specifically, in machine learning it is considered, as a fact, that learning algorithms will work better if we inject more training data in learning process. So if the proposed solution for a learning problem works in a satisfactory fashion with a small training data set it will clearly work better with more data. Therefore, using a small training data set makes the simulation easier, but more challenging.

We extracted the tf-idf bag-of-words for the corpus using WEKA. The dictionary of the bag of words contains 1225 words. Then we used WEKA to run different classifiers on the data set. WEKA is introduced in next subsection.

4.2 Introduction to WEKA[12]

WEKA, stands for Waikato Environment for Knowledge Analysis, is a widely used tool for data mining research. WEKA originates in New Zealand and the project has been officially initiated in late 1992.The WEKA project provides a set of machine learning algorithms and data preprocessing tools to process and compare different machine learning methods on a given data set. In short,

WEKA offers algorithms for regression, classification, clustering, association rule mining and selection of attributes.

WEKA has an extensible modular architecture. Not only it provides access through an API to help integrating new algorithms to the system but also it allows to more complex data mining processes to be created out of the base algorithms. This can be achieved easily by using the graphical user interface built-in the system. WEKA has support for external data sources featuring files, URLs and databases.

WEKA is very well adopted due to its built-in support for learning schemes and algorithms such as: Bayesian logistic regression, Best-first decision tree, Decision table Naive Bayes hybrid, Discriminative multinomial Naive Bayes, Functional trees, Gaussian processes, and etc.

Moreover WEKA benefits the capability to allow meta algorithms to be used along with the base learning algorithms to widen applicability or increase performance.

4.3 Discussion

The data set we are working on is an imbalanced dataset, since the number of legal samples is much more than the number of illegal samples. In such data sets, the accuracy could not be considered as a good evaluation measure. Consider that we have an imbalanced data set which 99% of its samples belong to the positive class. If the classifier classifies all the samples to the positive class the accuracy would be 99% which seems to be good but in reality it is not a good classifier because it is classifying all negative samples to the positive class. In such problems, we need to use other evaluation methods which can give us more information about the results, based on one class. Precision, recall and f-measure are evaluation measures that we can use to obtain information about the performance of the classifier with regard to only one class.

Precision for a class is the number of true positives (i.e. the number of items correctly labeled as belonging to the positive class) divided by the total number of elements labeled as belonging to the positive class (i.e. the sum of true positives and false positives, which are items incorrectly labeled as belonging to the class).

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{5}$$

Recall in this context is defined as the number of true positives divided by the total number of elements that actually belong to the positive class (i.e. the sum of true positives and false negatives, which are items which were not labeled as belonging to the positive class but should have been).

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{6}$$

The weighted harmonic mean of precision and recall, the traditional F-measure or balanced F-score is:

$$F = 2.\frac{Precision.Recall}{Precision + Recall} \tag{7}$$

In a classification task, a Precision score of 1.0 for a class C means that every item labeled as belonging to class C does indeed belong to class C (but says nothing about the number of items from class C that were not labeled correctly) whereas a recall of 1.0 means that every item from class C was labeled as belonging to class C (but says nothing about how many other items were incorrectly also labeled as belonging to class C). Often, there is an inverse relationship between precision and recall, where it is possible to increase one at the cost of reducing the other. In our problem we are interested to achieve a recall of 1.0 for *illegal* class because in that case any illegal sample will be classified as *illegal*. If we increase the recall, precision will decrease and that means that the number of legal samples classified as *illegal* will increase. We can accept these misclassifications because we can simply send the small number of items classified as *illegal* to a small team of monitoring experts for revision.

In order to increase the recall we can use cost-sensitive classification. In our problem the cost of classifying an illegal item to a *legal* class is very high. In cost-sensitive classification a cost table would be applied on computed posterior probabilities obtained for different values of the class before inference.

So we used cost-sensitive classification with Naive Bayes as the classifier[9] to reduce the cost and we set the cost of classifying an illegal item as a legal one to a high value. We compare different evaluation measures for simple Naive Bayes and cost-sensitive Naive Bayes classifier.

4.4 Results

We used WEKA to run simple Naive Bayes and cost-sensitive Naive Bayes with 10-fold cross-validation. We compared the results of these two classifiers to show what is the effect of each classifier on different evaluation measures.

Table 1. Evaluation measures using Naive Bayes

TP rate	FP rate	Precision	Recall	F-Measure	ROC Area	Class
0.992	0.176	0.978	0.992	0.985	0.989	legal
0.824	0.008	0.933	0.824	0.875	0.989	illegal

* TP- True Positive
FP- False Positive

Table 1 shows the results of Naive Bayes classifier and values of different evaluation measures.

The confusion matrix is also shown in table 2. The accuracy of the model is 97.3154% which seems not to be bad but as mentioned before our data set is

Table 2. Confusion matrix using Naive Bayes

a	b	<− classified as
131	1	**a= legal**
3	14	**b= illegal**

imbalanced and also the cost of misclassifying an illegal item is higher than the cost of misclassifying a legal item.

As we see in confusion matrix, 3 illegal items out of 17, in our small simulation, are misclassified which is unacceptable in our problem. Recall for the *illegal* class is an evaluation measure which shows this lack of performance of the Naive Bayes method. Its value is 0.824 which is not good for a class with a high cost. The best value for this measure is 1.0. So we used cost-sensitive classification with Naive Bayes, using the cost table given in table 3, to optimize this measure.

Table 3. Cost matrix

illegal	legal	<− classified as
0	1	**legal**
5000000	0	**illegal**

The accuracy of the cost-sensitive classifier is 89.9329% which is lower than normal Naive Bayes. But as we see in table 4, the recall for *illegal* class is 1.0 and also as we see in table 5 no illegal item is misclassified. The accuracy is lower because in this classifier more legal items are misclassified but it is more acceptable and to solve this problem we can assign a small team of monitoring experts to revise the items which are classified as *illegal*.

Table 4. Evaluation measures using cost-effective classification with Naive Bayes

TP rate	FP rate	Precision	Recall	F-Measure	ROC Area	Class
0.886	0	1	0.886	0.94	0.943	legal
1	0.114	0.531	1	0.694	0.943	illegal

* TP- True Positive.
FP- False Positive.

Based on these results we assert that this approach could be used to monitor new online auctions and postings together with a very small team of monitoring experts. This approach reduces the cost of hiring monitoring experts and decreases the monitoring time.

Table 5. Confusion matrix for cost-effective classification using Naive Bayes as classifier

a	b	<− classified as
117	15	**a= legal**
0	17	**b= illegal**

5 Related Works

The proposed problem is very close to the problem of spam filtering. Researchers have been working to solve this problem for more than a decade. Many significant solutions have been published for this problem yet including [7,15,14,10]. In spam filtering, mail servers use a text classification algorithm to classify new e-mails as *spam* or *legitimate*. In learning process, text classifiers consider each e-mail message as an unordered collection of words (bag of words) defined as *spam* or *legitimate*. Each word is an attribute and attribute values are number of appearance of words in email. *spam* and *legitimate* are class values. A classification algorithm, i. e. Naive Bayes, learns the relation between attribute values and class and then uses the extracted pattern for classification and prediction of new e-mails. This problem is very close to the problem of online auction and list monitoring. In our problem, similar to the problem of spam filtering, each sample contains some text as title and description of the item, which we could consider as a sequence of unordered words and then we can deal with them just like what they do in spam filtering.

In addition to spam filtering, another well-known application of document classification can be observed in search engines. As an instance, Google uses document classification in order to index documents and then using relevancy scores, also known as Page Rank, Google ranks web pages according to links pointing from one page with relevant content to another. In other words, Page Rank mechanism takes into account the content relevancy when assigning a weight to the link pointed from page A to page B [8,**?**].

Since the measurement of a page's popularity can be abused by creating multiple links to another page, Google has set Page Rank mechanism in place to avoid improper ranking of the links in between irrelevant web pages. It not only covers pages marked as important in the database, which were parsed and classified using a document classification technique [16], but also the system benefits from a text-matching algorithm to better identify web pages with relevant keyword content.

6 Conclusion and Future Work

In this study we proposed a solution to the problem of monitoring online auctions and listings which is a very costly process for online companies. We used machine learning approaches and particularly document classification to classify

new auctions or postings as *legal* or *illegal*. Our experiments show that using this approach, we can reduce the number of monitoring experts and this will decrease the monitoring cost and also this approach will reduce the monitoring time.

This approach is tested on a very small training set obtained from real auctions and manipulated to include illegal items. The small size of the data set makes it a more challenging problem but using a real large data set may give us much more reliable results. Testing this approach using a large real data set would be considered as a future work of this study.

We've used text as our auction data to feed our machine learning algorithm. In order to obtain more accurate classification, as a future work, we would suggest using regression methods to draw a trend of how auctions ended on items' sell price. The regression will be adjusted to present time/price data whereas the data feed will consist of new auction data, the auction ending price, to be measured against the item price trend and to have the outliers captured as flagged/illegal listings to be controlled by human experts.

Using different cost matrixes and comparing the result of changing the cost and investigating the effect of the cost defined for misclassification of illegal items, would also be considered as a future work.

Acknowledgments. I would like to thank my supervisor, Chris Drummond, for his kind suggestions and support in this study.

References

1. The Associated Press, http://www.ap.org
2. eBay Shenanigans, http://urbanlegends.about.com/library/weekly/aa090899.htm
3. Offensive Material Policy, http://pages.ebay.com/help/policies/offensive.html
4. Prohibited and Restricted Items, http://pages.ebay.ca/help/policies/items-ov.html
5. Reuters, http://www.reuters.com
6. Wikipedia, the free encyclopedia, http://www.wikipedia.org
7. Androutsopoulos, I., Koutsias, J., Chandrinos, K., Paliouras, G., Spyropoulos, C.: An Evaluation of Naive Bayesian Anti-Spam Filtering. In: Proc. of the Workshop on Machine Learning in the New Information Age, pp. 9–17 (2000)
8. Berin, S., Page, L.: The anatomy of a large-scale hypertextual Web search engine. In: Proceedings of the Seventh International Conference on World Wide Web (WWW 2007), pp. 107–117. Elsevier Science Publishers B. V., Brisbane (1998)
9. Chai, X., Deng, L., Yang, Q., Ling, C.X.: Test-Cost Sensitive Naive Bayes Classification. In: ICDM 2004: Proceedings of the Fourth IEEE International Conference on Data Mining, pp. 51–58. IEEE Computer Society, Washington (2004)
10. Cormack, G., Lynam, T.: A Study of Supervised Spam Detection applied to Eight Months of Personal E-Mail. Technical report, ACM Transactions on Information Systems (2004)
11. Elkan, C.: The Foundations of Cost-Sensitive Learning. In: 17th International Joint Conference on Artificial Intelligence, pp. 973–978 (2001)

12. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. SIGKDD Explor. Newsl. ACM 11(1), 10–18 (2009)
13. Mitchell, T.M.: The Grid: Machine learning. McGraw Hill, New York (1997)
14. Gary, R.: A statistical approach to the spam problem. J. Linux. Specialized Systems Consultants, Inc. 2003(107) (March 2003)
15. Sahami, M., Dumais, S., Heckerman, D., Horvitz, E.: A Bayesian Approach to Filtering Junk E-Mail. AAAI Technical Report WS-98-05, Learning for Text Categorization: Papers from the 1998 Workshop (1998)
16. Silva, S., Rodrigues, P., Albuquerque, A.: Document Classification. Technical report, Instituto Superior Tecnico, Universidade Tecnica de Lisboa (2008)
17. Zhao, H.: Instance weighting versus threshold adjusting for cost-sensitive classification. J. Knowl. Inf. Syst. 15(3), 321–334 (2008)

Gossip-Based Networking for Internet-Scale Distributed Systems

Etienne Rivière[1] and Spyros Voulgaris[2]

[1] Computer Science Department, Université de Neuchâtel, Switzerland
[2] Vrije Universiteit Amsterdam, The Netherlands
etienne.riviere@unine.ch, spyros@cs.vu.nl

Abstract. In the era of Internet-scale applications, an increasing number of services are distributed over pools of thousands to millions of networked computers. Along with the obvious advantages in performance and capacity, such a massive scale comes also with challenges. Continuous changes in the system become the norm rather than the exception, either because of inevitable hardware failures or merely due to standard maintenance and upgrading procedures. Rather than trying to impose rigid control on the massive pools of resources, we should equip Internet-scale applications with enough flexibility to work around inevitable faults. In that front, gossiping protocols have emerged as a promising component due to their highly desirable properties: self-healing, self-organizing, symmetric, immensely scalable, and simple.

