



Newsletter #7: January 2006

Deliverable D 12.2.9

Prepared for the European Commission
under Contract No. NoE IST-507838 as a deliverable from

Task: 12.2: Networking and Dissemination
Date of issue: 31/01/2006
Version: 1.0
Distribution: public

In this Issue:

- 1 – Editorial
- 2 – Efficient embedding of information and knowledge into collaborative learning applications
- 9 – Research Project Focus: LaCOLLA
- 15 – Technology Watch
- 17 – News
- 20 – When What Where

Editorial

Welcome to the seventh issue of the Kaleidoscope Learning GRID SIG newsletter and first issue of 2006.

Starting a new year is always the most suitable moment to make our own resolutions for the whole year. One of them should be to keep and increase if possible the current successful activity within the GRID SIG. Another one is to enhance our efforts, abilities and resources to understand the complexities of the learning processes occurring in our day-to-day classrooms either physical or virtual.

We really believe that the right way to keep these New Year's resolutions is to share our knowledge, experiences and results achieved from research with others. To this end, from the Open University of Catalonia in Barcelona, we would like to share one of our lines of research related to the GRID SIG's interests. We expect this generates an interesting feedback and a beneficial interchange of ideas and experiences on this issue's topic.

In particular, we dedicate this issue to generic platforms, middleware and frameworks to develop collaborative learning applications that focuses on both the analysis of the interaction generated during the collaboration and the provision of general functionalities involved in any collaborative learning application. Moreover, we are interested in those platforms that can support Grid technology so as, among other benefits, to speed up the provision of the knowledge acquired from the students' interaction.

We think, on the one hand, that extending the concept of reusability in developing new learning applications can not but make these applications

more accessible to all. On the other hand, constantly providing the relevant actors involved in the collaborative activities with useful knowledge in terms of awareness, feedback and monitoring may greatly enhance the effectiveness of the learning experience.

In Research Project Focus section we present another important part of our research by showing the current status of a middleware called LaCOLLA. This is a fully decentralized P2P middleware for building collaborative applications that provides general purpose collaborative functionalities and thus simplifying the process of incorporating collaborative functionalities to any application. LaCOLLA is prepared to support the basic features of the collaborative learning process in a decentralized, autonomous, and self-sufficient fashion.

Technology Watch section shows an overview of other existing platforms, middleware and frameworks for the construction of collaborative learning applications on the Grid.

News section reports the determination of the e-Learning community about adopting standards for improving learning through the use of technology. Others announcements and related events occurring recently are also reported.

Finally, When-What-Where section notifies the call for papers of the main workshops and conferences that will be held in this first term of the year.

We wish the best for this 2006 to all readers !

Enjoy your read.

Santi Caballé

Learning GRID SIG Member

Efficient Embedding of Information and Knowledge into CSCL Applications

This study aims to explore two crucial aspects of collaborative work and learning: the importance of enabling CSCL applications, on the one hand, to capture and structure the information generated by group activity and, on the other hand, to extract the relevant knowledge in order to provide learners and tutors with efficient awareness and support as regards collaboration. To this end, we first identify and define the main types of information generated in on-line group activity and then propose a process for efficiently embedding this information and the knowledge extracted from it into CSCL applications for awareness and feedback purposes. The conceptual model proposed finally gave rise to the design and implementation of a CSCL generic platform, called the Collaborative Learning Purpose Library (CLPL), which serves as a basis for the systematic development of collaborative learning applications and for providing full support to the mentioned process of knowledge management.

1. Introduction

When developing CSCL environments that support online collaborative learning, several issues must be taken into account in order to ensure full support to the online learning activity. One such key issue is interaction data analysis, a core function for the support of coaching and evaluation in CSCL environments. It relies on information captured from the actions performed by participants during the collaborative process [1], [2]. The efficient embedding of this information and of the extracted knowledge into CSCL applications sets the basis for enhancing monitoring, awareness [3] and feedback [4] to achieve a successful learning process in collaborative environments.

CSCL applications are characterized by a high degree of user-user and user-system interaction and hence are generating a huge amount of event data. This event information can be easily collected and automatically processed and analyzed by computers as a data source to extract relevant knowledge of the collaboration.

Therefore, the event information management is the cornerstone in this context, aiming at

achieving three main goals: (i) provide an analysis of the group's information at three levels, namely collaborative learning outcome, group functioning and scaffolding, by obtaining and classifying the necessary information gathered from the collaborative activity into these three essential types of categories; (ii), given that the large amount of information generated during online group activity may need much time to be processed, an effective way to collect, analyze and present this information is required; (iii) efficiently embed the information and knowledge obtained into CSCL applications so as to both facilitate tutors to monitor the learning activity and constantly provide group members with as much awareness and feedback as possible.

In order to achieve these goals, we first propose a conceptual model for data analysis and management that identifies and classifies the many kinds of indicators (variables) that describe collaboration and learning into the above-mentioned three high-level potential categories of effective collaboration. Then, we enter a process that, as a first step, collects and classifies the event information generated by the group activity according to these indicators. For efficiency purposes, this information may then be structured in a way that facilitates its later processing and analysis. The last stage of this process consists of interpreting the analysis outcomes and communicating the knowledge extracted to the group members for awareness and feedback purposes as well as to the tutors to effectively track the collaborative learning process.

The structuring and classification of the event information into the three high-level collaborative processes as well as the identification of potential mid- and low-level indicators that measure and evaluate each process can contribute and facilitate the building of a portable, general and reusable collaborative learning ontology for the representation, learning and inference of knowledge about each collaborative process. This allows the design of effective computational models that reflect as accurately as possible task performance, individual and group behavior, interaction dynamics, members' relationships and group support.

