A Grid Based Distributed Cooperative Environment for Health Care Research

Felipe Maia[1], Rafael Araújo[1], Luiz Carlos Muniz[1], Rayrone Zirtany[1], Luciano Coutinho[1], Samyr Vale[1], Francisco José Silva[1], Pierpaolo Cincilla[2], Ikram Chabbouh[2], Sébastien Monnet[2], Luciana Arantes[2], and Marc Shapiro[2]

[1] UFMA - Avenida dos Portugueses, s/n
65085-580 São Luís, MA, Brazil
{lrc,samyr,fssilva}@deinf.ufma.br
[2] UPMC - Paris, France
firstname.lastname@lip6.fr

Abstract. Providing a distributed cooperative environment is a challenging task, which requires a middleware infrastructure that provides, among others, management of distributed shared data, synchronization, consistency, recovery, security and privacy support.

In this paper, we present the ECADeG project which proposes a layered architecture for developing distributed cooperative environments running on top of a desktop grid middleware that can encompass multiple organizations. We also present a particular cooperative environment for supporting scientific research focused on the health domain. It uses the services supplied by the ECADeG architecture in order to allow researchers to share access to multiple institutions databases, visualize and analyze data by means of data mining techniques, edit research documents cooperatively, exchange information through forums and chats, etc.. Such a rich cooperative environment helps the establishment of partnerships between health care professionals and their institutions.

1 Introduction

A distributed cooperative environment provides a common user interface that enables collaborative tasks in a specific context. In such environments, many features can be provided, such as asynchronous and synchronous communication mechanisms, a repository for shared resources, or concurrent (synchronous) editing of content. Nevertheless, building a distributed cooperative environment is a challenging task, since several issues must be taken into consideration, such as managing the communication between the distributed entities, data replication, detection and resolution of update conflicts, privacy and security, provision of a WYSIWIS (What You See Is What I See) interface, and performance. As a consequence, a distributed cooperative environment is usually built on top of a middleware infrastructure that provides a set of services for hiding the complexity of the distributed environment.

The ECADeG project is a joint initiative of the Federal University of Maranhão (UFMA) Distributed Systems Laboratory, in Brazil, and the Regal Team, a joint research group of LIP6 Laboratory at University Pierre et Marie Currie (UPMC)

J. Weber and I. Perseil (Eds.): FHIES 2012, LNCS 7789, pp. 142–150, 2013.

and INRIA, France. ECADeG stands for "Enabling Collaborative Applications for Desktop Grids" and the project aims at the design and implementation of a middleware infrastructure to support the development of collaborative applications, and its evaluation through a case study in the health care domain. We established a formal partnership with the UFMA University Hospital (HU-UFMA), whose health care professionals provide the necessary medical background for the project development.

The ECADeG collaborative environment has particular concerns about security and privacy in order to deal with sensitive data such as patient and healthcare information. The challenge is to follow the set of defined legal standards for medical organizations along with the specific requirements of the collaborative environment. Besides the above challenge, the platform also addresses several key challenges such as sharing data and computing resources.

This paper presents the current status of the ECADeG project. It describes the two main building blocks used as the foundation for the development of the software infrastructure proposed in ECADeG: the InteGrade, an grid middleware for desktop grids; and Telex, a middleware that facilitates the construction of collaborative applications providing optimistic sharing of documents across a network of computers. The paper also describes a preliminary view of ECADeG middleware architecture, organized as a set of layers running on top of the InteGrade grid middleware and presents the applications that comprise the cooperative environment for supporting scientific research focused at the health domain.

2 Background: InteGrade and Telex

The ECADeG middleware is being developed having as its foundation an application execution environment based on grid computing, the InteGrade middleware [1], and a middleware for supporting the development of distributed collaborative applications called Telex [2], both described in this Section.