Through visiting a representative set of fundamental gossiping protocols, this paper provides insight on the principles that govern their behavior. By focusing on the rationale and incentives behind gossiping protocols, we introduce the reader to the alternative way of managing massive scale systems through gossiping, and we intrigue her or his interest to delve deeper into the subject by providing an extensive list of pointers.

1 Introduction

With the advent of worldwide networks and the Internet, computer systems have been going through an unprecedented shift in scale and complexity. Services that are distributed on thousands, if not millions, of machines, are gradually becoming commonplace.

Peer-to-peer systems are a well known example of massively distributed services. They employ end-user computers, often in the order of thousands or even millions. Each node acts both as a client of the provided service and at the same time as a server, collaborating with other peers to provide this same service. Examples of this first class of massive-scale systems include file sharing networks [1,2,3], collaborative search engines [4], multicast systems [5,6], publish/subscribe [7,8], etc.

A second area in which scale has grown from large to massive is that of *data centers*, providing services in a more traditional client-server fashion. For the major companies providing Internet services, this is largely due to the need

G. Babin, K. Stanoevska-Slabeva, P. Kropf (Eds.): MCETECH 2011, LNBIP 78, pp. 253–284, 2011.

to serve more and more complex applications to an ever-increasing number of clients. Note that the shift in scale does not concern only large companies. The advent of data center externalization and virtualization offered by the *cloud computing* paradigm allows short- and medium-scale companies to benefit from potentially very large infrastructures. In the following section we describe the important characteristics of large-scale distributed systems.

1.1 Challenges in Massive-Scale Distributed Systems

Both peer-to-peer systems and large data centers constitute large-scale distributed systems. Their massive scale, while allowing for an unprecedented capacity and performance potential, also comes with certain challenges. These challenges must be addressed from the very conception of the system design, supporting software, and applications.

First, *centralized management* is impractical for systems of such scale, due to the large number of entities involved, be they computers or data items. Tracking membership (which nodes join or leave the network), locating data and services among millions of nodes, and generally monitoring the system, are no longer possible to realize in a centralized manner. Such an omniscient node would have to maintain a *global and consistent view* of the system, becoming a bottleneck for system performance. Additionally it would constitute a single point of failure and a perfect target for attacks. Spreading the load of such operations on multiple nodes is much more appropriate in large-scale systems. Then, each node is responsible for only a fraction of global system knowledge, called the node's *local view* of the system. Maintaining individual local views is less complex than maintaining a single, centralized, global, and consistent view. It allows for greater scalability due to the elimination of single points of failure.

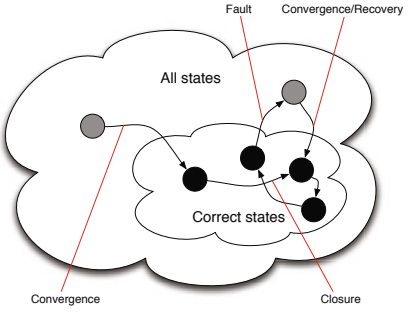


Fig. 1. Principle of self-stabilization

Second, systems of such scale are inherently of *highly dynamic nature*, either due to nodes leaving, joining, or merely failing. If in small- and medium-scale distributed systems faults were considered as exceptions and were mitigated by traditional reparation mechanisms (e.g., checkpointing and restarting individual nodes or the whole application), in large-scale systems faults must be considered as the norm. The number of nodes joining and leaving the system during any period of time is expected to be high. The rate at which nodes join and leave is often referred to as the *node churn* (or *churn*) of the system. High level of churn imposes that the removal of failed/leaved nodes and the insertion of newly joined ones must be integrated at the core mechanisms used for building large-scale applications and systems. The reader may find experimental studies of churn in real systems in [9,10].

Third, the *high complexity* of large-scale distributed systems make their explicit management in case of misconfiguration or faults totally impractical. It is thus vital that algorithms and protocols involved exhibit self-* properties: self-configuration, self-stabilization and self-optimization are some but few examples of these properties. The self-organization property, illustrated by Figure 1 ensures that divergence from a correct state of the system (e.g., only valid and non-failed nodes are available as potential communication partners in nodes' views) is possible but automatic recovery eventually happens as part of the protocols' operation and not due to the help of some external mechanism (e.g., human operation or restarting of the system).

This paper focuses on the use of the gossip-based communication paradigm for building large-scale applications[1].

1.2 Outline

We seek to survey, motivate, and exemplify a representative set of gossip-based building blocks for large-scale systems. These mechanisms are meant to be integrated as components of large and complex systems, and we give examples of such integration whenever applicable.

The remaining of this document is organized as follows.

- Section 2 introduces gossiping, from its seminal use to modern version.
- Section 3 presents gossiping protocols that emerge random overlay networks.
- Section 4 presents gossiping protocols that emerge structured overlay networks.
- Section 5 presents gossiping protocols for overlay slicing.
- Section 6 presents gossiping protocols for data aggregation.
- Section 7 presents additional uses of gossiping protocol for large scale distributed systems.
- And finally, Section 8 concludes our work.

Note that for each section, we do not only provide the description and motivation of the corresponding protocol but also provide additional links that are meant to guide the readings of a reader wishing to delve deeper into the subject.

2 Gossiping Protocols: From Traditional to Modern

Gossiping, also known as *epidemic*, protocols are not a new concept in computer science. They have been around for nearly three decades. However, the daunting Internet growth has created new challenges, and has shaped gossiping protocols in a new way.

This section introduces the seminal gossiping system, Clearinghouse, as well as the modern ones, explaining the reasons that led to the latter.

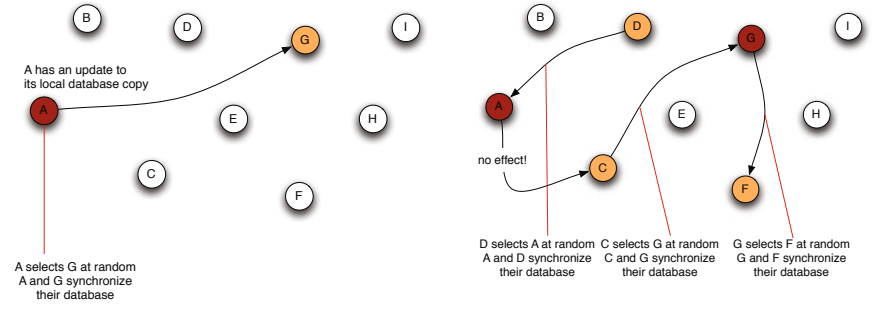
[1] Another introduction to gossip-based networking can be found in [11].

2.1 Clearinghouse: Synchronization of Database Replicas

The seminal paper by Demers *et al.* on the Clearinghouse project [12] introduced the use of gossiping in medium-scale networks for propagating updates to replicas of a database. The mechanisms introduced in this work still lie at the core of all proposals for gossip-based data dissemination.

The Clearinghouse project [12] involved a database replicated across a set of a few hundred replicas, dispersed across diverse geographic areas, where updates were allowed at any one replica of the system. Maintaining consistency across all replicas in the face of updates was a major objective. More accurately, the objective was to maintain consistency across all *alive* replicas, given that individual replicas would occasionally fail, as is the norm in any large scale system. Furthermore, the system should be highly failure resilient, that is, the failure of any node or set of nodes should not hinder the propagation of updates across the remaining set of alive replicas.

This work introduced two gossiping algorithms for the propagation of updates to all replicas, namely Anti-Entropy and Rumor Mongering. In *Anti-Entropy*, each node "gossips" periodically, that is, it periodically picks a *random* other node among all alive ones, and they exchange some data to synchronize their replicas. Figure 2 illustrates an example of anti-entropy.



(a) Initial state: node A has a local update that needs to be propagated

(b) Propagation of the update through anti-entropy operations

Fig. 2. Propagation of the update from node A using *anti-entropy*

In *Rumor Mongering*, nodes are initially "ignorant". When a node has a new update, it becomes a "hot rumor". While a node holds a hot rumor, it periodically selects a random node among the alive ones, and forwards the update to it. After having forwarded the update to a number of nodes that were already hot rumors, it stops being a hot rumor and, thus, maintains the update without forwarding it further. Figure 3 shows an example of rumor mongering.

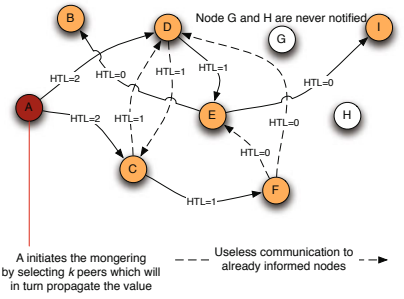


Fig. 3. Propagation of the update from node A using *rumor mongering*

2.2 Today's Challenges

Gossiping protocols have shown to possess a number of desirable properties for data dissemination, notably fast convergence, symmetric load sharing, robustness, and resilience to failures. The same applies for data aggregation, node clustering, network slicing, and other forms of decentralized data manipulation, as we will see in subsequent sections. We will be referring to these gossiping protocols as *traditional gossiping*.

However, traditional gossiping protocols are based on a common assumption: the *complete view of the network* by every node. This is in fact dictated by the need of every node to periodically sample the network for a random other node to gossip with. Although this assumption is acceptable for fixed sets of up to a few hundred machines, it becomes a serious obstacle in networks that scale to tens of thousands or millions of nodes. This is clear given the dynamicity inherent in systems of such scale, given the probability of nodes to crash or to voluntarily leave or join. Imagine the join of a single node triggering the generation of millions of messages, to inform the millions of other nodes of its existence.

Ironically enough, the solution to the complete network view assumption of traditional gossiping protocols is given by a new generation of gossiping protocols that handles overlay management. These protocols are generally known as the Peer Sampling Service and will be studied in Section 3.

In the Peer Sampling Service, nodes maintain just a *partial view* of the network, rather than a complete view. Periodically, a node picks a neighbor from its partial view. They exchange some data, which more specifically is *membership information*. That is, they send each other some of the neighbors they have. This way nodes refresh their partial views, and update them with new information on participating nodes. Deferring certain details to Section 3, these partial views can provide nodes with links to other nodes picked uniformly at random out of the whole network, bypassing the need for complete view of the network.

Figure 4 presents the model of executing traditional gossiping protocols (such as gossip-based dissemination) on very large scale systems. Each node executes

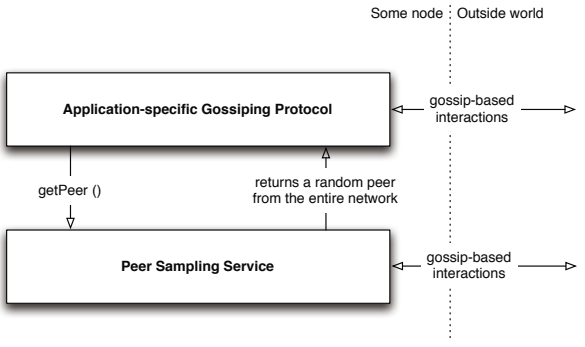


Fig. 4. Gossip-based dissemination using a Gossip-based Peer Sampling Service to implement its selectPartner() operation

(at least) two gossiping protocols. The first one (top) is the traditional gossiping protocol needed by a certain application. The second one (bottom) is the Peer Sampling Service, used to manage membership and serving as a source of uniformly randomly selected nodes from the whole network for the first gossiping protocol.

