

Web of Things as a Framework for Ubiquitous Intelligence and Computing

Tharam S. Dillon, Alex Talevski, Vidyasagar Potdar, and Elizabeth Chang

Digital Ecosystems and Business Intelligence Institute (DEBII)
Curtin University of Technology
GPO Box U1987
Perth, WA, 6845, Australia
{Tharam.Dillon,Alex.Talevski,Vidyasagar.Potdar,
Elizabeth.Chang}@cbs.curtin.edu.au

Abstract. In the last few years the web has evolved from just being a source of information to a platform for business applications. In parallel to this, wireless sensor network technologies have evolved to a stage where they are capable of connecting to the web. This amalgamation of technologies referred to as “Web of Things”, has created new opportunities for intelligent application development. However, the “Web of Things” has brought interesting challenges like - efficiently utilizing online sensors, sensor composition for just in time application development and others that require urgent attention. In this paper, we propose a conceptual framework and reference architecture for embedding the notions of Ubiquitous Intelligence and Computing within the “Web of Things” and empowering their use in a broader context.

1 Introduction

The last few years have seen an explosive growth in the use of the web for many aspects of business and human endeavor. The web has evolved from its very early days where it was purely used for information dissemination, to application deployment, the read-write web with web 2.0 and progressively the provision of semantics on the web. In parallel with this development has been that of ubiquitous computing with the deployment of sensors and minimal computing devices on the person or in the environment of interest. These two trends have initially developed essentially on their own separate tracks. More recently there has been an increasing trend towards connecting these sensors and wearable devices to the internet leading to the formulation of the notion of the so called “Internet of Things”. However, to be truly of the web rather than just being connected through it is necessary to have effective mechanisms for:

1. Locating these sensors.
2. Having a sharable interface description that is consistent with the developing web. Here it is important to remember that these sensors will not only have information but also some very limited computing capability.
3. Invoking and composing sensor data and functions.
4. Defining an architectural framework.

A realization of this has led to the definition of the “Web of Things“ which provides a framework for addressing these issues.

In this paper, we propose a conceptual framework and reference architecture for embedding the notions of Ubiquitous Intelligence and Computing within the “Web of Things” and empowering their use in a broader context. It also outlines some existing application areas.

2 Service-Oriented Architecture (SOA) vs. Resource Oriented Architecture (ROA)

In order to solve the UCDs and sensor integration problems, the supporting IT systems need to be simple and malleable. Research [1] has outlined the requirement for a service-based approach. Service-based approaches achieve loose coupling among interacting systems where disparate components may interact using a common interaction protocol and certain architectural and protocol constraints. By abstracting a component’s internals through an interface, components become well isolated and standardized. Such architectures define component location, integration, management, monitoring and security in a straightforward way. Tailorable systems may then be rapidly adapted to satisfy specialized and evolving requirements. Tailorable system architectures require specific tools and significant technological support. We have identified the following categories and features that must be investigated when developing architectures for the “Web of Things” to allow composition, Interaction and collaboration and these are:

- **Context independence** - Identifying the clear decompositions in such a system is a key to developing a hierarchical system architecture where systems are built from building blocks.
- **Service Model or a Resource Model** – A service or Resource model ensures that disparate devices which reside on different platforms, can interoperate.
- **Accessibility** - Services and Resources must be accessible by clients that are implemented using different technologies, and are distributed over a network.
- **Data Exchange** – A data exchange model is a communication protocol that enables components to interact and transfer data in a standardized way.
- **Location Transparency** - To achieve location transparency and seamless interoperability, a piece of wrapping middleware code lies between components that make interactions transparent.
- **Contracts** – An interaction contract guarantees that an interface exists, and provides its advertised services.
- **Plug and Play** - The use of stable, published interfaces represented in a shareable standardized representation enables assembly or integration of applications from disparate sources. In this way, components can be reconfigured, added, removed, or replaced after system deployment.
- **Automation** – The use of macros and scripts that are implemented to facilitate dynamic discovery, interaction and integration of reusable components at runtime [2][3].

In order to perform UCD and sensor service composition and application reconfiguration within a dynamically changing operation environment, UCD and sensor software and middleware systems need to be simple and malleable. A service based approach for sensor systems facilitate dynamic system self configuration, adaptation and adhoc network routing protocols. Service-based approaches may achieve loose coupling among interacting sensors and devices. Tailorable architectures allow generic sensors and devices to be rapidly adapted to satisfy the specialized, rapidly changing, unclear and / or evolving system requirements through hierarchical and iterative service composition. They provides a means for the straightforward creation and modification of application solutions from provided sensor and device services. Using a reconfigurable approach as a basis for the creation and modification of sensor applications, it possible to construct, customize, integrate and evolve software solutions as a response to production requirements. Such a service framework and platform are directed at increasing software quality and performance, reducing software development and maintenance costs and easing runtime software modification.