The InteGrade project[1] is a multi-university effort to build a robust and flexible middleware for opportunistic grid computing. By leveraging the idle computing power of existing commodity workstations and connecting them to a grid infrastructure, InteGrade enables the execution of computationally-intensive parallel applications that would otherwise require expensive cluster or parallel machines.

The basic architectural unit of an InteGrade grid is a cluster, a collection of machines usually connected by a local network. Clusters can be organized in a hierarchy, enabling the construction of grids with a large number of machines.

Currently, the InteGrade middleware offers a choice of programming models for computationally intensive distributed parallel applications, MPI (Message Passing Interface), and BSP (Bulk Synchronous Parallel) applications. It also offers support for sequential and bag-of-tasks applications.

Since opportunistic grid environments are highly prone to failures, special care was taken to circumvent application execution disruptions. InteGrade provides a task-level fault tolerance mechanism based on checkpointing, which periodically saves the process' state in stable storage during the failure-free execution time [3].

[1] Homepage: `http://www.integrade.org.br`

Upon a failure, the process restarts from the latest available saved checkpoint, thereby reducing the amount of lost computation. InteGrade includes a portable application-level checkpointing mechanism for sequential, bag-of-tasks, and BSP parallel applications written in C. For MPI parallel applications, it provides a system-level checkpointing mechanism based on a coordinated protocol.

Concerning the management of application data, which includes the application binaries, input and output data, InteGrade's OppStore component provides a reliable distributed data storage using the free disk space from shared grid machines. The system is structured as a federation of clusters and is connected by a Pastry peer-to-peer network [4,5].

Telex is a generic platform that eases the development of collaborative applications. It allows application programmers to concentrate on core functionalities, by taking care of the data distribution, replication and consistency issues. Telex supports optimistic sharing over a large-scale network of computers.

Telex implements an optimistic replication approach in which updates are made locally, and are then propagated to each remote site when a communication channel is established. Update propagation is transparent to the user, and data consistency is ensured by a reconciliation protocol which runs in the background (off the critical path).

Telex is application independent although application aware. Applications need to formalize their concurrency semantics by identifying the shared data, the actions that can be made on the data and the constraints between these actions. When an end-user interacts with the application, the latter translates the operations into actions and constraints then transmits them to Telex. Based on the received local and remote actions and constraints, Telex computes sound schedules and sends them to the application to be executed. A sound schedule is a sequence of actions that satisfy the application constraints.

Schedules are computed from the Action-Constraint Graph (ACG), a replicated, dynamic graph. Actions are the nodes of the graph, and constraints the edges and arcs. A schedule is a conflict-free sub-graph.

Telex sites may generate different sound schedules from the same set of actions and constraints. A Telex module called the replica reconciler makes the sites agree on a common schedule to apply and thus achieve (eventual) mutual consistency.

ECADeG uses Telex services to manage data consistency and synchronization of distributed collaborative applications.

3 The ECADeG Project

The first aim of ECADeG project is to develop a grid-based middleware platform for the execution of distributed collaborative applications. The second one is to validate the middleware with a real distributed cooperative application, targeting the support for multi-institutional research projects in the health care domain.

The health care cooperative research environment will be based on data provided by the AGHU platform (Aplicativo de Gestão para Hospitais Universitários - Management Application for University Hospitals)[2] currently under development

[2] http://aghu.mec.gov.br/

by the Brazilian Ministry of Education and Culture (MEC), which implements a unified management model to be adopted by Brazilian's federal university hospitals. The AGHU database is designed to store all the data concerning patients and their care (consultations, image-based examinations, hospitalizations, surgeries, prescriptions, etc.). Some Brazilian university hospitals are already running the first available modules of AGHU on their own servers. However, each hospital runs its own copy of the AGHU platform due to the fact that each institution is administratively independent and is responsible (a trustee) for the health data that is internally generated. Therefore, the access of health data maintained by an institution by other hospitals or physicians is carefully controlled. Our middleware solution will integrate all the AGHUs installations, which would allow the university hospitals to share data and information cooperatively, as well as other computing resources, such as processor power for performing computationally intensive tasks, and specialized peripherals in a safe and controlled environment. Therefore, we will be able to provide a rich collaborative environment which helps to establish partnerships between health care professionals and their institutions.