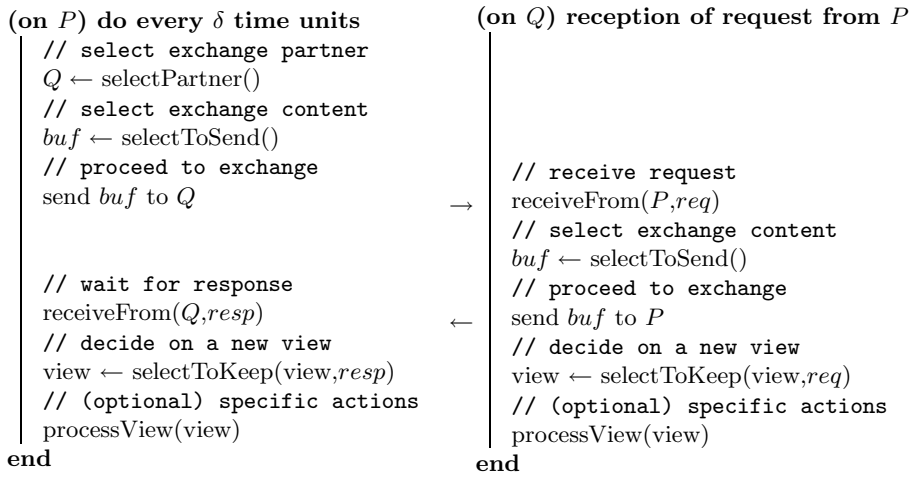
2.3 The Gossiping Framework

We define a gossiping framework, which is generic enough to apply to all gossiping protocols, both traditional and peer sampling ones. Each node (or peer) in the system maintains a local, often partial, view of the system. This view can be of various types: a replica of a database, a set of published events, localization information, or even sets of other peers participating in the system.

Gossip interactions are pair-wise periodic exchanges of data among peers. Each peer periodically selects a partner to gossip with, amongst the nodes it knows in the system (selectPartner() function). Then, it selects the information from its local view that will be exchanged with this partner (selectToSend() function). The partner proceeds to the same operation, which results in a bidirectional exchange between the partners. Thereafter, each of the two decides on its new local view based on the information it had before the exchange (available in its view) and the one received (in *resp* or *req* buffers). Additionally, extra actions (e.g., notifying the upper layers that the local view has changed) can be taken depending on the protocol considered.

All the protocols we present in this document follow this very simple algorithmic framework, where no global vision of the system is assumed whatsoever, and where only local update decisions based on the local view and the received information are key to local convergence, and as we shall see, to the global convergence in the system as a whole.

Gossip-based networking is often based on probabilistic decisions (e.g., for the selection of the partner, the selection of the data to send, etc.). The local



Algorithm 1. Gossip-based interaction framework

decisions made by each node are often driven by local convergence criteria. The convergence to a *better local view* according to these criteria leads to global convergence: the state of the system as a whole, when carefully engineered, converges to an expected property that in fine allows implementing the desired service, without any assumptions on one node having a global view of the system. Moreover, the many interactions between nodes in the system support a certain level of redundancy, which is key to robustness: the loss of some of the interactions (due to failed nodes, message loss, etc.) can impact the convergence speed but seldom impact the eventual convergence. The local convergence vs. global state is the key for the self-stabilization, self-repair and self-configuration offered by gossip-based protocols.

2.4 Further Reading

A number of systems employ gossiping techniques. Many are focused on scalable group communication and multicast [13,14,15,16,17,18,19]. Others have focused on data aggregation [20], live streaming of video [21], maintenance of Distributed Hash Table routing tables [22], social network links to propagate data more efficiently [23], or specific network characteristics for gossiping with lower cost [24]. Finally, a number of researchers have worked on theoretical analysis of gossiping properties [25,26].

Dynamo [27] is a distinguished example of a gossip-based system applied in an industrial environment. More specifically, it is used in Amazon's infrastructure to spread indexing information across all servers involved in a Distributed Hash Table handling crucial data, such as customer records.

3 Random Overlays

As explained in the previous section, a number of gossiping protocols rely on the ability to select random samples of alive nodes from the network. Clearly, providing each node with a complete view of the network is unrealistic for very large scale networks, particularly in the face of node churn, that is, nodes joining and leaving. Similarly, building a centralized service for maintaining such information is not a viable solution either.

Along these lines, a class of entirely *decentralized* protocols has emerged to collaboratively maintain membership information. These protocols are collectively known as the Peer Sampling Service [28], and they are based on a gossiping framework themselves.

In a nutshell, each node maintains a small (e.g., a few dozen nodes) *partial view* of the network, and periodically refreshes its partial view by gossiping with one of its current neighbors. It turns out that, by following this gossiping paradigm with certain policies, the partial view of each node constitutes a periodically refreshed sliding random subset of all nodes in the network. Making a random selection out of a random subset of all nodes is equivalent to making a random selection out of *all* nodes. This is exactly the assumption which traditional gossiping protocols are based on: sampling peers from the whole network at random. It becomes now evident that, by employing a Peer Sampling protocol, a node is able to select peers at random out of the whole network by means of a *local operation*. This essentially overcomes the scalability barrier for executing traditional gossiping protocols in very large scale networks.

Sampling peers uniformly at random is not the sole utility of Peer Sampling protocols. It turns out that by running a Peer Sampling protocol on a large set of nodes, the nodes self-organize in an overlay that shares a lot of similarities with *random graphs* and inherits most of their properties. Namely, the overlay becomes very robust and extremely resilient to failures, in the sense that failures, even large scale ones involving much more than half of the nodes do not put the *connectivity* of the overlay at risk.

Peer Sampling refers to a *family* of protocols, whose design space is extensively analyzed in [28]. This analysis is out of the scope of this paper. Instead, we will focus on the two most prominent instance protocols of the Peer Sampling Service, namely NEWSCAST [29,30] and CYCLON [31].

3.1 The NEWSCAST Protocol

In NEWSCAST each node maintains a small partial view of the network of length ℓ, and periodically picks a random node from it to gossip with. The two nodes share with each other their views, including newly generated links to themselves.

The principal design objective in NEWSCAST is to keep overlay links *fresh* by giving newer links priority over older ones. In doing so, NEWSCAST policies consider the *age* of a link, that is, the time elapsed since the link was injected into the network by the node it points at.

Link ages can be precisely determined if links are timestamped at generation time, assuming network-wide time synchronization is possible. Otherwise, they can be sufficiently approximated by associating each link with an *age* counter. Here we follow the second—more realistic—approach.

In terms of our gossiping framework shown in Algorithm 1, NEWSCAST implements the following policies:

selectPartner() Select a random node from the view. Also, increase the age of all nodes in the view by one.

selectToSend() Select all nodes from the view, and append own link with age 0.

selectToKeep() Merge received links with own view, sort by the age, and keep the ℓ freshest ones, including no more than one link per node.

Note that after a gossip exchange between nodes P and Q, the two nodes have the same view, except for a link to each other. This, however, is a temporary situation, as next time they gossip (either initiating it or being contacted by others) their views will most likely be mingled with views of different other nodes.

3.2 The CYCLON Protocol

CYCLON [31] is a Peer Sampling protocol, where view refreshing is based on *exchanging contacts*. That is, a node sends a few links to its gossiping partner, and receives the same number of links in return. Each node accommodates *all* received links by discarding the links it just sent away. The main intuition behind this operation is to mix links, resulting in overlays resembling random graphs.

Like in NEWSCAST, the age of a link is also utilized in CYCLON, alas in a different way. It is used to select which neighbor to gossip with, rather than to select which links to keep in the view. This serves two fundamental goals, that will be discussed later in this section.

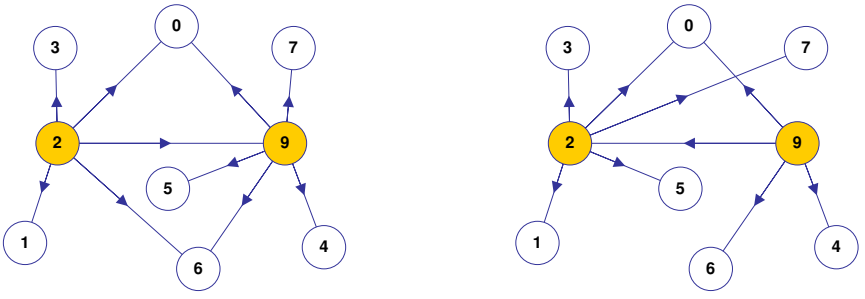
Let g denote the number of links traded in a gossip exchange. In terms of our generic gossiping framework shown in Algorithm 1, CYCLON employs the following policies:

selectPartner() Select the node whose link has the *oldest* age. Also increase the age of all links in the view by one.

selectToSend() Select g random links, and *remove* them from the view. If this is the initiating node, the link selected in selectPartner() should be among these g links, and after removal it should be substituted by a link to itself, with age 0.

selectToKeep() Add all g received links to the view, by replacing the g links selected in selectToSend().

Note that after a gossip exchange, the link between the two nodes involved changes direction, as illustrated in Figure 5. E.g., if node P knows node Q and selects it as a gossip partner, after the gossip exchange P will have discarded Q from its view, while Q will deterministically know P. In other words, P's indegree



(a) Before swapping (b) After swapping

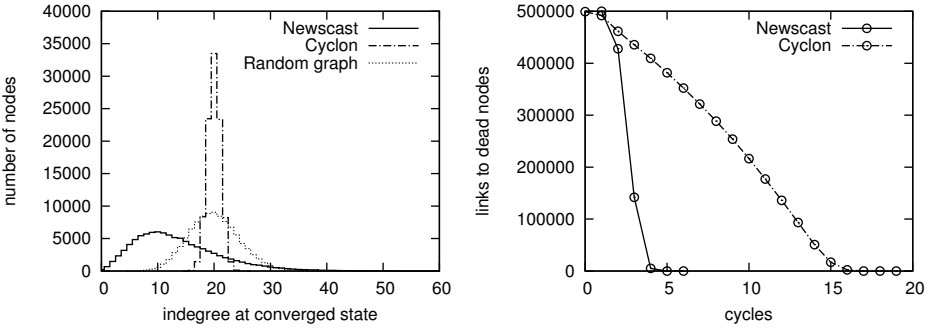
Fig. 5. An example of swapping between nodes 2 and 9. Note that, among other changes, the link between 2 and 9 reverses direction.

was increased by one, while Q's indegree dropped by one. Third nodes' indegrees have not been altered, even if some nodes changed from being neighbors of P to being neighbors of Q or the other way around.

This provides for an interesting self-adaptive mechanism for indegree control. A node's indegree increases when it initiates gossiping, which happens at constant intervals (due to Cyclon's periodic operation). However it decreases when it is contacted by another node, which happens at a frequency proportional to the node's indegree: the more known you are, the more gossip exchanges you will be invited to in a given period. Statistically, if exactly ℓ other nodes know you, you will be contacted exactly once per gossiping period, and as you will also initiate exactly one gossip exchange, your indegree will remain stable and equal to ℓ. However, the lower a node's indegree is below ℓ, the faster it will grow higher, while the higher it is above ℓ, the faster it will drop lower. This leads to a natural equilibrium of indegrees, a self-adaptive mechanism for balancing links evenly across all nodes.