In exploring reference architectures we review the potential of SOA and ROA as potential candidates as a reference architecture for the “Web of Things”.

Service-Oriented Computing (SOA) in general and Web services in particular have recently received tremendous momentum because the encapsulation, componentization, decentralization, and integration capability provided by SOC. Web services are substantial; they provide both architectural principles and software specifications to connect computers and devices using standardized protocols across the Internet. SOA is a reference architectural style that defines a set of constraints on how service-oriented applications interact with each other in order to achieve high degree of flexibility and dynamicity [4]. Here by reference architecture [5] we mean it is the principal architecture that determines the overall software system behavior such as flexibility and dynamicity which are two important requirements for the Web of Things. As an architectural style [6] SOA, through its “publish-find-bind” triangle, achieves loose-coupling which facilitates the dynamicity and flexibility. Web services technology realizes SOA through providing one with an implementation that builds a one-to-one mapping between the SOA architecture and the implementation. For example, matchmaker in SOA is implemented as the UDDI registry/repository in Web services. Service capability information is described using the WSDL (Web Services Description Language) documents. Service consumers and providers can be realized as SOAP (Simple Object Access Protocol) clients and servers that communicate via SOAP messages over the Internet. From the technological viewpoint, a Web service is “an interface that describes a collection of operations that are network accessible through standardized XML messaging” [7]. The standardized interface differentiates Web services from other distributed object technologies (e.g. CORBA, DCOM, etc.), in which interfaces are not defined using open standards and tools. Standardized interfaces represent ‘a shared understanding towards implementation independence’; a key element for building loosely-coupled distributed applications through interoperability. This implementation independence has made Web services a technology of choice for building complex applications in a loosely-coupled manner across distributed environments. Two types of Web service interfaces are common: the traditional Big Web services (WSDL, SOAP, and UDDI) and the RESTful Web services which treat

HTTP as a semantically capable application protocol rather than just as a transport protocol which is the case in SOAP.

Resource-Oriented Architecture (ROA) was first proposed in [8], in which four key concepts were defined for a ROA: Resources, URL, Representation, and Links. The key properties of ROA compliant software architecture is: addressability, statelessness, connectedness, and a uniform interface. A useful paradigm here is termed as Resource-Oriented Computing (ROC). ROC is an emerging simplified paradigm to perform computational tasks. In ROC, everything (e.g. processes, data, business logic, etc) is treated as a logic resource that is an abstract set of information. ROC provides a high-level view towards computation, which abstracts away from low-level data and objects situated at the physical layer. One of the unique uses of identity (URI) in ROC is that URI is used for both identity and for representing a functional program. This property is of crucial importance for the Web of Things.

3 Architectural Framework for the Web of Things

The “Web of Things” uses a standardized protocol HTTP as an application protocol instead of just as a transport protocol to provide connection of sensors with the internet. This relies on the so-called “Restful Services” discussed above. Within this framework, the “Web of Things” refers to the next generation of network and web where each and every object on the globe will be identifiable, interactive and a part of an adhoc network. The “Web of Things” relies on the numerous electronic devices or sensors, which form a collaborative system. The important aspect of these Ubiquitous Computing Devices (UCD) is that they can measure quantities and events and provide information on these and also carryout some very limited logic operations or computations. We choose to represent a UCD as a resource. We use the following definition from [9] namely a resource, is “*a thing that has a universal identifier, a name, may have reasonable representations and which can be said to be owned*”. Note this definition allows a Web services resource to be universally identifiable through the Web, thus permitting discovery, selection, and invocation desirable of a Web services identified by the URL on the Web. We distinguish between a *Web service resource* and a *Web resource* in that the former one does not necessarily encode information about the “state” of the resource i.e it is stateless whilst the *Web resource* may have state even though it does not maintain any state information this is encoded in the representation sent back to the resource requester.