Figure 1 shows an overview of the ECADeG architecture which consists of four main layers: Execution Environment, Core Services, Applications, and Security.

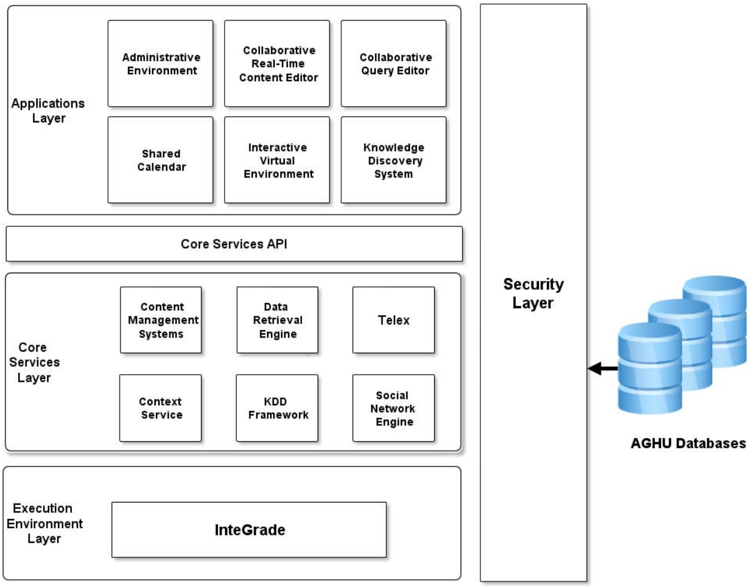
Applications Layer
Administrative Environment
Collaborative Real-Time Content Editor
Collaborative Query Editor
Shared Calendar
Interactive Virtual Environment
Knowledge Discovery System
Core Services API
Core Services Layer
Content Management Systems
Data Retrieval Engine
Telex
Context Service
KDD Framework
Social Network Engine
Execution Environment Layer
InteGrade
Security Layer
AGHU Databases

Fig. 1. ECADeG architecture

Execution Environment Layer: This layer holds the execution environment for the set of collaborative services defined by the ECADeG architecture. It is based on the InteGrade grid middlewareas described in Section 2.

Core Services Layer: The core services layer offers the set of services that supports the execution of distributed collaborative applications. The *Content Management System* provides a data space for storing any content (text, spreadsheets, data fragments retrieved from the AGHU databases, images, audio, video) shared by researchers in the context of a research project. Any stored content can be described through meta-data, whose set of attributes are defined by the application. The *Social Networking Engine* provides a complete social networking environment with bulletin boards, chats, video conferences, forums, and the tools for creating and managing users and groups. The *Data Retrieval Engine* is a tool that performs parallel data retrieval using multiple AGHU databases given an application query. It provides a SQL based API for applications and transparently manages the parallel access to multiple AGHU databases, composing a single result set as if the query was performed in a single global database. The *KDD Framework* is a component for knowledge discovery using data mining techniques to obtain relevant information from data retrieved through the Data Retrieval Engine. The framework will support classification and associations discovery tasks, since there are several possible applications of those tasks considering the medical field: (i) characterization of patients to provide further consultation, (ii) identification of successful therapies for different diseases, (iii) prediction of which patients are more likely of catch a certain disease, according to historical patient data. The *Context Service* provides a publish/subscribe interface for managing context data useful in collaborative environments, such as users availability for on-line interaction, their current activity, and location. *Telex*, as described in Section 2, allows users to create and concurrently edit shared content.