Regarding the selection of the *oldest* link for a gossip exchange, as mentioned earlier it serves two goals. The first one is to limit the time a link can be passed around until it is chosen by some node for a gossip exchange. Since by selecting a link for a gossip exchange also removes the link from the network, selecting always the oldest one prevents links to dead nodes from lingering around indefinitely. This results in a more up-to-date overlay at any given moment.

The second—and far less obvious—goal is to impose a predictable lifetime on each link, in order to control the number of existing links to a given node at any time. During one gossiping period, a node P initiates *one* gossip exchange, therefore pushing its oldest age link out of its view, and increasing all other links' ages by one. Also, P is contacted on average by *one* other node for a gossip exchange, thus accepting a new link of age 0 in its view. As a result, a node's view contains on average one link of each age, from 0 to $\ell - 1$. This means that links selected for gossip exchanges are typically of age around $\ell - 1$. In other words, a pointer has a lifetime of about ℓ cycles. This implies that



(a) Indegree distributions (b) Links to dead nodes

Fig. 6. Some main properties of NEWSCAST and CYCLON

besides the constant birth rate of links, their death rate is also close to constant, which results in an almost constant population of ℓ links for each node. This is validated by extensive simulations.

3.3 Properties

Although a detailed discussion on properties of these protocols is out of the scope of this paper, it is worth noting the effect that certain policies have on some of the properties.

Figure 6(a) shows the indegree distribution for NEWSCAST and CYCLON, running on 100K nodes with view length $\ell = 20$ for both protocols. The self-adaptive mechanism for indegree control in CYCLON detailed in Section 3.2, becomes evident in this graph. Indegrees follow a very narrow distribution, centered around the nodes' outdegree (i.e., their view length). Each node has an indegree of $\ell \pm 3$. This results in higher robustness of the overlay in the face of errors, as no node has indegree of lower than 17, which means there are no "weak links" in the topology. In NEWSCAST, the indegree distribution is very much spread out, which is an expected outcome of the nature of gossip interactions: a link to a node can either be duplicated on both gossiping partners or completely discarded, which results in high fluctuations of node indegrees. Specifically, we note that there are several nodes with indegree 0, becoming more vulnerable than others in the face of failures.

Another implication of the indegrees is that CYCLON offers much better load balancing, as nodes are invited to the same number of gossip exchanges per time unit. Contrary to that, in NEWSCAST there is a long tail of nodes with indegrees up to 60, which receive proportionally more gossip requests (and therefore load) per time unit. Nevertheless, extensive simulations in [28] showed that in NEWSCAST nodes fluctuate across the whole spectrum of indegrees withing a few gossiping periods, therefore in the long run load is fairly balanced among NEWSCAST nodes as well.

Figure 6(b) shows the efficiency of each protocol in relieving the overlay of links to dead nodes. More specifically, in this experiment an overlay of 100K nodes with views of length $\ell = 20$ was let emerge by each protocol. At some point a very large failure was simulated by killing half of the network, and letting exactly 50K nodes survive. At that point, statistically $\ell/2$ links of each surviving node was pointing at dead nodes, accounting to 500K links. The graph in Figure 6(b) shows how the total number of links to dead nodes drops in each of the protocols, as a function of cycles (a time unit representing one gossiping period). NEWSCAST's eagerness on keeping the *freshest* links is evident in this graph, as links to dead nodes vanish within six cycles. Contrary to that, CYCLON's self-adaptive control of links' lifetimes to ℓ gossiping periods is clear in this figure, as it takes exactly ℓ cycles for eliminating all links to dead nodes. With respect to this metric, NEWSCAST is more efficient, and shows it is better at handling overlays of very high node churn.

3.4 Further Reading

Work on overlay management for random overlays assumes the understanding of fundamental concepts such as random graphs [32], scale free networks [33], and small worlds [34,35].

With respect to computer networks, Lpbcast [17] is a peer sampling protocol targeted at broadcasting messages. Scamp [36] is a reactive protocol that creates a static overlay that resembles a random graph. HyParView [37] is a gossiping protocol that creates overlays targeted at disseminating data in the face of high node churn.

4 Structured Overlays

Besides creating random overlays as a basis for massively decentralized systems, many distributed applications require structured overlays to operate on. Examples include, but are not limited to, clustering nodes based on interest (e.g., for a file sharing system), sorting nodes based on some metric (e.g., ID, load, memory, etc.), forming more complex structures (e.g., distributed hash tables, publish/subscribe systems, etc.) and more.

T-MAN [38,39] and VICINITY [40,41] are two very similar gossiping protocols that provide a generic *topology construction* framework, suitable for the construction of a large variety of topologies. Through such a framework, nodes flexibly and efficiently self-organize in a completely autonomous fashion to a largely arbitrary structure. The advantages of these frameworks are their generic applicability, flexibility, and simplicity.

The target topology is defined by means of a *selection function*, which selects for each node the set of ℓ neighbors it should be linked to. This selection function is executed locally by every node to determine its neighbors. The selection is made based on some application-specific data associated with each node, which is called the node's *profile*. In topology construction protocols, each link to a node also carries that node's profile.

When fed with the complete list of nodes and their profiles, the selection function returns a node's optimal neighbors for the target topology. When fed with a subset of the nodes, it returns a selection of neighbors that brings the overlay as close to the target topology as possible.

Typically, the selection function is based on a globally defined peer proximity metric. Such a metric could include semantic similarity (see Section 4.1), ID-based sorting, domain name proximity, geographic- or latency-based proximity, etc.

Like in other gossiping protocols for overlay management, each node maintains a partial view of the network of length ℓ. As mentioned above, each link to a node also carries that node's profile. The protocol framework is similar to that of Peer Sampling Service protocols, except that nodes decide which links to keep in their views based on the selection function.

In the context of the hooks defined in our generic gossiping framework of Algorithm 1, topology construction protocols employ the following policies:

selectPartner() Select a random link from the view.

selectToSend() Select all links from the view, and append own link with own profile.

selectToKeep() Merge received links with own view and apply the selection function to determine which ℓ links to keep in the view, including no more than one link per node.

Figure 7 depicts a sample gossip exchange for topology construction, assuming a selection function that opts for minimizing the 2D Euclidean distance between nodes.

A key point in topology construction protocols is the *transitivity* exhibited by the selection function. In a selection function with high transitivity, the "better" a selection node Q is for node P, the more likely it is that Q's "good" selections are also "good" for P. This transitivity is essentially a correlation property between nodes sharing common neighbors, embodying the principle "my friend's friend is also my friend". Surely, this correlation is fuzzy and generally hard to quantify. It is more of a desired property rather than a hard requirement for

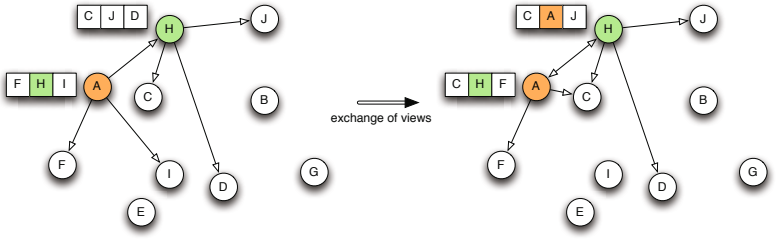


Fig. 7. Example of a Vicinity exchange and resulting views. The distance here is the 2D Euclidean distance and the nodes seek to find the $\ell = 3$ closest nodes in terms of this distance metric.

topology construction. The higher the transitivity, the faster an overlay will converge to the desired topology.

There are two sides to topology construction. First, assuming some level of transitivity in the selection function, a peer should explore the nearby peers that its neighbors have found. In other words, if P_2 is in P_1's view, and P_3 is in P_2's view, it makes sense to check whether P_3 would also be suitable as a neighbor of P_1. Exploiting the transitivity of the selection function should then quickly lead to high-quality views. The way a node tries to improve its view resembles *hill-climbing* algorithms. However, instead of trying to locate a single optimal node, here the objective is to optimize the selection of a whole set of nodes, namely the view. In that respect, topology construction protocols can be thought of as distributed, collaborative hill-climbing algorithms.

Second, it is important that *all* nodes are examined. The problem with following transitivity alone is that a node will be eventually searching only in a single cluster of related peers, possibly missing out on other clusters of also related—but still unknown—peers, in a way similar to getting locked in a local maximum in hill-climbing algorithms. This calls for randomized candidate nodes to be considered too in building a node's view. This points directly at the two-layered approach depicted in Figure 4.

Survey of overlay networks [42].

4.1 Test Case: Interest-Based Overlays

A direct application of Vicinity and T-Man is the self-organization of nodes participating in a social network in a way that reflects their interests. Social networks inherently exhibit interest locality, that is, a user encompassing content on a certain topic is highly likely to address additional content on that same topic or related ones. Additionally, social networks tend to share many properties with Small Worlds [43,44], that is, highly clustered networks of short diameter. High clustering (i.e., the friend of a friend is likely to be a friend) implies a highly transitive selection function in a topology construction protocol.

In [40], Voulgaris and van Steen apply Vicinity (with Cyclon as the underlying Peer Sampling Service instance) on nodes participating in the e-Donkey [45] file sharing network, to cluster them based on the degree of overlapping in their shared file collections. The selection function sorts a node's neighbors based on the number of shared files in common to the node's own collection, and selects the top ℓ ones. The experiments on 12,000 nodes indicate that by setting the view length to $\ell = 10$ neighbors, over 90% of the optimal relationships between nodes are established within 50 rounds of the protocol, starting from an arbitrarily connected initial topology. Furthermore, these 10 "semantic neighbors" per node prove to be capable of serving on average *one third* of each node's queries for new files, a ratio that significantly boosts decentralized search performance.

The same setting can be applied for automated *recommendations* in a decentralized file sharing system. Files that are popular among someone's "semantic neighbors" are likely to be interesting for that user as well.

4.2 Further Reading

GosSkip [46] employs gossip to implements skip lists [47] in a distributed fashion similar to SkipNets [48]. RayNet [49] uses gossiping to create structured overlays inspired from Voronoi diagrams. Other work, like [50] create overlays that resemble Small World networks [51,43]. Rappel [52] uses gossip to leverage clustering of interests to build dissemination trees.

5 Overlay Slicing

Overlay slicing is an alternative form of overlay management to the clustering mechanisms introduced in Section 4. It relates to the problem of overlay *provisioning*. Slicing allows to separate a network in relative-sized groups in a self-organizing manner.

Here, we are no longer interested in creating a graph of links between nodes in the network that form a given structure, but rather to split the network in a set of *slices*. Each slice has a size that is expressed as a proportion of the total network size, and that can aggregate the nodes with the highest value for a given, node-specific metric. It is important to note here that the actual size of the network does not need to be known to proceed to the slicing operation.

Slicing has many useful applications. It allows to provision parts of the network to dedicate each such parts to a particular applications, or to different services pertaining to one application:

- One may want to provision the most powerful nodes to support the critical services of some application. In this case, each node is attached to a metric that depicts its relative power. One can think of the available bandwidth, the available storage capacity, the processing power, among others. The nodes that have a smallest value for this metric can be dedicated to less critical operations of the applications, e.g., the powerful nodes can support a naming mechanism that allows to locate data or services, while the less powerful nodes can support a less critical part of the application, such as caching or monitoring mechanisms.
- One may want to split the network into groups regardless of the nodes' characteristics, in order for each slice to be used for a different application with the same nodes' characteristics distribution for each slice. For instance, if a network needs to support three different applications, and the nodes supporting each application must be dedicated to only this application, one can express the size of each slice to be $\frac{1}{3}$ of the network regardless of the nodes' characteristics.