This “Web of Things” thus provides an integrated framework for Ubiquitous Intelligence and Computation. The “Web of Things” envisions a network implementation (which is frequently wireless) for its sheer ubiquity. “Web of Things” brings enormous challenges on the conceptual and technical levels which must be resolved to produce a successful “Web of things “ strategy for ubiquitous intelligence and computation and these include:

- Abstractions for sensor and event representations
- Compositions of sensors to deliver particular information

- Semantics for compositions of sensors and events
- Robust wireless devices and networks
- The storage, structure and retrieval of the tsunami of wireless sensor data
- User interfaces for visualizing, configuring, monitoring and controlling such networks and their outputs
- Middleware to seamlessly integrate and tailor adhoc devices and the “Web of Things” network
- Convergence of technologies to deliver truly synchronized wireless voice, video and data.
- Wireless network and device security

When these are combined with wearable devices and sensors associated with the immediacy of a person, it leads to the concept of “Human Space Computing”. Human Space Computing (HSC) refers to an end user or consumer interacting with technology-based system(s) or embedded computer devices through interactive protocols or modalities such as audio, video, hearing, feeling, vision, touching, sound, voice, vibrating, reading, writing, tagging, blogging, binding etc.. all human sensations [10].

The distinction between this and Human Computer Interaction (HCI) lies in the definition of what the Human interacts with; system or computers, - what are the differences between the two; the interaction protocols and what do they include. It is obvious that Human Space Computing requires a much larger scale of emphasis and automation on ‘system’ and ‘interaction’ and it is where the challenges are.

4 Abstractions for Ubiquitous Computing Devices (UCD) and Event Representations

The first challenge with the deployment of the “Web of Things” is to abstract the UCD information for event representation and information representation. As mentioned earlier “Web of Things” envisions many items on this globe would transmit data (such as identity, temperature, location, etc.) from UCDs such as sensor data on a regular basis which give us information on an object or the state of a particular ambient environment. This data requires processing to generate information which can then be used for event identification and representation. As both individual UCDs and individual events are really UCD instances and event instances we need abstractions for these. Abstracting UCD and event representations is a challenging task that requires frameworks that can deal with this problem in its entirety. At the first level as mentioned above, UCDs can be represented as resources which associate a unique identifier that allows them to be located in the “Web of Things” space. At the next level, it is important to develop the notion of

1. UCD classes which represent collections of UCD instances with particular properties. e.g. a temperature UCD monitor and alarm class. This class could consist of several instances. To allow for mobility, this class would have both location and time as two amongst other properties.
2. Event classes which represent collections of event instances with particular properties. e.g. an intrusion event detection class which could consist of

several event instances. Event classes too would have location and time (or at least position in a sequence in time) as two amongst other properties. This need for developing event abstractions has also been put forward by Ramesh Jain in his “Web of Events” keynote delivered in the 2008 Keynote at the Semantics, Knowledge and Grid Conference.

In addition, there will be relationships between these classes, which will allow for representation of generalization/specialization, composition and associations.

It would be useful to record the sensor data in a portable format like XML or UCDML so that it can seamlessly integrate with high level applications. This will allow us to meet the following challenges:

- Manage how raw sensor data is kept, maintained and exported by disparate sources.
- Interpreting events associated with particular sensor configurations and outputs.
- Transforming system level knowledge from distinct sensors or UCDs into higher-level management applications.
- The fact that the abstract level representation of the UCDs or sensors is implemented will allow substitution of new UCDs or sensors as technology progresses by keeping the interfaces standardized.

5 Compositions of UCDs and Events to Address Complex Requirements

The second challenge with the deployment of the “Web of Things” is to provide an ability to compose data from multiple sensors based upon the requirements laid by a particular application scenario. This has its own set of challenges that again require a dedicated framework on how information from multiple sensors is composed and correlated for meeting the QoS requirements of a particular scenario.

Here we will need a decomposition technique which will allow for the decomposition of complex functionality that is required into lower level resources which can then be met by specific UCDs or Events. Here we distinguish between

1. Decomposition into aggregations of simpler resources.
2. Decomposition of complex events into aggregations of simpler events.

It has to be clearly understood that this decomposition requires one to specify the items in the aggregation as well as their dynamic sequence of execution or arrangement. To model the dynamics, it has often been suggested that workflows specified in a language such as BPEL would be appropriate. An alternative, particularly for resources (but also suitable for services particularly Restful services) is the use of Mashups which allow easier configuration of these workflows. Our preferred approach is to utilize Mashups for this representation of the workflows.