As a consequence, a generic, robust, reusable platform for the systematical construction of CSCL applications is provided. This platform, called the Collaborative Learning Purpose Library (CLPL), translates the conceptual model and its indicators into a computational model of collaborative learning interaction and can be used to embed information and knowledge into CSCL applications in an efficient manner.

The paper is organized as follows. We first present a conceptual model that identifies and classifies the three main categories of information generated in group activity. In Sect. 3 we show a process to

collect and analyze this information generated and present the extracted knowledge to the interested actors. This process and the categories defined in the conceptual model are translated into a computational model based on a CSCL generic platform presented in Sect. 4. We conclude in Sect. 5 with some comments and point out ongoing and further work.

2. A Conceptual Model for Managing Group Activity Information

The conceptual model we propose aims at modeling different aspects of interaction and thus at helping all the actors involved understand the outcomes of the collaborative process. To this end, we classify group activity information into three generic categories of activity: the members' contributing behavior to the task (the outcome of collaboration), the functioning of the group (the interaction processes underlying the collaborative learning activities, such as participation behavior, role playing, etc.), and individual and group scaffolding (social support and task- or group functioning-oriented help). We summarize these issues in this section (for more details, see [20]).

Collaborative Learning Product. This is the first top-level activity category featuring the production function and task performance of on-line groups. This category is characterized by the type of actions (i.e. events) that capture and describe the functional knowledge, cognitive processes and skills of the students and the group as a whole in solving problems and producing learning outcomes in a collaborative learning practice. The aim is to analyze and evaluate the individual and group effectiveness as far as task achievement concerns. To this end, this category can be measured as a qualitative and quantitative parameter by the type of user task-based actions that represent contributions which express basic and supporting active learning skills as well as perception skills.

Group Functioning. This is the second top-level activity category which is made up of the type of events that represent and are used to measure and analyze the individual and group effectiveness regarding participation and interaction behavior that facilitate the group's well-being function [6]. As a quantitative parameter, it enables the measurement of important participant contributions (in terms of specific types of user actions) which indicate skills related to: active or passive participation, well-balanced contributions and role playing, participation quality and communication flow among group members, as well as the necessary skills that facilitate and

enhance group interaction, namely active processing skills (such as task, workspace and communication processing skills). In addition, interaction behavior can also be measured as a qualitative parameter by group reflection (i.e. group and individual self-evaluation).

Scaffolding. This last top-level activity category is specified by the type of events that refer to social support among members as well as to task- or group functioning-oriented help provided to a participant who is not quite able or ready to achieve a task on his or her own. As for the former, it is related to the event information that includes actions which support and promote group cohesion, such as motivational and emotional support, conflict resolution, etc. As for the latter, it must be focused on those specific actions designated to provide effective help to the peers when they need it during the collaborative learning activities. The participants' actions aiming at getting or providing help are classified and measured according to whether they refer to the task or group functioning. These actions involve different types of social support and help services [7] that have been identified in this model.

3. The Process of Embedding Information and Knowledge into a CSCL Application

In this section, we present a process to provide a learning group with relevant knowledge extracted from learners' interaction data in CSCL applications for awareness, feedback and monitoring purposes. The aim is to greatly improve the effectiveness of the learning exercise.

The process of correctly providing information and knowledge in collaborative learning applications involves three separate, necessary steps: collection of information, analysis and presentation (see Fig. 1). The entire process fails if any one of these steps is omitted. During the first step, a structuring and classification of the generated event information is needed. This information is then analyzed in order to extract the desired knowledge. The final step is to provide users with the essential awareness and feed-back from the obtained knowledge.

3.1 Collection of Event Information

The most important issue while monitoring group activity is the collection and storage of a large amount of event information generated by the high degree of interaction among the group participants. Such a large amount of informational data may need a long time to be processed. Therefore, collaborative learning systems have to be designed in a way that classifies and pre-

structures the resulting information in order, on the one hand, to correctly collect the group activity and, on the other hand, to increase the efficiency during data processing in terms of analysis techniques and interpretations.

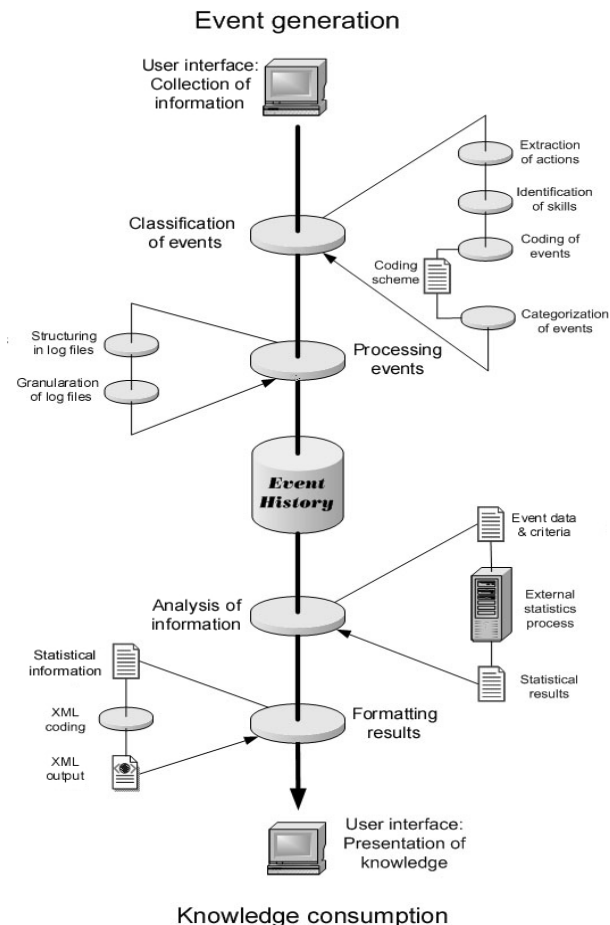


Figure 1: The process of embedding information and knowledge into CSCL applications

As shown in the previous section classification of the information is achieved by distinguishing several categories of effective collaboration. Based on this, we further categorize and specify users' particular actions according to the following criteria:

- Who is doing something? (i.e. the originator of the event).
- When did s/he do it? (i.e. timestamp).
- Where does s/he do it? (i.e. the location of the affected resource).
- What is s/he doing? (i.e. the action type and the object involved).
- Why did s/he do it? (i.e. student intentions and motivation which are captured by the indicators associated with each action; for example a

user performs the action "create document" or "edit document" in order to generate new information or refine existing information in the shared workspace).