An example of slicing based on nodes' characteristics is given by Figure 8(a). Here, we are interested in creating three slices. The first slice shall contain half (50%) of the network, and contain the nodes that have the smallest value for the considered metric (say, the available disk capacity). The second slice must contain the 30% nodes that have intermediate values for this same metric. Finally, the third slice must be composed of the 20% nodes with the highest values

for the metric, say, with the highest available disk capacity. We note that we do not want the assignment of nodes to slices to be static. If the metric changes for some node, e.g., the disk capacity is reduced due to the use of this resource by another application running on the same machine, then the self-organization objective requires that the assignment of nodes to slices reflects this change, and so on for the lifetime of the application (unless one needs to fix the assignment once and for all due to application requirements, which would require freezing the assignment).

The assignment of one node to some slice is autonomous. After the gossip-based overlay slicing mechanism has run for enough cycles, each node must be able to determine which slice it belongs to. We consider that the parameters and relative size of slices are known by all nodes, that is, each node knows what metric to consider for itself and its peers, and what are the relative sizes of the slices (e.g., {50%,30%,20%} in the case of Figure 8(a)).

The overlay slicing protocol based on gossip-based networking that we present in this document is the one proposed by Jelasity and Kermarrec in [53]. In order to determine which slice they do belong to, nodes must determine what is the relative position of their metric value in the set of all metric values, if such an ordered set was known. Obviously, due to the large-scale of the considered network, it is impractical to consider that any one node will know this sorted set. The relative position must be known without globally sorting all the nodes' metrics in order to decide on any one node's position. This relative position determination is illustrated by Figure 8(b). Let us consider node J. Here, in order for J to determine that it belongs to the second slice, the node must determine an approximation of:

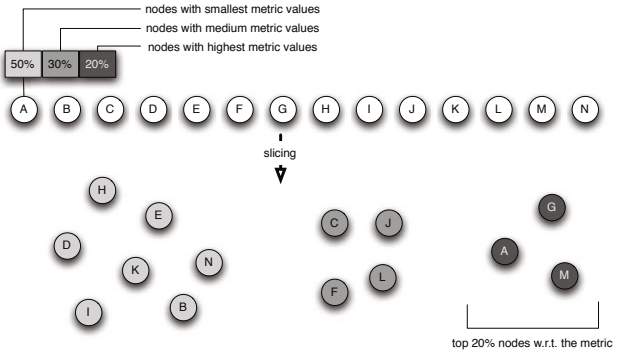
- the number of nodes that have a higher value than J for the metric;
- the number of nodes that have a smaller value than J.

The position of nodes in the set is approximated by a value in [0:1]. More specifically, this position is the *relative position* of the node in the set of all metric values. Note that while the relative positions are contained in a bounded range, this is not the case for the metric values, which can range over any space.

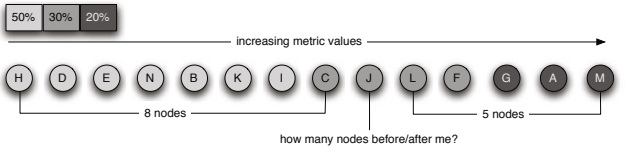
The case where the network must be split into relative-sized slices but independently of any metric is simply a special case. In order to allow a random sampling of nodes in each slice (according to the slice size relative to the size of the network of course), each node simply emulates a metric by picking a random value in [0:1] as its metric value.

Each node starts with a random relative position in [0:1], which obviously is unrelated to the final expected position. This starting position is illustrated by the upper representation of Figure 8(c). We can see here that the initial relative position of J, which is around 0.3, has nothing to do with the expected relative position, that is, around 0.6.

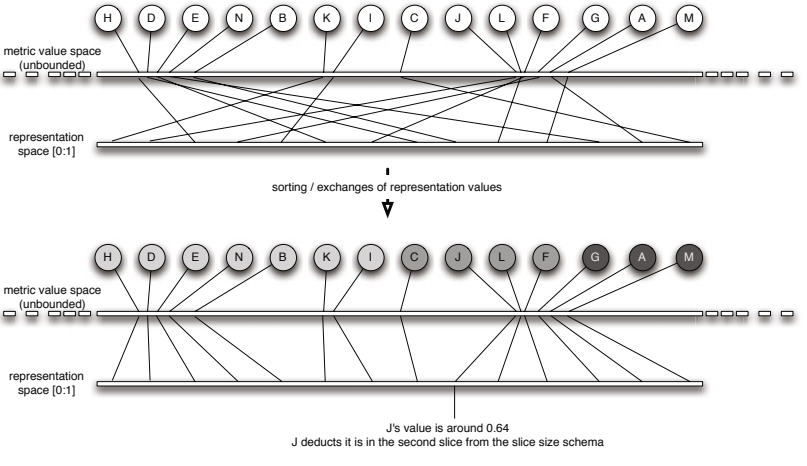
After deciding randomly on these initial values, nodes engage in a gossip-based self-organization, that must lead to each node holding a relative position that reflect its slice, as illustrated on the bottom representation of Figure 8(c). Here, one can see that the relative position of J, being 0.64, reflects correctly the



(a) Problem definition

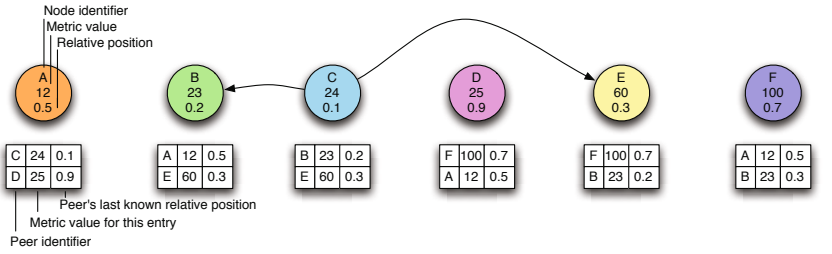


(b) Using the relative position of a node to autonomously determine its slice

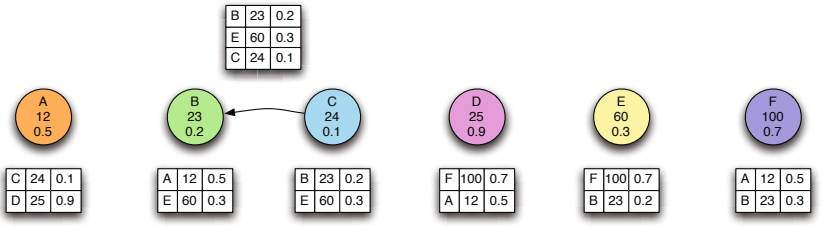


(c) Gossip-based sorting to determine relative positions

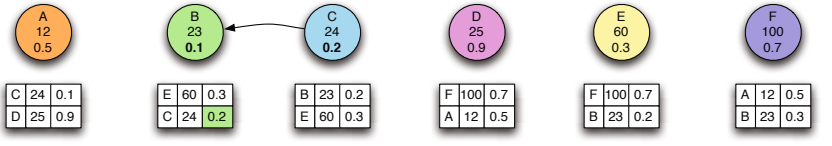
Fig. 8. Gossip-based *slicing*: problem representation and determination of the relative position using gossip-based sorting



(a) selectPartner() operation: there is no order violation for peer E, but there is an order violation for peer B (B's metric value 23 is smaller than C's metric value 24 but their metric value have an opposite order: 0.2>0.1). B is selected.



(b) selectToSend() on peer C: selection of the entire view, plus an entry for C.



(c) selectToKeep() on peer B: after removing duplicates entries (not shown), C sees that there still is an order violation. Henceforth, it takes the current relative position of C. C does the same upon receiving the view from B.

Fig. 9. A pair-wise interaction for gossip-based sorting that exchange the relative position of nodes B and C and resolves an order violation w.r.t. their metric values

slice it must belong to. The final positions are obtained by pair-wise *gossip-based sorting* of the relative position with respect to the metric of the two nodes. Each node has a view, of bounded size c, and can pick random nodes from a Peer Sampling Service.

Figure 9 represents a single gossip-based interaction between two nodes B and C. This operation, following the framework of Algorithm 1, is as follows:

selectPartner() (Figure 9(a)) The peer selects at random a partner for the exchange among the peers it knows, including the peers it obtains from the Peer Sampling Service (see Section 3), and for which an order violation exists (that is, the metric of the peer is higher than the metric of the initiating node but their relative values follow a different order, or conversely). This peer

is selected in order to proceed to an exchange of the relative values and to resolve the order violation[2].

selectToSend() (Figure 9(b)) The node sends its entire view, and includes itself in the sent view (as links are unidirectional, there is indeed no guarantee that B would know the metric value and the relative position of C that it needs for the exchange).

selectToKeep() (Figure 9(c)) If there still is an order violation between the gossip initiator and the partner, the partner exchanges its own relative position with the one of the initiator, and returns its view in the very same way. The initiator, upon receiving the view, also trades its relative position with the one of the partner. Henceforth, the order violation is resolved. Note that it can happen that the order violation that was witnessed by the initiator did not longer exist. Indeed, the partner node may have already exchanged its value with another node since the last update of the view entry, resolving the violation. In this case, the gossip exchange is simply cancelled.

Each such gossip step reduces the global *disorder metric*, that is, the average squared distance between a nodes' relative position and its "correct" position. This metric is simply the standard deviation over the relative position incorrectness. Interestingly, the convergence of the gossip-based sort is empirically independent from the size of the network. Within 20 cycles (during one cycle, each node exchanges with one other node if there exist one with an order violation in its view or in the Peer Sampling Service view), the average error is no more than 1% of the network size, already allowing a very good estimation of the slice a node belongs to: in a 10,000 nodes network, nodes are on average 100 positions away from their ideal position (which means that, if they are not considered in the correct slice, they still have very similar metric values to the correct nodes for that slice). If one lets the protocol converge for 40 cycles, the average error becomes 0.1%, which is a negligible value in a dynamic and large-scale systems such as the ones considered.

5.1 Further Reading

In [54], the authors improve over the original slicing protocol presented in [53] in two ways. First, they propose measures to speed up the convergence of the protocol by using a local disorder measure for the selectPartner() operation: peers choose the gossip partner with which an exchange is the most likely to reduce the global disorder measure. Second, they revisit the use of random values sorting for the slicing operation. The rationale is that, when the metric value is based on the nodes' characteristics, e.g., the uptime or the available bandwidth, there is a correlation between this metric and the relative position that tends to bias the distribution of relative values. Instead of using random values sorting, the authors propose to estimate the rank of nodes based on the history of

[2] Note that an additional aging mechanism can allow to ensure that nodes are contacted within a bounded amount of time, in order to detect changes/failures in a timely manner. For the sake of simplicity, we do not consider this optimization in this paper and encourage the reader to refer to [53] for details.

recently-seen values for the relative positions, which proves to be more robust to churn and bias. In [55], the Sliver protocol is presented, that integrates further optimizations to gossip-based slicing.

6 Distributed Aggregation

Aggregation is the collective estimation of system-wide properties, expressed as numerical values. It is a key functionality to a number of large-scale distributed systems, in particular to implement monitoring mechanisms.

The properties that can be aggregated by gossip-based distributed aggregation can relate to a large variety of metrics. One can mention the average system load, the identity of the node with the lowest or highest load or disk capacity, the total available disk capacity in a distributed storage system, among others. As we shall see in the Subsection 6.1, aggregation can also be used to solve in an elegant and autonomous way a difficult problem in decentralized large-scale systems: network size estimation.

Each node starts with its own value for the metric that is to be aggregated. Thereafter, aggregation should be carried out collectively by all participating nodes in a purely distributed fashion, and the result(s) of the aggregation should become known to all nodes.