The “Web of Things” envisions that all the items would be tagged with sensors, with or without having a prior idea of how this sensed information would be utilized. For example, a set of things could be tagged with a generic sensor that could monitor

external phenomenon. Now it is up to the middleware or different application to decide how this sensor data can be used to suit application specific requirements. However, many challenges in composing sensor data remain. The composition of events remains a challenging issue with a focus on sequencing them to produce a composite event.

6 Semantics for Compositions of Sensors and Events

The third challenge with the deployment of the “Web of Things” is to provide semantics during the sensor composition phase so that automatic sensor discovery, selection and composition can happen. Semantics here is the study of the meaning of Things (Resources that represent UCDs or sensors) and Events. It represents the definitions of the meaning of elements, as opposed to the rules for encoding or representation. Semantics describe the relationship between the syntactical elements and the model of computation. It is used to define what entities mean with respect to their roles in a system. This includes capabilities, or features that are available within a system.

Ontologies are formal, explicit specifications of a shared semantic conceptualization that are machine-understandable and abstract models of consensual knowledge. We merge Gruber’s [11], Borst’s [12], and Studer’s [13] definitions of an ontology as a basis for our discussion here. Using such an ontology, it is possible to define concepts through uniquely identifying their specifications and their dynamic and static properties. Concepts, their details and their interconnections are defined as an ontology specification. Ontology compositions are typically formed from many interconnected ontology artifacts that are constructed in an iteratively layered and hierarchical manner. It is necessary to use a graphical representation of ontologies at a higher level of abstraction to make it easier for the domain experts to capture the semantic richness of the defined ontology.

The ontology-based architecture is grounded on the notion of a base ontology, sub-ontologies [14] and/or commitments [15]. These sub-ontologies are used as independent systems (in functionality) for the various decentralized users. This would allow for a very versatile and scalable base of operation. Numerous sub-ontologies provide custom portals for different expert groups to access information and communicate with other groups [16][17].

They are concerned with all processes of sensor and device operation. They define concepts, abstractions, relationships and interactions as domain concepts and instantiations for manual or automated reasoning. These ontologies signify information which evolves to reflect the operational environment. Reaching such a formal consensus of understanding is of benefit in an environment that deals with self-contained, distributed and heterogeneous sensors and devices. In general the following system properties can be defined:

- **Information and Communication** - refers to the basic ability to gather and exchange information between the parties involved.
- **Integrability** - relates to the ability of sensors and devices from different sources to mesh relatively seamlessly and the ability to integrate these sensors to build the final solution in a straightforward way.

- **Coordination** - focuses on scheduling and ordering tasks performed by these parties.
- **Awareness and Cooperation** - refer to the implicit knowledge of the operation process that is being performed and the state of the system.

The integration of a number of ontologies can be used to augment the functionality of any framework namely;

- **Process Ontology** - develops a representation of the different types of processes involved in an operation.
- **Resource Ontology** - provides a representation of all the resources involved in the operation activity at the different levels of granularity. As each large granularity the composite resource is likely to consist of sub resources itself, it is important that these are all represented within the ontology.
- **Outputs Ontology** - represents the different types of outputs or units that will result from the operation activity. They involve classification of the outputs and its features.

Each defined ontology conceptually represents the perceived domain knowledge through its concepts, attributes, taxonomies, relationships, and instances for operations.

Two approaches to the representation of semantics here would be (1) use of ontologies (2) lightweight semantic annotations. The choice of a method that would be suitable will depend on the circumstances. For representing the knowledge in a given domain ontologies and for adding semantics to individual resources annotation may suffice. A special challenge here is developing ontologies for events.

7 Robust Wireless Devices and Networks

A key element of the “Web of Things” is the ability to deploy sensors with flexibility and mobility. An important technology here is WSN technology. Recent advances in wireless technology have enabled the development of wireless solutions capable of robust and reliable communication in various environments. International standards such as the IEEE 802.11a/b/g/n for wireless local area networks and the IEEE 802.15.4 for low-rate wireless personal area networks, as well as numerous RFID (radio-frequency identification) specifications, have enabled applications such as wireless networking, sensing, monitoring, control, and asset tracking. Such wireless sensing technologies have the potential to be beneficial in both domestic and industry applications in a number of ways. Introducing these technologies can contribute to;

- The monitoring of the working and living environment, systems and devices
- Reduced installation, integration, operation and maintenance costs
- Speedy installation and removal
- Mobile and temporary installations
- Up-to-date information services are available at anytime, anywhere
- Enhanced visualization, foresight, forecasting and maintenance schedules
- Determine certain patterns and characteristics of our everyday lives