Applications access the functionalities provided by the Core Services Layer through a standardized API that comprises the Core Services API Layer. A first version of the Content, Context and Data Retrieval tool is already implemented and is being tested. We are currently defining the data mining library algorithms to be deployed in the KDD framework and porting them to the InteGrade execution environment. As a typical usage scenario that illustrates possible interactions among the several components that comprise the core services layer, consider a group or researchers working in different locations accessing a Collaborative Query Editor (CQE) developed using Telex to create a custom query to be sent to the AGHU databases of several health institutions. They use the CQE visual editor in order to choose which tables, columns and search criteria that will be used. After the query is ready, the CQE sends it to the Data Retrieval Engine which, in turn, process the query and start a process in each grid node that communicates with individual AGHU database servers in order to execute the query. Each process collects the database results and returns them to the Data Retrieval Engine that performs a merge procedure. The Data Retrieval Engine returns the final results to be presented to the researchers in the Collaborative Query Editor. They can use the Social Network engine tools to discuss and share information using chat or video-conference. If they want to process the data retrieved, e.g. for statistical analysis of the number of patients stricken by a disease in some region, the KDD framework can be used in order to run data mining algorithms to this end.

Applications Layer: In the ECADeG project we foresee the development of six main applications. The *Administrative Environment* is the tool that will be used by administrators to create, edit and delete users, define roles and enforce security policies. The *Interactive Virtual Environment* will allow users, once logged in, to interact with each other using chat, videos, forums and bulletin boards. This application uses the services provided by the Social Networking Engine and the Context Service. The *Shared Calendar* will be used for creating synchronized appointments between the platform users. Telex will be used to enforce the appointments consistency. The *Collaborative Query Editor* will create a visual query editor to the AGHU databases. Users will be able to collaboratively build queries together in real time. Telex will provide the mechanisms for maintaining the query consistency during the concurrent editing, while the Data Retrieval Engine will be used for the parallel access of the AGHU databases. The *Knowledge Discovery System* is a font-end for extracting information through data mining techniques from data retrieved using the Collaborative Query Editor. This application will be used by a KDD expert that will work together with health care researchers in order to better serve the platform users with relevant information for their research. It will use the KDD framework defined at the Core Services Layer. The *Collaborative Real-Time Content Editor* is an application that allows users to collaboratively edit shared content. In its first release, we will focus on text editing. Users will be able to concurrently edit technical reports, papers, masters and PhD thesis in real time. Again, Telex will provide the necessary tools for maintaining text consistency in the course of the concurrent editing. We already developed a first version of the ECADeG distributed collaborative environment prototype that is being validated by its final users (health care research professionals). We are also still working on the integration of the prototype functionalities with the core services layer components.

Security Layer: The security layer is transversal to all the other ECADeG layers and is based on a model that comprises a set of security policies that take into account several aspects of a collaborative environment, such as privacy and confidentiality, the organization of collaborative processes, data sharing between organizations with different administrative domains, context information and notifications of presence [6]. In addition, ECADeG security model follows the safety standards established by the Certification Manual for Electronic Registration Systems in Health (M/S-RES) [7], a document developed through a partnership between the Brazilian Society of Informatics in Health (Sociedade Brasileira de Informática na Saúde, SBIS) [3] and the Brazilian Federal Council of Medicine (Conselho Federal de Medicina, CFM)) [4]. This document describes a certification process for applications that deal with patients data, demanding that they have a complete privacy and security model, meeting the needs of users and, especially, being compliant with the legislation requirements. The ECADeG security layer includes a set of components (identity management, access control, data anonymization, auditing, and abuse report) to ensure compliance with all

[3] http://www.sbis.org.br/

[4] http://portal.cfm.org.br/

security policies established by its security model. The ECADeG development process also follows the steps defined in CLASP (*Comprehensive, Lightweight Application Security Process*) [8], a well defined process guided by a set of activities associated with a set software development roles that emphasizes security since the very initial phases of software development. Several security components are already implemented, such as the identity management, access control, and auditing. We are currently working on the provision of mechanisms that will allow the specification of privacy policies based on statements.