We present as an example in this section a basic aggregation protocol that follows the push-pull gossip-based networking paradigm. This protocol appeared in [56]. Each node has a local *estimate* of the property being aggregated and a set of *neighbors*. At random times, but once every δ time units, a node picks a random neighbor and they exchange their local estimates. This random neighbor is typically obtained by calling the getPeer() operation of the Peer Sampling Service (see Section 3), as we assume that no global view of the system exists at any node. After the exchange, each node updates its local estimate based on its previous estimate and the estimate of the partner it has received.

Averaging constitutes a fundamental aggregation operation, in which each node is equipped with a numeric value, and the goal is to estimate the average, or arithmetic mean, of all nodes' values. We start by describing the calculation of the average, and later show how it can form the basis for the computation of other aggregates, as detailed in Jelasity et al. [56].

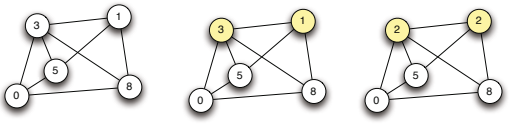


Fig. 10. An exchange in average calculation

In averaging, a node updates its estimate to the average between its previous local estimate and the estimate received. That is, when nodes p and q with estimates s_p and s_q proceed to a gossip exchange, their estimates are updated as follows:

$$s_p = s_q = \frac{s_p + s_q}{2}$$

Note that the sum of the two nodes' estimates does not change, therefore neither does the global average. However, the variance:

$$V = \sqrt{\frac{1}{N}\sum_{p=1}^{N}\left(s_p - \sum_{q=1}^{N} s_q/N\right)^2}$$

decreases after each exchange, unless s_p and s_q were already equal, in which case it remains unaltered. (N denotes the size of the system.) Experiments and theoretical analysis in [56,57,58,59,60] show that the variance V converges to zero. Moreover, it converges at an exponential rate, whose exponent depends on the communication graph defining the nodes' neighbors. The rule of thumb is that the higher the link randomization in an overlay, the faster the aggregation convergence. We propose to illustrate this fact by some experimental figure based on simulation.

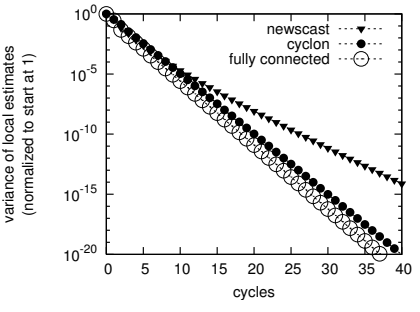


Fig. 11. Performance of gossip-based aggregation, for different peer selection mechanisms: realistic with Peer Sampling protocols Cyclon and Newscast, and ideal, with a purely random selection of gossip partners among all nodes (*fully connected* graph)

We consider a 100,000 nodes network, with the availability of a Peer Sampling Service [28], namely the two instances CYCLON [31] and NEWSCAST [61] described in Section 3. In this case, the gossip exchange partner for the aggregation is obtained from the Peer Sampling Service's view. For the sake of simplicity, we consider only a static network in which either CYCLON or NEWSCAST has converged and where the views are frozen for the duration of the aggregation. Similar conclusions as the ones we present for a static network can be made with a dynamic network where nodes' views keep evolving at each exchange cycle. Figure 11 presents the evolution of the variance as a function of the aggregation cycles (δ time units) elapsed. To have a point of reference, we plot the variance evolution for averaging over a fully connected graph, in which a node exchanges estimates with a node picked randomly out of the whole network.

First, we observe that in all cases the variance converges to zero at an exponential rate. Second, we record a clear difference between the aggregation

efficiency of static CYCLON and NEWSCAST Peer Sampling protocols, the former converging significantly faster. This is a direct consequence of CYCLON's narrow in-degree distribution and very low clustering. Each node has roughly the same number of incoming links, and therefore participates in roughly the same number of estimation exchanges as all other nodes. Moreover, the very low clustering ensures that each node's initial value is uniformly spread across "all directions" of the network, not being confined to any highly clustered subset. This leads to faster convergence of nodes' estimates to the global average. On the other hand, NEWSCAST's skewed in-degree distribution results in an uneven distribution of estimation exchanges across nodes. Also, due to high clustering, local estimates spread quickly within highly clustered communities, but take longer to spread globally.

These observations illustrate that the type of Peer Sampling Service used can have an important impact on the efficiency of the protocols that use them as a basis, and, as pointed out in Section 3, one must carefully consider the impact of the parameters of the Peer Sampling Service on the served protocols.

Other aggregates. The computation of the geometric mean is similar to the computation of the arithmetic mean (average) we just presented. It also exhibits the same convergence properties. One simply needs to replace the update function by: $s_p = s_q = \sqrt{s_p \times s_q}$. The computations of the the average (arithmetic mean) and the geometric mean serve as a basis for the computation of most aggregates. Let us consider however for now that we know the number of nodes in the network N. Obviously, knowing this number in a large-scale system is all but a trivial task; however we explain in the next Subsection how we can actually obtain it based on a gossip-based aggregation calculation.

Based on N and gossip-based aggregation, the following aggregates can be composed:

- The sum of all values in the system, which is useful for instance if one needs to know the total available disk space in a distributed, collaborative data storage system, is simply obtained by multiplying the locally available arithmetic average by the number of nodes: $S = s_p \times N$.
- Similarly, the product can be composed with the geometric mean and the size of the network: $P = {s_p}^N$.
- Finally, the variance can be composed with the computation of the arithmetic average of initial values (here, denoted as $avg(s_p)$) and the arithmetic average of the squares of the initial values (here, $avg({s_p}^2)$):

$$V = avg({s_p}^2) - (avg(s_p))^2$$

Finally, the computation of the minimal and maximal values is also possible using gossip-based aggregation. These computations are simply performed by replacing the update function by: $s_p = s_q = min(s_p, s_q)$, or $s_p = s_q = max(s_p, s_q)$ respectively. One can note here that the propagation of the minimal or the maximal value from the node where it was present initially, to all nodes in the system, will be strictly similar to what a propagation of a single value using the *anti-entropy* mechanism introduced in Section 2 would be, if only one value (that

is, this minimum or maximum) is considered and each node periodically polls another, randomly chosen node for the availability of this value.

6.1 Decentralized System Size Estimation Using Aggregation

We have considered in the previous description of the *sum* and *product* composed aggregations that an important parameter, the total size of the system N, was supposedly available but without explaining how it was obtained. Knowing the total size of a large-scale system is not a trivial task. As no node has a global view of the system, and as the population of nodes is dynamic, the knowledge of N cannot be based on membership information. It is instead necessary to engage into a specific protocol for determining this number N, or more specifically to determine a sufficiently precise estimation of it (as we consider inherently dynamic networks, an exact size estimation would be impractical anyway).

In this Section, we present the use of aggregation for calculating the size of the network in an autonomous manner [56].

Note however that other techniques exist for large-scale decentralized size estimation, that are not necessarily based on gossip-based networking. For instance, Kostoulas et al. [62] rely on interval density sampling of the history of hashed nodes' identifiers over a bounded range to determine the population of nodes in the system; Massoulié et al. propose to use random walks methods and the principle of the inverse anniversary-problem to determine the size of the network based on the occurrences of collisions amongst random walkers [63]. These techniques are experimentally compared to the one we present in this document in [64].

The idea behind the peer counting based on aggregation is conceptually simple: one single peer in the system starts with the value 1, and all other peers start with the value 0. The average value of all the starting values is thus $\frac{(\sum_{N-1} 0)+1}{N} = \frac{1}{N}$, and this is the value that all local estimates will be equal to after the convergence of the gossip-based aggregation. Thereafter, each peer can autonomously use the inverted value of their local estimate to infer N.

However, there is no direct possibility for a single peer in the system for deciding which one of them will hold this initial value of 1 while ensuring that the others will hold the value 0. There is indeed no pre-existing omniscient peer that can decide that it can act as such a *leader* for the aggregation start, or this would require a global view of the network, contradicting the large-scale characteristics of the network. Several decentralized mechanisms exist to decide upon the initial peer in an autonomous way and without the need to maintain any global information. We simply sketch one such mechanism below.

We allow several concurrent instances of the average computation. Each such instance is associated with a different peer starting with the value 1. This peer is called the *leader* of each instance. The messages exchanged during the gossip-based aggregation are tagged with a unique identifier, e.g., the identifier of the leader for the corresponding instance. We note already that running several instances has the inherent advantage of added stability: each node cannot only base its estimation on one, potentially imperfect aggregate (if the system has

not converged, or due to high levels of churn), but on the average of several such aggregates. Nodes decide to act as a leader for a new instance autonomously, based on a parameter i that denotes the target average number of gossip cycles one requires between the start of two aggregation instances. Each node knows the current estimate of the system size (or a reasonable guess for the first round), denoted by N_e. Each node decides at each of the aggregation cycle, that it will start a new aggregation of which it will be the leader with a probability of $\frac{i}{N_e}$.

In order to not let the number of local aggregates grow indefinitely, and in order to support variations in the system size, a periodic restarting mechanism is used. The time is divided in *epochs*, that are local to each node but that use the same duration λ for all nodes. At the end of an epoch, a node delivers the average of the local size estimates obtained during the last epoch to the application, and starts collecting new estimates for the current epoch. The garbage collection of previously known estimates can be done when one entire epoch has passed since their delivery to the application, as no other node will send a gossip exchange request pertaining to these values anymore. The duration of the epochs allows to express the tradeoff between the reactiveness of the size estimation to changes, versus the accuracy of this estimation.

7 The Many Other Uses of Gossip-Based Networking

In this Section, we wish to give the reader a quick overview of the many other uses of gossip-based networking that we did not introduce in details in this paper. Our goal is not to be comprehensive in surveying the usages of gossip (that would require a monograph on its own), but rather to highlight the fact that gossip-based networking can be applied in a wide range of large-scale distributed systems-related problems.

7.1 Self-organizing Publish and Subscribe

The publish and subscribe communication paradigm allows to greatly simplify the design of large-scale applications by providing a *decoupled* communication model. The producers of information do not need to know the interested consumers of the information they produce. Similarly, the consumers do not need to know beforehand which node is likely to issue information that match their interest. Here, producers simply *publish* to the publish and subscribe middleware, and consumers can express their interest in new data by the means of *subscriptions*. It is then the system's responsibility to match the publications to the existing subscriptions, and to route the messages to all interested subscribers. Publish and subscribe mechanisms are very appealing for the design of large-scale applications as they allow to delegate the management of the application data flow to an external service.

Gossip-based networking is a strong contender to build and operate publish and subscribe services. One typically distinguishes between publish and subscribe mechanisms based on the expressiveness allowed for the subscriptions.

The simplest model, topic-based, allows nodes to register their interest to a set of predefined topics, and publications are also attached to one of these topics. TERA [65] is an example of a topic-based publish and subscribe mechanism that leverages the principles of gossip-based structure emergence (Section 4) and a modified Peer Sampling Service. Gossip-based interactions allow to emerge clusters where the nodes interested in the same topic are linked. The publication then requires to reach the correct group, which is made possible by a biased peer-sampling mechanism and random walks, and then to disseminate among the group using gossip-based dissemination. The Rappel system [52] also leverages gossip-based networking to construct dissemination structures that take into account both the network characteristics (delays) and the presence of clustering in the users' subscriptions to reduce the number of links. STaN [66] is another system that leverages gossip for creating a network where nodes with similar interests are grouped together, with the additional guarantee that the subscriptions of nodes are not made public.

Content-based publish and subscribe allows expressing subscriptions based on the content of the events. This is a more powerful model, but since the matching of publications to subscribers shall be done dynamically for each publication, it is typically more complex to support. Sub-2-Sub [8] proposes to let a routing layer emerge from the use of the Vicinity [40] gossip-based networking framework. The very structure of the overlay that emerges allows publications to reach all interested subscribers autonomously, while the system inherits the self-organization allowed by the use of gossip for its construction. DPS [67] leverages gossip-based protocols to group the nodes with overlapping interests (based on their subscriptions) and form the basis of a self-organizing matching and dissemination layer.