Our aim is to provide a guideline to correctly classify the user actions on the resources during group activity. To this end, we propose a classification process in which the event information collected from the log files is handled in sequential steps consisting of extraction, identification, coding, and categorization (see Fig. 1). In particular, we first extract the specific action performed by a user on a resource (e.g. file document, debate, etc.). Second, this action is interpreted according to the type of event that was involved in (this information should be provided implicitly by the system according to the context where the action was triggered or explicitly by the user who triggered it). This provides the basic information that is used for the identification of the real intentions or skills shown by the user. Subsequently, we codify the user event taking into account both the user action and the event type. Doing so, we associate a unique code to the user skill identified in the context of the action. Finally, we categorize the user event into one of the group activity indicators defined in Sect. 2.

Due to the large amount of event information generated in CSCL applications, once this information activity has been correctly collected and classified we may come across the issue of demanding computational requirements while processing this information. In order to facilitate this step, CSCL applications may structure this information as log files in a way that takes advantage of the parallelism of a distributed environment such as Grid in order to process several files (e.g. all the groups in a classroom) at the same time and thus dramatically reduce the overall computational time to process them (see [8] and [9] for more details). As a result, it is possible for these applications to process a large volume of collaboration activity data and make the extracted information available even in real time. Note that this optimization is optional within this process of embedding information and knowledge and it is proposed for efficiency purposes only.

3.2 Data Analysis and Extraction of Knowledge

The second stage of this process consists in processing all the information previously collected and classified according to the indicators mentioned before by means of analysis techniques. There is a fair deal of research focused on the analysis of online group interaction. A detailed description of an integrated approach we followed to analyze the collaborative learning interaction can be found in [10]. As a consequence of this analysis, knowledge

is generated providing meta-cognition about the state and evolution of interaction, which enhances awareness about the efficiency of group activity, group behavior and the individual attitudes of its members in shared workspaces.

Knowledge extraction is based on criteria related to the three socio-cognitive functions that operate simultaneously during group interaction, namely production function, group well-being and member support [6] and their associated indicators. In that sense, as regards the production function, we can extract knowledge by constantly observing the members' activities (e.g. showing each group member's absolute and relative amount of contributions) or the status of shared resources. In addition, we can obtain knowledge that is relevant to individual and group well-being by exploring the communication and interaction flow among group members (such as members' motivational and emotional state, comparative studies of effective and ineffective groups and so on). Finally, knowledge can be acquired by ill-functioning situations, such as missing or insufficient contributions, lack of participation, etc., which can reveal the need for helping individual members by providing them specific scaffolding where and when this is necessary (i.e. member support).

The definition of a variety of indicators at several levels of description allows us to determine the granularity of information to be transmitted to the interested parties. In other words, based on a model of desired interaction (establishing a comparison of the current and desired state of interaction), the analysis approach detects and highlights the indicators which were not satisfied and need to be corrected by redirecting group and individual attitudes. These indicators reveal those aspects of the collaborative learning activity (task performance, group functioning, or scaffolding) that present problems and need to be corrected adequately. Thus, they set up rules and filters in order to extract and summarize only that information which refers to the malfunctioning aspect. The summarized information is finally transformed into useful knowledge that is communicated to and acquired by the group members who use it to improve the performance of the problematic aspect.

On the one hand, our approach enables group members to become aware of the progress of their peers in performing the learning exercise both at individual and group level, as well as of the extent to which other members are participating in the collaborative process as this influences their decision making [11]. On the other hand, our approach provides tutors with information about students' problem-solving behavior, group processing [12] and performance

analysis [13] for assessment and guiding purposes [17]. This approach is presented below and constitutes the last stage of the process of embedding information and knowledge into CSCL applications.

3.3 Presentation of the Knowledge Acquired

Here the problem consists in identifying the roles and needs of each learner and the tutor in every moment and being able to decide what information is required to be provided, in which granularity and how to present it. For example, the knowledge obtained from the interaction analysis should be tailored in such a way that the support provided for self-regulation or peer assessment is adapted to the role the learner plays at a particular moment. In that way, scaffolding information would be different for a learner playing a coordinator role from one that plays a software designer role. Moreover, the format used to present the information could vary from case to case.

Consequently, we proceed to define three different levels that dictate how the acquired knowledge is to be presented, that is, at what format and detail level:

Awareness level. At this level, we need to inform participants about what is going on in their shared workspace, providing information about their own actions or the actions of their peers, or presenting a view of the group interaction, behavior and performance [3]. To this end we display plane indicator values that show the state and specific aspect of the collaborative learning interaction and processes that take place. The information presented to the learner can support him/her at a meta-cognitive level.

Assessment level. At this level, we need to provide data and elements to assess the collaborative activity, so the indicators used are associated with specific weights that measure the significance of each indicator in the assessment process. As in the previous case, the information provided acts at a meta-cognitive level, giving the actors the possibility to evaluate their own actions and behavior as well as the performance of their peers and the group as a whole.