- Safe living and working environment and optimized processes
- Effective and efficient operation

The challenge with the deployment of the “Web of Things” is to provide robust wireless sensing devices and extremely reliable communication networks. This is the most challenging task so far as there are many parameters that need to be considered. The key challenges when we consider deploying “Web of Things” using wireless devices can be categorized as follows:

- Restricted size, shape, construction and certification.
- Make do with limited processing power, memory, storage, battery consumption and screen real-estate resources.
- Self contained. When possible, devices should generate their own power or, contain battery packs with extended battery life of many years to reduce maintenance.
- Operate in a difficult wireless environment both in terms of radio noise and obstructions but also where certain restrictions on such radio devices are present (such as flammable areas).
- Operate in a hostile areas where environmental conditions may be difficult.
- Embedded platforms.
- Implement complex network algorithms with real-time requirements and adaptive routing protocols.
- Contribute in a simplified ad-hoc and multi-hop network.
- Seamlessly integrate with existing IT solutions.
- Self re-configurable, dynamic and adaptive.
- Provide services within a dynamically changing system environment.
- Exhibit fault tolerance and recovery (self-healing, robust and reliable).
- Based on open, international standards.
- Operate in the unlicensed portions of the frequency spectrum.
- Low maintenance.
- Implement strict encryption, transmitter authentication and data consistency validation.
- Clearly defined operational reliability and availability of the wireless network within the operational environment.

8 The Tsunami of Information in the “Web of Things”

IP traffic will nearly double every two years through to 2012 [18]. Therefore, total IP traffic will increase by a factor of six from 2007 to 2012. High-definition video, audio and other high-resolution, high update streaming data such as that produced by cctv cameras and sensors over high-speed connections will be the main driver behind the 46%pa IP growth in the coming years. Such streaming data may now account for approximately one-quarter of all Intranet traffic. Streaming data grew from 12 percent in 2006 to 32 percent in 2008 [18]. The rapid growth of the use of high resolution devices on our networks has produced a significant challenge in dealing with such a tsunami of data.

We may utilize 1000s of streaming sensors and cameras to monitor certain conditions. These sensors will become an integral part of the safe and high performance operations. These sensors produce significant amounts of unstructured and unclassified data that is simply used to momentarily visualize conditions. It is essential that sensor systems have sufficient network bandwidth, processing power and storage capacity to cater for the tsunami of data that is experienced.

9 Application Areas

There are many application areas of the above such as;

- Oil, gas, resources and manufacturing industries
- Vehicle, road, traffic and transportation
- Human space technologies, smart appliances and wearable devices
- Social Networking
- Convergence

For reasons of space we will only discuss one of these below.

9.1 Oil, Gas, Resources and Manufacturing Industries

Oil, gas and resource production plants use a variety of sensors that produce a visual representation of the just-in-time plant operational state. Such sensors are particularly important when the platform functions are operated remotely. The use of sensor data supports the move from “Decision Support” operations to true “Remote Operations. Many technologies have not yet become mature to successfully facilitate remote operations. A requirement for large remotely operated oil, gas and resources plants is for the engineers and technical personnel to be remotely distributed, while plant operators are located on-site. This physical distance can become a crucial issue if remote interactions are incomplete and/or ambiguous; problematic and/or potentially dangerous events may occur. Therefore, the need for a formal remote operation strategy has considerable significance. Operations ontologies may foster domain knowledge, process knowledge, standards and procedures, plant architecture, components, composition and systems, staff and resources, projects and schedules, stakeholders, and supply chain (vendors, suppliers). There is even ongoing research projects in the oil and gas industry looking into the development of completely unmanned production platforms. For such a scheme to be successful, it is essential that cctv cameras, microphones, communications, control systems and other data that is transmitted is synchronized and converged.

Existing sensor output handling has many deficiencies in information processing and presentation. System operators are often presented with large amounts of data that must be processed quickly and decisively. However, the amount of information required to be digested is quite large when crucial decisions need to be made within a constrained timeframe. This can leave the system operator suffering from cognitive overload when decisions have the greatest impact. This problem requires sources of sensor data, methods of obtaining that sensor data, and methods of interpreting the sensor data and producing a relevant message to be investigated. Expert system

approaches have been quite useful in the areas of sensor output processing. There are a number of problems that cannot be practically solved and result in inadequate results because of several difficulties. The continuing increase in the size and complexity of oil, gas and resources platforms and production has led to monitoring and control involving a significantly extensive and complex data set. This is further complicated in emergency situations when rapid decisions are required. Sensor output processing is a problem which has been investigated for quite some time. Various problems associated with such processing techniques such as system complexity, output relationships, response time, reasoning, incomplete and incorrect data have been investigated. Expert system approaches have been useful in the areas of sensor output processing. However, several issues have not been addressed. The integration of Expert Systems and Artificial Neural Networks to solve many sensor output processing problems have proved to be quite successful in addressing a large set of problems in the power system [19]. However, the use of such techniques in oil, gas and resource production are seldom discussed or effectively employed.