7.2 Taming Networked Systems Complexity

Gossip-based networking can also be used to abstract the complexity of the network onto which it operates, in order to ease the development of applications.

A first example is the Nylon [68] NAT-aware Peer Sampling Service. Indeed, in real networks a majority of nodes lie behind NATs and firewalls and cannot be contacted directly, which may have a strong impact on the operation of applications, including the gossip-based protocols we presented in this document. In a similar way to the Peer Sampling Service protocols [28] presented in Section 3, Nylon provides a continuous set of random peers from the network in the view of each node, but each such node is attached with the necessary information for bypassing NATs, be it by opening connections from the destination node behind the NAT or by the use of relay nodes. This is done in a purely gossip-based fashion, with nodes periodically exchanging the information about their peers and the associated contact details. Leitao et al. [69] also propose to leverage gossip-based self-organization to tackle the natural imbalance that arises in networks where some nodes are more difficult to reach than others.

Another example is given by the Dr Multicast [70] system. While IP multicast is an efficient mechanism for dissemination, it is not always available in the whole

network, and when it is, it is typically limited in the number of groups supported by the network elements. Vigfusson et al. propose to use gossip-based techniques for propagating information about the memberships and activities of multicast groups, and let a mapping that leverages the available IP multicast resources for as much of the communications as possible, and relying on point-to-point communication for the others.

7.3 Gossip-Based Networking and Security Aspects

An important aspect of large-scale systems that we did not mention yet in this introduction paper is that of the security. Indeed, large scale systems are composed of nodes that typically span over multiple administrative domains or end-users, and there is a risk of witnessing byzantine behaviors from part of the nodes.

First, several proposal have been made to handle the case of nodes wishing to bias the Peer Sampling Service, e.g., in order to favor a node over the others or isolate a part of the network. Brahms [71], the *Secure Peer Sampling* [72], and PuppetCast [73] are three examples of PSS protocols that take into account the presence of byzantine nodes wishing to bias the sampling.

The BAR (Byzantine, Altruistic, Rational) model is used by the authors of BAR gossip [74] to support byzantine and rational (selfish) nodes in a gossip-based dissemination of messages in a network. An interesting aspect of the protocol is that the selection of partners for gossip is no longer made at random (e.g., by using a PSS), but by a pseudo-random selection that keeps the properties of randomness that a PSS would typically provide. StarblabIT [75] is another proposal of an intrusion-tolerant gossip-based dissemination protocol. It allows to ensure the complete and authenticated dissemination of messages within a group; while making sure that external attackers cannot hamper the dissemination (by guaranteeing reliability, authenticity and consistency).

Finally, gossip-based mechanisms are used in Whisper [76] for implementing confidential group membership and communications in an autonomous and self-organizing manner. Nodes leverage gossip-based overlay construction principles to exchange alternative paths, that are used as anonymizing routes between members of the group, without the need for a trusted third party for protecting group members' identifies and communications. In the context of authentication, Yan et al. [77] propose to use gossip-based networking to implement group key distribution.

Gossip-based networking has also been leveraged in [78] to implement decentralized failure detection in a large-scale setting. Guo et al. propose to implement garbage collection using gossip in [79].

8 Conclusion

In massive-scale distributed systems, rigid control of the system's operation is inapt and can lead to severe bottlenecks and operational deficiencies. In the face of –inevitable for such systems– continuous errors and failures, a self-adaptive

scheme with self-configuring and self-healing properties is more appropriate for working around the errors. Such properties are known as self-* properties.

In this paper we advocated the importance of gossiping protocols in providing self-* properties to distributed systems of very large scale. We classified gossiping protocols in two categories. Peer sampling protocols, serving as sources of randomly selected nodes from the whole network, and the standard ("traditional") gossiping protocols for serving specific application needs (data dissemination, node clustering, data aggregation, etc.).

We identified a number of properties in gossiping protocols that make them particularly attractive for large scale distributed systems. They are remarkably robust and tolerant to faults, even to failures of very large scale. They distribute the load across many nodes, leading to load balanced networks. Gossip-based systems are inherently symmetric, in the sense that no node has special responsibility at a particular task, therefore no fault at any single node may harm the smooth operation for the whole community. They are very scalable, to millions of nodes, and they disseminate, aggregate, or cluster data at exponential speed. Last but not least, they are very simple.

By visiting a set of representative gossiping protocols, we gave insight to the conceptual framework of epidemics, within which elegant, simple, and robust algorithmic building blocks for large-scale systems can be proposed.

References

1. Maymounkov, P., Mazières, D.: Kademlia: A peer-to-peer information system based on the XOR metric. In: Druschel, P., Kaashoek, M.F., Rowstron, A. (eds.) IPTPS 2002. LNCS, vol. 2429, pp. 53–65. Springer, Heidelberg (2002)
2. Loo, B.T., Huebsch, R., Hellerstein, J.M., Shenker, S., Stoica, I.: Enhancing p2p file-sharing with an internet-scale query processor. In: Proc. 30th International Conference on Very Large Data Bases, VLDB (August 2004)
3. Chawathe, Y., Ratnasamy, S., Breslau, L., Lanham, N., Shenker, S.: Making gnutella-like p2p systems scalable. In: SIGCOMM 2003: Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, pp. 407–418. ACM Press, New York (2003)
4. Felber, P., Kropf, P., Leonini, L., Luu, T., Rajman, M., Rivière, E.: Collaborative ranking and profiling: Exploiting the wisdom of crowds in tailored web search. In: Eliassen, F., Kapitza, R. (eds.) DAIS 2010. LNCS, vol. 6115, pp. 226–242. Springer, Heidelberg (2010)
5. Castro, M., Druschel, P., Kermarrec, A.M., Nandi, A., Rowstron, A., Singh, A.: Splitstream: high-bandwidth multicast in cooperative environments. In: SOSP 2003: Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles, pp. 298–313. ACM Press, New York (2003)
6. Zhang, C., Jin, H., Deng, D., Yang, S., Yuan, Q., Yin, Z.: Anysee: Multicast-based peer-to-peer media streaming service system. In: Proceedings of the Asia-Pacific Conference on Communications(APCC05), Pernth, Western Australia (October 2005)
7. Rowstron, A., Kermarrec, A.M., Castro, M., Druschel, P.: SCRIBE: The design of a large-scale event notification infrastructure. In: Crowcroft, J., Hofmann, M. (eds.) NGC 2001. LNCS, vol. 2233, pp. 30–43. Springer, Heidelberg (2001)

8. Voulgaris, S., Rivière, E., Kermarrec, A.M., van Steen, M.: Sub-2-sub: Self-organizing content-based publish subscribe for dynamic large scale collaborative networks. In: Proceedings of IPTPS 2006: 5th International Workshop on Peer-to-Peer Systems, Santa Barbara, USA (February 2006)
9. Bhagwan, R., Savage, S., Voelker, G.M.: Understanding availability. In: Kaashoek, M.F., Stoica, I. (eds.) IPTPS 2003. LNCS, vol. 2735, Springer, Heidelberg (2003)
10. Stutzbach, D., Rejaie, R.: Understanding churn in peer-to-peer networks. In: IMC 2006: Proceedings of the 6th ACM SIGCOMM on Internet Measurement, pp. 189–202. ACM Press, New York (2006)
11. Kermarrec, A.M., van Steen, M.: Gossiping in distributed systems. SIGOPS Oper. Syst. Rev. 41(5), 2–7 (2007)
12. Demers, A., Greene, D., Hauser, C., Irish, W., Larson, J., Shenker, S., Sturgis, H., Swinehart, D., Terry, D.: Epidemic algorithms for replicated database maintenance. In: PODC 1887: Proceedings of the Sixth Annual ACM Symposium on Principles of Distributed Computing, pp. 1–12. ACM Press, New York (1987)
13. Birman, K.P., Hayden, M., Ozkasap, O., Xiao, Z., Budiu, M., Minsky, Y.: Bimodal multicast. ACM Transactions on Computer Systems 17(2), 41–88 (1999)
14. Eugster, P.T., Guerraoui, R., Kermarrec, A.M., Massoulié, L.: Epidemic information dissemination in distributed systems. Computer 37, 60–67 (2004)
15. Gupta, I., Kermarrec, A.M., Ganesh, A.: Efficient epidemic-style protocols for reliable and scalable multicast. In: Proceedings of the 21st Symposium on Reliable Distributed Systems (SRDS 2002), p. 180 (2002)
16. Khambatti, M., Ryu, K., Dasgupta, P.: Push-pull gossiping for information sharing in peer-to-peer communities. In: Proceedings of International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA), pp. 1393–1399 (June 2003)
17. Eugster, P.T., Guerraoui, R., Handurukande, S.B., Kouznetsov, P., Kermarrec, A.M.: Lightweight probabilistic broadcast. ACM Transactions on Computer Systems 21(4), 341–374 (2003)
18. Vogels, W., van Renesse, R., Birman, K.: The power of epidemics: robust communication for large-scale distributed systems. SIGCOMM Comput. Commun. Rev. 33(1), 131–135 (2003)
19. Kermarrec, A.M., Massoulié, L., Ganesh, A.J.: Probabilistic reliable dissemination in large-scale systems. IEEE Transactions on Parallel and Distributed Systems 14(3), 248–258 (2003)
20. Renesse, R.V., Birman, K.P., Vogels, W.: Astrolabe: A robust and scalable technology for distributed system monitoring, management, and data mining. ACM Trans. Comput. Syst. 21(2), 164–206 (2003)
21. Locher, T., Meier, R., Schmid, S., Wattenhofer, R.: Push-to-pull peer-to-peer live streaming. In: Pelc, A. (ed.) DISC 2007. LNCS, vol. 4731, pp. 388–402. Springer, Heidelberg (2007)
22. Rhea, S.C.: Opendht: a public dht service. PhD thesis, Berkeley, CA, USA, AAI3211499 (2005)
23. Patel, J.A., Gupta, I., Contractor, N.: Jetstream: Achieving predictable gossip dissemination by leveraging social network principles. In: NCA 2006: Proceedings of the Fifth IEEE International Symposium on Network Computing and Applications, pp. 32–39. IEEE Computer Society, Washington, DC, USA (2006)
24. Serbu, S., Rivière, E., Felber, P.: Network-friendly gossiping. In: Guerraoui, R., Petit, F. (eds.) SSS 2009. LNCS, vol. 5873, pp. 655–669. Springer, Heidelberg (2009)