Scaffolding (or Guiding) level. Supporting participants during collaborative activities has become a main concern of current research [4], [14], [15], [16]. At this level, we need to produce information aiming at guiding, orienting and supporting students in their activity. This information is determined by the unsatisfied indicators and helps students to diagnose problematic situations and self-estimate the appropriateness of their participation in a collaborative activity as well as to counsel their peers whenever insufficient collaboration is detected.

4. A Computational Model to Embed Information and Knowledge

We propose a generic, robust, reusable component-based Collaborative Learning Purpose Library (CLPL) [9] based on the Generic Programming paradigm [18] so as to enable a complete and effective reutilization of its generic components for the construction of specific CSCL applications. We use this platform as a computational model especially for the implementation of the conceptual model of information management and the process of embedding this information and the knowledge extracted into CSCL applications as described in Sect. 2 and 3.

The CLPL is made up of five components related to user management, administration, security, knowledge management, and functionality mapping the essential needs in which any CSCL application is involved. In this context, we are especially interested in using this platform as a computational model for data analysis and management. This is mainly performed by two components, namely *CSCL Knowledge Management* and *CSCL Functionality* components, which form the core of the CLPL in the construction of CSCL applications. Due to its importance, they are described here briefly.

4.1 CSCL Knowledge Management Component

In developing our CLPL we paid special attention to event analysis and management. To this end, this component is made up of two subsystems, namely CSCL Activity Management and CSCL Knowledge Processing so as to support the first two stages of the information and knowledge embedding process. The third stage, presentation (of user awareness and feedback), is accomplished by a different component called CSCL Functionality that we will describe in the next subsection.

CSCL Activity Management Subsystem. This subsystem manages the system log files made up of all the events occurring in a certain workspace over a given period of time. It represents the automatic process of collection and classification of the event information as the source of information that is later used for the creation of the appropriate statistics. To this end, a generic *log file* is provided as a key entity that first collects all the action events generated during the group activity. This event information may then be codified following the coding scheme proposed in Sect. 3 so that it can be correctly classified according to a hierarchy of events based on the mentioned three types of activity which is also provided in this subsystem.

CSCL Knowledge Processing Subsystem. This performs the management and maintenance of statistical analysis through the generated events which have been previously collected, classified and stored in log files. It represents the analyses of the event information thus providing the necessary knowledge to control and facilitate the collaborative learning process as well as improving the general performance of the system.

To this end, a *statistics* abstraction is provided which will have the information source stored in the database along with generic criteria with the aim of performing the desired quantitative analysis of the individual and group activity and learning. To make it possible, a data structure of generic and parameterized *criteria* was also designed to classify the most usual requests for information in CSCL environments (e.g. "How many users accessed the system during a period of time?", "Which users read a document?") thus making it possible to reuse them in as many statistics as possible.

The ultimate objective of this component is to define a bottom-up analysis approach that analyses the user events in order to decode the specific actions of the users describing their interaction during the collaboration activities. The analysis aims at identifying those sequences of actions that can be used to determine typical patterns of interactions [19]. Thus, at this point in our research our objective is to identify as many best collaborative learning practices as possible, which can then be translated into typical collaborative learning patterns. Based on a model of desired interaction, the system allows us to compare the learners' real interaction processes with the typical interaction patterns in order to infer whether or not the process is effective for the learner. Furthermore, the knowledge revealed by this analysis can enhance self and peer evaluation, which in turn improves the efficiency of group activities, monitoring group behavior and the individual attitudes of its members in the shared workspace. In addition, this knowledge is useful in assisting the tutor by providing the necessary means to support and assess individual and group learning outcomes.

4.2 CSCL Functionality Component

This component, which has five subsystems in all, defines the three elemental parts involved in any form of cooperation, namely coordination, communication and collaboration [9]. Coordination involves the organization of groups to accomplish the important objectives of members such as workspace organization and group structure and planning. Collaboration lets group members share any kind of resources while communication represents the basis of the whole component since it enables coordination and collaboration to be

achieved by providing them with low-level communication support.

The final objective of this component is to provide functional support to CSCL applications in terms of group organization, resource sharing, user interaction and so on. Moreover, this component implements the last stage of the process of embedding information and knowledge into CSCL applications by presenting the knowledge generated to users in terms of immediate awareness and constant feedback of what is going on in the system. Due to this importance, we describe here the specific subsystems of this component that explicitly provide support for awareness and feedback.

CSCL Awareness Subsystem. In order to provide the essential awareness information to effectively support the three areas seen, this subsystem defines three generic entities, namely *resource state*, *user status* and *group memory*, which support the collaboration, communication and coordination respectively. Each of these abstractions acts as a vehicle so that awareness information can be classified and presented to users in the correct form depending on the type of activity involved. Thus, first, in sharing (e.g. a multi-user editor session), participants are continuously modifying the state of the shared application (e.g. writing a new text comment, deleting somebody else's sketch, etc.) and so the current application state has to be continuously propagated to users as a news warning signal. Secondly, it is essential to show the current participants' status so as to be aware of the availability of them for communication (e.g. before sending a message to others it is crucial to know whether or not they are available). Finally, the persistent storage of awareness information is needed during coordination since it allows us to access documents and data, which are commonly stored for later retrieval, and also the context in which they were created. Thus, being aware of the activities of others is essential for coordination (e.g. decision-making, group organization, social engagement, etc.).

On the other hand, for the purposes of presentation format, this subsystem defines a *flag* as a single abstraction supporting the presentation of awareness information to users through the user interface by any means: from a visual and simple sign for warning purposes to complex visual and audio effects to keep participants aware of what is happening in the group activity.