4 Related Work

Since the 80's a large amount of research has been done on distributed collaborative environments and their characteristics, as observed in [9], [10] and [11]. We can find in the literature proposals of distributed collaborative environments for various fields such as engineering ([12],[13], [14]), education ([15],[16], [17]) and health ([18], [19], [20]).By involving a large number of professionals from different areas for the design and use of collaborative environments, their development becomes complex and very dependent on the applicative domain.

A key distinctive feature of the ECADeG project is the proposal of an environment for parallel and cooperative access to databases that share a common data model and are distributed across several university hospitals. The proposals in [20] are different in the sense that they must define a WSDL[5] (Web Service Definition Language) interface with a restricted set of operations for each database where searches are made. This makes the extension of this platform more complex, and less transparent than the approach used by the middleware proposed in this article (see the Data Retrieval Engine, in the Section 3).

Similarly to [15], [16] and [18], the architecture of the ECADeG project is organized into a set of layers and modular services, including the use of computing grids. In this regard, an important difference between ECADeG and the cited works is that ECADeG is based on an opportunistic grid framework (InteGrade) that is able to run the services of ECADeG together with several classes of applications like regular, loosely coupled and tightly coupled applications. In [18] and [15], for instance, a dedicated computing grid is used only to execute processes that require intensive computation.

Compared to [18], [15], [19], [13], [12] and [20] where users have few forms of collaboration, the ECADeG project proposes a more complete set of tools to provide the users with a richer experience of collaborative working, for instance through chat rooms, video conferencing, collaborative editing of documents, file sharing, conducting search patterns using data mining algorithms, within a single workspace in order to assist them in their research.

Regarding the security aspect, the majority of the work in the literature fails to implement or suggest a full security model for the collaborative environment being proposed (with the exception of [18] that implements an initial privacy and security model based on [21]). As a consequence, there is a gap when the security

[5] `http://www.w3.org/TR/wsdl`

matters, as noted by [22]. With this concern in mind, the whole process of development, tests and validation of collaborative applications and infrastructures, in the ECADeG project, is governed by a security engineering process based on the Comprehensive Lightweight Application Security Process version 1.2 [8] (CLASP v1.2), and the set of rules present in the certification Manual for Electronic Registration Systems in Health described in the previous section. This results in a complete model of privacy and security, meeting the needs of users and, especially, which is in accordance with the requirements of Brazilian law.

5 Conclusion

This paper has described the ECADeG project, which proposes a layered architecture for developing distributed cooperative environments running on top of a desktop grid middleware. An ECADeG cooperative environment can encompass multiple organizations, sharing a variety of resources. We have also presented a particular cooperative environment for supporting scientific research focused at the health domain, which, by using the services supplied by the ECADeG architecture, provides applications for parallel access to databases of Brazilian university hospitals (using the AGHU platform), data visualization and analysis by means of data mining techniques, cooperative editing of research papers, interchange of information through forums and chats, among others. ECADeG project intends thus to provide a rich cooperative environment, which can help the partnerships between health care professionals and their institutions. By offering a virtual environment for cooperatively creating and sharing data and information, it mitigates the problem of physical distance of participants.

Acknowledgment. The authors would like to thank FAPEMA (State of Maranhão Research Agency) for the support of this work, grant INRIA-00114/11.

References

1. da Silva e Silva, F.J., Kon, F., Goldman, A., Finger, M., de Camargo, R.Y., Filho, F.C., Costa, F.M.: Application execution management on the Integrade opportunistic grid middleware. JPDC 70(5), 573–583 (2010)
2. Benmouffok, L., Busca, J.M., Marquès, J.M., Shapiro, M., Sutra, P., Tsoukala, G.: Telex: A semantic platform for cooperative application development. In: Conf. Française sur les Systemes d'Exploitation, CFSE (2009)
3. de Camargo, R.Y., Kon, F., Goldman, A.: Portable checkpointing and communication for BSP applications on dynamic heterogeneous Grid environments. In: SBAC-PAD 2005: The 17th International Symposium on Computer Architecture and High Performance Computing, Rio de Janeiro, Brazil, pp. 226–233 (October 2005)
4. de Camargo, R.Y., Kon, F.: Design and implementation of a middleware for data storage in opportunistic grids. In: CCGrid 2007: Proceedings of the 7th IEEE/ACM International Symposium on Cluster Computing and the Grid. IEEE Computer Society, Washington, DC (2007)