10 Conclusion

In this paper we explored the use of the “Web of Things” as a framework for Ubiquitous Intelligence and Computing. Several issues that need to be resolved were identified and some solutions were proposed. Several open questions that must be addressed were also identified for future research.

References

- [1] Schutte, R.: SOA is changing software (2003), <http://www4.gartner.com/resources/111900/111987/111987.pdf>
- [2] Chang, E., Annal, D., Grunta, F.: A Large Scale Distributed Object Architecture – CORBA & COM for Real Time Systems. In: Proceedings of the Third IEEE Conference on Object Oriented Real Time Distributed Systems, Newport Beach, California (March 2000)
- [3] Annal, D.: NetCaptain Client Architecture. In: System Development Document. Philips Public Communications Pty. Ltd, Australia (1999)
- [4] Dillon, T.S., Wu, C., Chang, E.: Reference Architectural Styles for Service-Oriented Computing. Springer, Heidelberg (2008)
- [5] Barber, K.S., Graser, T.J., Holt, J.: Evolution of Requirements and Architectures: An Empirical-based Analysis. In: 1st International Workshop on Model-based Requirements Engineering, San Diego, CA (2001)
- [6] Perry, D.E., Wolf, A.L.: Foundations for the Study of Software Architecture. Software Engineering Notes, ACM SIGSOFT 17(4), 40–52 (1992)
- [7] Kreger, H.: Fulfilling the web services promise. Communications of the ACM 46(6), 29–34 (2003)
- [8] Richardson, L., Ruby, S.: RESTful Web Services. O’Reilly, Sebastopol (May 2007)
- [9] Booth, D., Haas, H., McCabe, F., Newcomer, E., Champion, M., Ferris, C., Orchard, D.: Web services architecture, W3C Working Group, Hewlett-Packard, Software AG, IBM, Fujitsu Labs of America, Iona, BEA Systems, W3C (2004)

- [10] Chang, E., Dillon, T., Calder, D.: Human system interaction with confident computing. The mega trend. In: Proceedings of Conference on Human System Interaction, Poland (2008)
- [11] Gruber, T.R.: A translation approach to portable ontology specification. In: Knowledge Acquisition (1993)
- [12] Borst, W.: Construction of Engineering Ontologies, Centre of Telematica and Information Technology, University of Twente, Enschede, The Netherlands (1997)
- [13] Studer, R., Fensel, D.: Knowledge Engineering: Principles and Methods. IEEE Transactions on Data and Knowledge Engineering (1998)
- [14] Jarrar, M., Meersman, R.: Formal Ontology Engineering in the DOGMA Approach. In: Proc. Intl. Conf. on Ontologies, Databases, and Applications of Semantics, pp. 1238–1254 (2002)
- [15] Spynus, P., Meersman, R., Mustafa, J.: Data Modeling versus Ontology Engineering. In: Proc. 21st ACM SIGMOD Intl. Conf. on Management of Data, pp. 14–19 (2002)
- [16] Klein, M., Fensel, D., Kiryakov, A., Ognyanov, D.: Ontology versioning and change detection on the web. In: Gómez-Pérez, A., Benjamins, V.R. (eds.) EKAW 2002. LNCS, vol. 2473, pp. 197–212. Springer, Heidelberg (2002)
- [17] Wouters, C., Dillon, T.S., Rahayu, W., Chang, E., Meersman, R.: A Practical Approach To The Derivation of A Materialized Ontology View. In: Taniar, D., Rahayu, W. (eds.) Web Information Systems, pp. 191–226. Idea Group Press, Melbourne (2004)
- [18] Cisco, Visual Networking Index – Forecast and Methodology 2007–2012 (2009), <http://www.cisco.com/>
- [19] Dillon, T.S., Morsztyn, K., Phua, K.: Short-Term Load Forecasting Using Adaptive Pattern Recognition and Self-Organising Techniques. In: Proceedings of the Power System Computation Conference, Cambridge, UK (March 1975)