CSCL Feedback Subsystem. Feedback in CSCL collaborative learning environments is receiving a lot of attention due to its positive impact on the motivation, emotional state, and problem-solving abilities of groups in on-line collaborative learning [4]. It aims to influence group partici-

pants in a positive manner by means of a steady tracking of parameters outside the task itself (such as motivation and emotional state) and by giving a constant feedback of these parameters to the group.

Feedback goes one step further than awareness by providing exhaustive information of what is going on in the group over a long period of time (e.g. constantly showing to each group member the absolute or relative amount of the contributions of others). Furthermore, feedback may be obtained about the emotions and motivation of participants through asking them about these states. In all cases, feedback implies receiving information simultaneously both synchronously and asynchronously since the history information shown is continuously updated.

As a consequence of the complex knowledge provided to participants in form of feedback (e.g. group's member relative and absolute amount of contributions, group's members variation in motivation and emotional state during last two hours, etc) this subsystem makes a strong use of the statistical analysis and need to show the results obtained in complex graphical formats.

In this subsystem we define certain generic entities such as history, pool and diagram and functions such as sorting. These abstractions allow dynamically gathering and storing great amounts of history data and statistical results from the group activity in order to constantly update and present them to participants in the appropriate diagrammatic form (e.g. pie chart, histograms, etc.).

5. Conclusions

In this paper we have shown the importance of providing efficient knowledge and information from collaborative learning activity and how to efficiently embed them into CSCL applications as crucial aspects in the enhancement of the collaborative learning process. To this end, we have first described a conceptual model that captures and classifies three main types of information generated in the group activity by means of potential indicators of effective collaboration and secondly we presented the process of embedding this information and the knowledge extracted into CSCL applications. Then, the categories defined in the conceptual model are translated into a computational model of collaborative learning interaction by means of a generic platform called Collaborative Learning Purpose Library for the construction of collaborative learning applications which is specially used to embed information and knowledge into CSCL applications in an efficient manner.

Further work will focus on how to incorporate information retrieval and document filtering tech-

niques into the stage of information collection to automatically extract knowledge from information with a high degree of informality (e.g. to evaluate the group's well-being function) and thus making it possible to embed it into CSCL applications. We would also like to investigate how to integrate a portable, general and reusable CSCL ontology in our generic platform as a declarative representation of the knowledge embedded into CSCL applications with the aim to both describe how these systems are built and understand how real groups work.

References

- [1] Dillenbourg, P. (1999). Introduction; What do you mean by "Collaborative Learning"? In P. Dillenbourg (Ed.), *Collaborative learning. Cognitive and computational approaches* (p. 1-19). Oxford: Elsevier Science
- [2] Martínez, A., de la Fuente, P., Dimitriadis, Y. (2003) Towards an XML-Based Representation of Collaborative Action. In: Proc. of the Computer Support for Collaborative Learning, CSCL 2003, Bergen, Norway
- [3] Gutwin, C., Stark, G. and Greenberg, S. Support for Workspace Awareness in Educational Groupware. in Proceedings of the ACM Conference on Computer Supported Collaborative Learning, Bloomington, Indiana, USA October 17-20, 1995
- [4] Zumbach, J., Hillers, A. & Reimann, P. (2003). Supporting Distributed Problem-Based Learning: The Use of Feedback in Online Learning. In T. Roberts (Ed.), *Online Collaborative Learning: Theory and Practice* pp. 86-103. Hershey, PA: Idea
- [5] R. Bentley, Hosrtmann, T., Trevor, J. (1997). The World Wide Web as enabling technology for CSCW: The case of BSCW. In: *Computer-Supported Cooperative Work: Special Issue on CSCW and the Web*, vol. 6, 1997, pp. 111-134
- [6] McGrath, J.E.: Time, Interaction and Performance (TIP). *A Theory of Groups*. *Small Group Research*, 22, (1991) 147-174
- [7] Webb, N.: Testing a theoretical model of student interaction and learning in small groups. In: R. Hertz-Lazarowitz and N. Miller (Eds.), *Interaction in Cooperative Groups: The Theoretical Anatomy of Group Learning*. Cambridge Univ. Press, NY (1992) 102-119
- [8] Xhafa, F., Caballé, S., Daradoumis, Th. and Zhou, N. (2004). A Grid-Based Approach for Processing Group Activity Log Files. In: proc. of the GADA'04, Agia Napa, Cyprus
- [9] Caballé S., Xhafa, F., Daradoumis, T. and Marquès, J.M. (2004). Towards a Generic Platform for Developing CSCL Applications Using Grid Infrastructure. In: Proc. of the CLAG/CCGRID'04, Chicago, USA
- [10] Daradoumis, T., Martínez, A. & Xhafa, F. (2004). An integrated approach for analyzing and assessing the performance of virtual learning groups. In: Proc. of the 10th Int. Workshop on Groupware (CRIWG 2004), Berlin: Springer
- [11] Dillenbourg, P. (ed.) (1999): *Collaborative Learning. Cognitive and Computational Approaches*. Elsevier Science Ltd. 1-19
- [12] Kiesler, S. and Sproull, L.S. (Eds.) (1987). *Computing and change on campus*. New York: Cambridge Press
- [13] Daradoumis, T., Xhafa, F. and Marquès, J.M. (2003) Exploring Interaction Behaviour and Performance of Online Collaborative Learning Teams, 9th Int. Workshop on Groupware, CRIWG'03, France. *Lecture Notes in Computer Science*, Vol. 2806, pp. 126-134
- [14] Soller, A. Supporting Social Interaction in an Intelligent Collaborative Learning System. *Int. J. of Artificial Intelligence in Education*, 12, (2001) 40-62
- [15] Lund, K. Human Support in CSCL : what, for whom, and by whom ? In J.-W. Srijbos, P. Kirscher & R. Martens (Eds.). *What we know about CSCL in Higher Education*. pp. 167 -198. Dordrecht : Kluwer Academic Publishers, 2004
- [16] Baloian, N., Galdames, P., Collazos, C. and Guerrero, L. A model for a Collaborative Recommender System for Multimedia Learning Material. In: Proc. of the 10th Int. Workshop on Groupware (CRIWG 2004), Berlin: Springer
- [17] Ellis, C., Gibbs, S., and Rein, G. Groupware: some issues and experiences. *Communications of the ACM*, 34,1, 9-28, 1991
- [18] Caballé, S. and Xhafa, F. (2003). A Study into the Feasibility of Generic Programming for the Construction of Complex Software. 5th YRWS. In: Proc. of the GPCE/NODE'03 pp.441-446, Erfurt, Germany
- [19] Inaba, A., Ikeda, M., & Mizoguchi, R. What Learning Patterns are Effective for a Learner's Growth? - An ontological support for designing collaborative learning. In: Proceedings of the International Conference on Artificial Intelligence in Education (AIED2003), pp. 219-226, Sydney, Australia, July 20-24, 2003
- [20] Daradoumis T., Xhafa, F. and Juan Pérez, A. (2005). A Framework for Assessing Self Peer and Group Performance in e-Learning. In: *Self, Peer, and Group Assessment in e-Learning*, Roberts, T. S. (Eds). Idea Group Press, Hershey, PA, USA.