5. de Camargo, R.Y., Cerqueira, R., Kon, F.: Strategies for checkpoint storage on opportunistic grids. IEEE Distributed Systems Online 18(6) (September 2006)
6. Ahmed, T., Tripathi, A.R.: Security policies in distributed CSCW and workflow systems. IEEE Transactions on Systems Man and Cybernetics Part A Systems and Humans 40(6), 1220–1231 (2010)
7. Silveira, A.S., de Faria Leão, B., da Costa, C.G.A., Marques, E.P., Kiatake, L.G.G., Evangelisti, L.R., da Silva, M.L., da Costa Galvão, S., Takemae, T.T.R.: Manual de Certificação para Sistemas de Registro Eletrônico em Saúde (S-RES) (2009)
8. OWASP: CLASP v1.2 Comprehensive, Lightweight Application Security Process version 1.2. OWASP (2011)
9. Borghoff, U.M., Schlichter, J.H.: Computer-Supported Cooperative Work: Introduction to Distributed Applications. Springer, New York (2011)
10. Grudin, J.: CSCW: history and focus. IEEE Computer 27(5), 19–26 (1994)
11. Ahmed, T., Tripathi, A.R.: Security policies in distributed CSCW and workflow systems. IEEE Transactions on Systems Man and Cybernetics Part A Systems and Humans 40(6), 1220–1231 (2010)
12. Zhao, Y., Shi, X.: Collaborative computational chemical grid based on CGSP. In: Proceedings of the 2007 IFIP International Conference on Network and Parallel Computing Workshops, NPC 2007, pp. 199–202. IEEE Computer Society, Washington, DC (2007)
13. He, F., Han, S.: A method and tool for human-human interaction and instant collaboration in CSCW-based cad. Computers in Industry 57(8-9), 740–751 (2006)
14. Fan, L., Zhu, H., Bok, S.H., Kumar, A.S.: A framework for distributed collaborative engineering on grids. Computer-Aided Design 4, 353–362 (2007)
15. Jiang, J., Zhang, S., Li, Y., Shi, M.: CoFrame: a framework for CSCW applications based on grid and Web services, p. 577. IEEE (2005) Number 90412009
16. Li, Y., Yang, S., Jiang, J., Shi, M.: Build grid-enabled large-scale collaboration environment in e-learning grid. Expert Systems with Applications 31(4), 742–754 (2006)
17. Chen, J., Xiong, Z., Zhang, X.: Research on a Novel E-Learning Architecture Integrated Grid Technology, pp. 94–97. IEEE (2008)
18. Brussee, R., Porskamp, P., van den Oord, L., Rongen, E., Bloo, H., Erren, V., Schaake, L.: Integrated health log: Share multimedia patient data. In: ICME, pp. 1593–1596 (2005)
19. Lu, X.L.: System design and development for a CSCW based remote oral medical diagnosis system. In: IEEE (ed.) International Conference on Machine Learning and Cybernetics, vol. 6, pp. 3698–3703 (2005)
20. Phung, H.M., Hoang, D.B., Lawrence, E.: A novel collaborative grid framework for distributed healthcare. In: CCGRID, pp. 514–519 (2009)
21. Baumer, D., Earp, J.B., Payton, F.C.: Privacy of medical records: IT implications of HIPAA. SIGCAS Comput. Soc. 30(4), 40–47 (2000)
22. Hongxue, X., Fucai, W., Hong, Z., Mingtong, X.: A security architecture model of CSCW system. In: Management and Service Science, pp. 1–4. IEEE (2010)