25. Fernandess, Y., Fernández, A., Monod, M.: A generic theoretical framework for modeling gossip-based algorithms. SIGOPS Oper. Syst. Rev. 41(5), 19–27 (2007)
26. Allavena, A., Demers, A., Hopcroft, J.E.: Correctness of a gossip based membership protocol. In: PODC 2005: Proceedings of the Twenty-Fourth Annual ACM Symposium on Principles of Distributed Computing, pp. 292–301. ACM Press, New York (2005)
27. DeCandia, G., Hastorun, D., Jampani, M., Kakulapati, G., Lakshman, A., Pilchin, A., Sivasubramanian, S., Vosshall, P., Vogels, W.: Dynamo: amazon's highly available key-value store. SIGOPS Oper. Syst. Rev. 41(6), 205–220 (2007)
28. Jelasity, M., Voulgaris, S., Guerraoui, R., Kermarrec, A.M., van Steen, M.: Gossip-based peer sampling. ACM Transactions on Computer Systems 25(3) (August 2007)
29. Jelasity, M., Kowalczyk, W., van Steen, M.: Newscast Computing. Technical Report IR-CS-006, Vrije Universiteit Amsterdam, Department of Computer Science, Amsterdam, The Netherlands (November 2003)
30. Voulgaris, S., Jelasity, M., van Steen, M.: A robust and scalable peer-to-peer gossiping protocol. In: Moro, G., Sartori, C., Singh, M.P. (eds.) AP2PC 2003. LNCS (LNAI), vol. 2872, pp. 47–58. Springer, Heidelberg (2004)
31. Voulgaris, S., Gavidia, D., van Steen, M.: Cyclon: Inexpensive membership management for unstructured p2p overlays. Journal of Network and Systems Management 13(2) (June 2005)
32. Erdös, P., Rényi, A.: On the evolution of random graphs. Publications of the Mathematical Institute of the Hungarian Academy of Sciences 5, 17–61 (1960)
33. Barabasi, A.L., Albert, R.: Emergence of scaling in random networks. Science 286, 509–512 (1999)
34. Barabasi, A.L., Albert, R.: Statistical mechanics of complex networks. Reviews of Modern Physics (2002)
35. Barabasi, A.L.: LINKED: The New Science of Networks. Perseus Books Group (2002)
36. Ganesh, A.J., Kermarrec, A.M., Massoulié, L.: SCAMP: Peer-to-peer lightweight membership service for large-scale group communication. In: Crowcroft, J., Hofmann, M. (eds.) NGC 2001. LNCS, vol. 2233, pp. 44–55. Springer, Heidelberg (2001)
37. Leitão, J., Pereira, J., Rodrigues, L.: Hyparview: a membership protocol for reliable gossip-based broadcast. In: Proceedings of the 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, Edinburgh, UK, pp. 419–428 (June 2007)
38. Jelasity, M., Babaoglu, O.: T-man: Gossip-based overlay topology management. Engineering Self-Organising Systems 1(15) (2005)
39. Jelasity, M., Montresor, A., Babaoglu, O.: T-man: Gossip-based fast overlay topology construction. Computer Networks 53(13), 2321–2339 (2009)
40. Voulgaris, S., van Steen, M.: Epidemic-style management of semantic overlays for content-based searching. In: Cunha, J.C., Medeiros, P.D. (eds.) Euro-Par 2005. LNCS, vol. 3648, pp. 1143–1152. Springer, Heidelberg (2005)
41. Voulgaris, S.: Epidemic-Based Self-Organization in Peer-to-Peer Systems. PhD thesis, Vrije Universiteit Amsterdam (2006)
42. Lua, E.K., Crowcroft, J., Pias, M., Sharma, R., Lim, S.: A survey and comparison of peer-to-peer overlay network schemes. IEEE Communications Survey and Tutorial (March 2004)

43. Kleinberg, J.: The small-world phenomenon: An algorithmic perspective. In: Proceedings of the 32nd ACM Symposium on Theory of Computing, Portland, OR, USA, pp. 163–170 (May 2000)
44. Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world' networks. Nature 393, 440–442 (1998)
45. eDonkey (no date), http://www.edonkey2000.com
46. Guerraoui, R., Handurukande, S.B., Huguenin, K., Kermarrec, A.M., Fessant, F.L., Rivière, E.: GosSkip, an efficient, fault-tolerant and self organizing overlay using gossip-based construction and skip-lists principles. In: P2P 2006: Proceedings of the Sixth IEEE International Conference on Peer-to-Peer Computing, pp. 12–22. IEEE Computer Society, Cambridge (2006)
47. Pugh, W.: Skip lists: A probabilistic alternative to balanced trees. Communication of the ACM 32(10), 668–676 (1990)
48. Harvey, N.J.A., Jones, M.B., Saroiu, S., Theimer, M., Wolman, A.: Skipnet: A scalable overlay network with practical locality properties. In: The Fourth USENIX Symposium on Internet Technologies and Systems (USITS 2003), Seattle, WA (2003)
49. Beaumont, O., Kermarrec, A.M., Rivière, E.: Peer to peer multidimensional overlays: Approximating complex structures. In: Tovar, E., Tsigas, P., Fouchal, H. (eds.) OPODIS 2007. LNCS, vol. 4878, pp. 315–328. Springer, Heidelberg (2007)
50. Bonnet, F., Kermarrec, A.M., Raynal, M.: Small-world networks: From theoretical bounds to practical systems. In: Tovar, E., Tsigas, P., Fouchal, H. (eds.) OPODIS 2007. LNCS, vol. 4878, pp. 372–385. Springer, Heidelberg (2007)
51. Milgram, S.: The small world problem. Psychology Today 2, 60–67 (1967)
52. Patel, J.A., Rivière, E., Gupta, I., Kermarrec, A.M.: Rappel: Exploiting interest and network locality to improve fairness in publish-subscribe systems. Computer Networks 53(13), 2304–2320 (2009)
53. Jelasity, M., Kermarrec, A.M.: Ordered slicing of very large-scale overlay networks. In: P2P 2006: Proceedings of the Sixth IEEE International Conference on Peer-to-Peer Computing, pp. 117–124. IEEE Computer Society, Cambridge (September 2006)
54. Fernandez, A., Gramoli, V., Jimenez, E., Kermarrec, A.M., Raynal, M.: Distributed slicing in dynamic systems. In: Proceedings of the International Conference on Distributed Computing Systems (ICDCS 2007), IEEE Computer Society Press, Toronto (June 2007)
55. Gramoli, V., Vigfusson, Y., Birman, K., Kermarrec, A.M., van Renesse, R.: Slicing distributed systems. IEEE Transactions on Computers – Special Issue on Autonomic Network Computing (IEEE TC) 58(11), 1444–1455 (2009)
56. Jelasity, M., Montresor, A., Babaoglu, O.: Gossip-based aggregation in large dynamic networks. ACM Trans. Comp. Syst. 23(3), 219–252 (2005)
57. Jelasity, M., Montresor, A.: Epidemic-Style Proactive Aggregation in Large Overlay Networks. In: 24th Int'l Conf. on Distributed Computing Systems, pp. 102–109 (2004)
58. Montresor, A., Jelasity, M., Babaoglu, O.: Robust aggregation protocols for large-scale overlay networks. In: DSN 2004: Proceedings of the 2004 International Conference on Dependable Systems and Networks (DSN 2004), p. 19. IEEE Computer Society, Washington, DC, USA (2004)
59. Kempe, D., Dobra, A., Gehrke, J.: Gossip-based computation of aggregate information. In: FOCS 2003: Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science, p. 482. IEEE Computer Society, Washington, DC, USA (2003)

60. Kowalczyk, W., Vlassis, N.: Newscast EM. In: Advances in Neural Information Processing Systems (NIPS), vol. 17, MIT Press, Cambridge (2004)
61. Voulgaris, S., van Steen, M.: An epidemic protocol for managing routing tables in very large peer-to-peer networks. In: Brunner, M., Keller, A. (eds.) DSOM 2003. LNCS, vol. 2867, pp. 41–54. Springer, Heidelberg (2003)
62. Kostoulas, D., Psaltoulis, D., Gupta, I., Birman, K., Demers, A.: Decentralized schemes for size estimation in large and dynamic groups. In: NCA 2005: Proceedings of the Fourth IEEE International Symposium on Network Computing and Applications, pp. 41–48. IEEE Computer Society, Washington, DC, USA (2005)
63. Massoulié, L., Merrer, E.L., Kermarrec, A.M., Ganesh, A.: Peer counting and sampling in overlay networks: random walk methods. In: PODC 2006: Proceedings of the Twenty-Fifth Annual ACM Symposium on Principles of Distributed Computing, pp. 123–132. ACM Press, New York (2006)
64. Merrer, E.L., Kermarrec, A.M., Massoulié, L.: Peer to peer size estimation in large and dynamic networks: A comparative study. In: Proceedings of the 15th IEEE International Symposium on High Performance Distributed Computing, Paris, France, pp. 7–17 (June 2006)
65. Baldoni, R., Beraldi, R., Quema, V., Querzoni, L., Tucci-Piergiovanni, S.: Tera: topic-based event routing for peer-to-peer architectures. In: DEBS 2007: Proceedings of the 2007 Inaugural International Conference on Distributed Event-Based Systems, pp. 2–13. ACM Press, New York (June 2007)
66. Matos, M., Nunes, A., Oliveira, R., Pereira, J.: Stan: Exploiting shared interests without disclosing them in gossip-based publish/subscribe. In: Proc. of IPTPS 2010: 9th International Workshop on Peer-to-Peer Systems, San Jose, CA, USA (2010)
67. Anceaume, E., Gradinariu, M., Datta, A.K., Simon, G., Virgillito, A.: A semantic overlay for self-* peer-to-peer publish/subscribe. In: Proceedings of the International Conference on Distributed Computing Systems, ICDCS 2006 (June 2006)
68. Kermarrec, A.M., Pace, A., Quema, V., Schiavoni, V.: Nat-resilient gossip peer sampling. In: Proceedings of the 2009 29th IEEE International Conference on Distributed Computing Systems. ICDCS 2009, pp. 360–367. IEEE Computer Society, Washington, DC, USA (2009)
69. Leitão, J., van Renesse, R., Rodrigues, L.: Balancing gossip exchanges in networks with firewalls. In: Proceedings of the 9th International Conference on Peer-to-Peer Systems. IPTPS 2010, p. 7. USENIX Association, Berkeley (2010)
70. Vigfusson, Y., Abu-Libdeh, H., Balakrishnan, M., Birman, K., Burgess, R., Li, H., Chockler, G., Tock, Y.: Dr. multicast: Rx for data center communication scalability. In: Proceedings of Eurosys 2010, Paris, France (April 2010)
71. Bortnikov, E., Gurevich, M., Keidar, I., Kliot, G., Shraer, A.: Brahms: Byzantine resilient random membership sampling. Computer Networks 53(13), 2340–2359 (2009)
72. Jesi, G.P., Montresor, A., van Steen, M.: Secure Peer Sampling. Elsevier Computer Networks - Special Issue on Collaborative Peer-to-Peer Systems 54(12), 2086–2098 (2010)
73. Bakker, A., van Steen, M.: Puppetcast: A secure peer sampling protocol. In: Proc. of the European Conference on Computer Network Defense (EC2ND 2008), Dublin, Ireland, pp. 3–10 (December 2008)
74. Li, H., Clement, A., Wong, E., Napper, J., Alvisi, L., Dahlin, M.: Bar gossip. In: Proc. of 7th Symposium on Operating System Design and Implementation, OSDI 2006 (2006)

75. Kihlstrom, K.P., Elliott, R.S.: Performance of an intrusion-tolerant gossip protocol. In: Proc. of the 21st IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS), Cambridge, MA, USA (November 2009)
76. Schiavoni, V., Rivière, E., Felber, P.: Whisper: Middleware for confidential communication in large-scale networks. In: Proc. of ICDCS 2011: 31st Int'l Conference on Distributed Computing Systems, Minneapolis, Minnesota, USA (2011)
77. Yan, Y., Ping, Y., Yi-Ping, Z., Shi-Yong, Z.: Gossip-based scalable and reliable group key distribution framework. In: InfoSecu 2004: Proceedings of the 3rd International Conference on Information Security, pp. 53–61. ACM Press, New York (2004)
78. van Renesse, R., Minsky, Y., Hayden, M.: A gossip-style failure detection service. In: IFIP (ed.) Proc. of Middleware, the IFIP International Conference on Distributed Systems Platforms and Open Distributed Processing, The Lake District, UK, pp. 55–70 (1998)
79. Guo, K., Hayden, M., van Renesse, R., Vogels, W., Birman, K.P.: Gsgc: An efficient gossip-style garbage collection scheme for scalable reliable multicast. Technical report, Cornell University, Ithaca, NY, USA (1997)

Author Index