**S. Caballé, T. Daradoumis
and F. Xhafa**

Open University of Catalonia, Department of Information Sciences

Research Project Focus: LaCOLLA

Asynchronous collaborative applications and systems have to deal with complexities associated with interaction nature, idiosyncrasy of groups and technical and administrative issues of real settings. Existing solutions address asynchronous collaboration via simplified and centralized models. In this paper we present LaCOLLA, a fully decentralized middleware for building collaborative applications that provides general purpose collaborative functionality without requiring anyone to provide resources for the whole group. This helps applications to incorporate collaboration support and deal with most complexities derived from groups and its members. The implementation of LaCOLLA follows the peer-to-peer paradigm and pays special attention to the autonomy of its members and to the self-organization of its components. Resources (e.g. storage, task execution) and services (e.g. authorization) are provided by its members, avoiding dependency from third party agents or servers. We built the middleware and adapted some collaborative applications.

1 Introduction

One of the most significant benefits of the Internet has been the improvement on people's interactions and communication. E-mail, Usenet News, Web and Instant Messaging are four of the most well-known and successful examples of this. Internet has allowed the creation of asynchronous virtual communities where members interact in a many-to-many basis. Many-to-many interaction, uncommon in the physical world, has transformed the way people learn, do work together, find others with common interests and share information among them, etc. After a decade of great excitement, the pace of this transformation is slowing down because collaboration is much more than these tools, because the Internet is designed for one-to-one interaction (the Internet transport is designed for the communication between two hosts) and that applications with collaborative necessities have to deal with complexities derived from:

- *Interaction nature*: participants are dispersed, many-to-many collaboration, people participate in the collaboration at different times, the same person connecting from different locations at different times of the day (home, work, mobile).

- *Idiosyncrasy of groups*: variety of issues such as flexibility, dynamism, decentralization, autonomy of its participants, different kinds of groups (task oriented, long-term, weak commitment groups, etc), groups exist while its members participate in group activities and provide necessary resources, etc.
- *Technical and administrative issues*: guarantees for the availability of information generated in the group, interoperability among applications, security aspects (authorization, access rights, firewalls), participants belonging to different organizations or departments with different authorities that impose rules and limits to facilitate administration, internal work and individual use, etc. [1]

Development of applications that take into account all those requirements are too complex and costly, therefore collaborative applications focus only in a few key aspects while neglecting others. In that way, most of the solutions resort to simpler client/server centralized models using resources administrated by a third party (a service provider). Client/server solutions –or more generally speaking, all solutions that require some sort of centralization– impose technical, administrative and economic restrictions that interfere with the interaction nature and idiosyncrasy of groups.

In contrast, Peer-to-Peer (P2P) systems or networks are distributed systems formed only by the networked PCs of the participants. All machines share their resources: computation, storage and communication. They all act both as servers and as clients. P2P systems are self-sufficient and self-organizing, applying protocols in a decentralized way to perform search and location, and sharing the burden of object transfers. As resource provision and coordination is not assigned to a central authority, all participants have similar functionalities and there is no strict dependency to any single participant. P2P networks may be robust and attain tolerance to failures, disconnections and attacks. [2]

In this paper we present LaCOLLA, a fully decentralized P2P middleware for building collaborative applications that provides general purpose collaborative functionalities based on the resources provided by group participants only. The provision of these functionalities will avoid applications deal with most of complexities derived from groups, members working across organizational boundaries and requiring additional resources. This simplification (transparency) will help include collaborative aspects into applications in an ad-hoc manner.

LaCOLLA began as a middleware implemented following the peer-to-peer paradigm paying special attention to the autonomy of its members and to self-organization of its components. Another key aspect was that resources (e.g. storage) and services (e.g. authorization) were provided by its

members (avoiding dependency from third party agents). At this first stage, it provided support to: storage, awareness, groups, members, instant messaging and location transparency. Now we are incorporating the ability to execute tasks using computational resources provided to group. With that ability, groups will definitely evolve to become entities per se, not only gatherings or collections of members.

Having groups as units of organization and use of resources would help to change to a view of the Internet as a collection of communities: groups of individuals sharing resources among them (an individual may belong to different groups and a resource may belong to different groups). As an example, in virtual learning environments students may need to do activities in groups using some kind of software. It will be useful that any member of the group could install the software by deploying it using the computational resources available to the group. After that, any member of the group could use that software and the results will be stored on storage resources belonging and available to group.

This ability of pooling resources belonging to groups has been strongly influenced by grid systems. Grids are large-scale geographically distributed hardware and software infrastructures composed by heterogeneous networked resources owned and shared by multiple administrative organizations which are coordinated to provide transparent, dependable, pervasive and consistent computing support to a wide range of applications [3]. In contrast, our focus is on the ad-hoc creation of groups based solely on the resources provided by the participants, independently of underlying administrative organizations or external service providers.

2 Requirements for an Asynchronous Collaborative Middleware

As mentioned previously, asynchronous collaborative applications have to deal with many aspects to support collaboration. The basic requirements a middleware should satisfy to facilitate the development of this kind of applications are [4]:

- *Decentralization*: no component is responsible of coordinating other components. No information is associated to a single component. Centralization leads to simple solutions, but with critical components conditioning the autonomy of participants.
- *Self-organization of the system*: the system should have the capability to function in an automatic manner without requiring external intervention. This requires the ability of reorganizing its components in a spontaneous

manner in presence of failures or dynamism (connection, disconnection, or mobility).

- *Oriented to groups*: group is the unit of organization.
- *Group availability*: capability of a group to continue operating with some malfunctioning or not available components. Replication (of objects, resources or services) can be used to improve availability and quality of service.
- *Individual autonomy*: members of a group freely decide which actions perform, which resources and services provide, and when connect or disconnect.
- *Group's self-sufficiency*: a group must be able to operate with resources provided by its members (ideally) or with resources obtained externally (public, rent, interchange with other groups, ...)
- *Allow sharing*: information belonging to a group (e.g. events, objects, presence information, etc.) can be used by several applications.
- *Security of group*: guarantee the identity and the selective and limited access to shared information (protection of information, authentication).
- *Availability of resources*: provide mechanisms to use resources (storage, computational, etc.) belonging to other groups (public, rented, interchange between groups to improve availability, etc.)
- *Internet-scale system*: formed by several components (distributed). Members and components can be at any location (dispersion).
- *Scalability*: in number of groups, guaranteed because each group uses its own resources.
- *Universal and transparent access*: participants can connect from any computer or digital device, with a connection independent view (e.g. as a web browser).
- *Transparency of location of objects and members*: applications don't have to worry about where are the objects or members of the group. Applications use a location independent identifier and may access to different instances as people move, peers join and leave, or any other conditions change.
- *Support disconnected operational mode*: work without being connected to the group. Very useful for portable devices.

3 LaCOLLA

LaCOLLA is a middleware that follows the requirements presented in the previous section. Four main abstractions have inspired the design process of LaCOLLA: oriented to groups, all members know what is happening in the group, all members have access to latest versions of objects, and tasks can be executed using the computational resources belonging to the group. These abstractions take shape in the following functionality.

3.1 Functionality

LaCOLLA provides to applications the following general purpose functionality [4]:

- *Communication by "immediate" and consistent dissemination of events:* information about what is occurring in the group is spread among members of the group as events. All connected members receive this information right after it occurs. Disconnected members receive it during the re-connection process. This immediate and consistent dissemination of events helps applications to provide awareness to members.
- *Virtually strong consistency in the storage of objects:* components connected to a group obtain access to latest version of any object. Objects are replicated in a weak-consistent optimistic manner. Therefore, when an object is modified, different replicas of the object will be inconsistent for a while. However, LaCOLLA guarantees that, when an object is accessed, the last version will always be provided (given that events are disseminated immediately).
- *Execution of tasks:* members of a group (or the applications these members use) can submit tasks to be executed using computational resources belonging (or available) to the group. In the present version, tasks are Java classes executed locally that perform computational activities. In future versions we want to be able to deploy services that would provide services at group level. Examples of this kind of services could be a service to coordinate some dynamic and volatile aspect in a synchronous collaborative activity, a session group-level awareness service, or any other service that can provide an added value to groups and that the fact of being deployed in a centralized manner (using only computational and storage resources belonging to group) doesn't affect the decentralization, autonomy and self-sufficiency of the group.
- *Presence:* know which components and members are connected to the group.
- *Location transparency:* applications don't have to know the location (IP address) of objects or members. LaCOLLA resolves them internally (similar to domain name services like DNS).
- *Instant messaging:* send a message to a subgroup of members of the group.
- *Management of groups and members:* administrate groups and members: add, delete or modify information about members or groups.
- *Disconnected mode:* allow applications operate offline. During re-connection, the middle-ware automatically propagates the changes and synchronizes them.

3.2 Architecture

The architecture of LaCOLLA [4] is organized in five kinds of components (figure 1). Each component behaves autonomously. Each member decides to instantiate any number of the following components in the peer is using:

- *User Agent (UA):* interacts with applications (see section 3.4). Through this interaction, it represents users(group members) in LaCOLLA
- *Repository Agent (RA):* stores objects and events generated inside the group in a persistent manner.
- *Group Administration and Presence Agent (GAPA):* in charge of the administration and management of information about groups and their members. It is also in charge of the authentication of members.
- *Task Dispatcher Agent (TDA):* distributes tasks to executors. In case that all executors were busy, the TDAs would queue tasks. Also guarantees that tasks will be executed even though the UA and the member disconnects.
- *Executor Agent (EA):* Executes tasks.

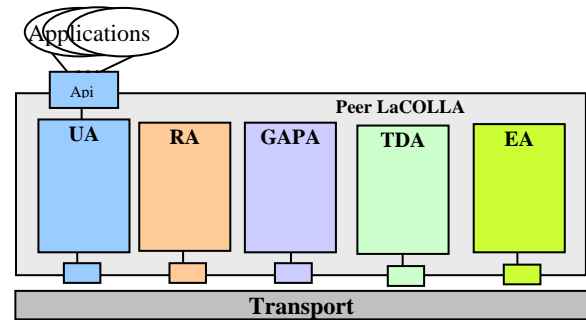


Fig. 1. Peer LaCOLLA.

Components interact one to each other in an autonomous manner. The coordination among the components connected to a group is achieved through internal mechanisms. Internal mechanisms [4] have been grouped in: events, objects, tasks, presence, location, groups, members and instant messaging. They are implemented using weak-consistency optimistic protocols [5, 6] and random decision techniques [7].

Components and mechanisms related to tasks are based on the ideas used to design JNGI [8], a decentralized and dynamic framework for large-scale computations for problems that feature coarse-grained parallelization.

Among the aspects that characterize LaCOLLA one that deserves special attention is what we have named virtual synchronism.

Virtual synchronism

LaCOLLA guarantees to applications that all events delivered to LaCOLLA will be received almost immediately (i.e. immediately or just after reconnection) by the rest of connected members. This

guarantee provides the feeling of knowing what is happening in the group while it is occurring. Disconnected members will receive the events during the re-connection process.

LaCOLLA also guarantees that the last version of all objects (based on the previous guarantee) belonging to group will be available immediately for all members.

The sum of both guarantees is what we have named virtual synchronism. Apart from the up-to-date perception that members of the group have at any moment, virtual synchronism has an interesting side effect. This side effect is very useful in an autonomous, decentralized and dynamic storage system: since all components know the location of all objects (and their replicas), components access them directly (without a resolver that informs about location of last version of objects). This allows LaCOLLA to have an autonomous and decentralized policy to handle objects and their replicas at the same time that guarantees immediate access to last versions.

3.2 Example of LaCOLLA group

Figure 2 is a snapshot of a collaborative group that uses applications connected to LaCOLLA. Each member belonging to group provides to it the resources that she/he wants. As we have said, that decision depends on the capacity and connectivity of the computer the member is using and on the degree of involvement that she/he has in the group. In this example, two members (C and D) provide all possible components (RA, GAPA, EA and TDA). Other two members (B and F) provide all components except execution components (provide RA and GAPA). Three of the members (A, E and G) provide no resources to group.

The members of the group use several applications to perform the collaborative tasks. At the moment the picture was taken, they were using an asynchronous forum, a file sharing tool and an instant messaging application. Not all members use all applications at same time.

Those applications share presence, members and group information. On the one hand, this prevents users to register to each application and also provides presence information even though they are using different applications. On the other hand, application developers don't have to worry about where the necessary information is located. LaCOLLA middleware also facilitates the sharing of information among applications (if compatible formats are used) due the fact that information, events and objects are stored in LaCOLLA storing resources (RA).

Member D (represented by discontinuous lines) is not connected to group at this moment. Even

though, her/his peer is connected to group, providing all its resources to it. That means that all generated events and some of the objects would be stored in her/his peer LaCOLLA (RA), tasks would be executed or planned using its resources (EA, TDA), or that users would be authenticated by her/his peer (GAPA), information of members and groups would be also stored in it.

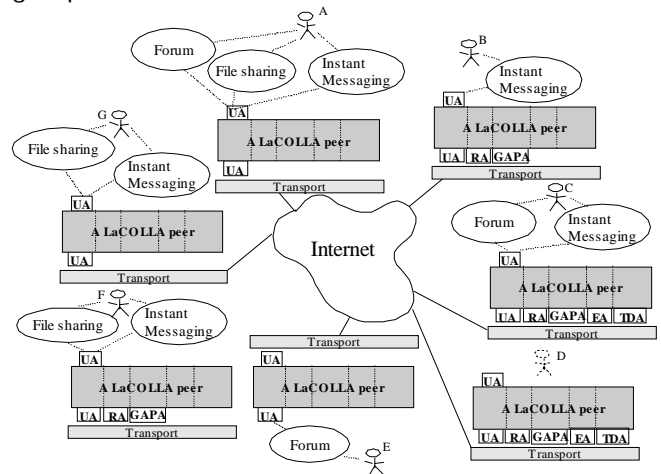


Fig. 2. Snapshot of a collaborative group that uses applications connected to LaCOLLA

An example of a group like the one presented in figure 2 could be a collaborative learning practice in a virtual university (as is UOC - Universitat Oberta de Catalunya). The learning practice could be a software development project or a case study. In those cases, a member of the group initiates the group (providing at least one RA and one GAPA components) and invites other members (who contribute with more resources and components to the group). From that point on, the group operates using the resources provided by its members. Although any member disconnects its resources or is removed as member of the group, the group will be operative. And most important, nothing will happen if the initiator of the group disconnects its resources or is removed from the group. As long as members provide resources to the group, it will exist. Whenever no member provides resources, the group would extinguish.

LaCOLLA is independent of the applications that use its functionalities. Many applications (not only the kind of applications presented in the figure) involved in a collaborative task could benefit from the general purpose collaborative functionalities that LaCOLLA provides. These applications could range from applications that only share generated information to sophisticated collaborative applications exploiting awareness information and coordinating actions (as events) of participants in the collaboration.

3.4 LaCOLLA API

LaCOLLA middleware can be found at: <http://lacolla.uoc.edu/lacolla/>. As can be seen in figure 3, the API of LaCOLLA is divided in two parts. One part is used by applications to invoke LaCOLLA functions on the UA where is connected (invoke functions in UASideApi). The other part is used by LaCOLLA to make notifications to applications (invoke functions in ApplicationSideApi). Table 1 and table 2 contain the list of functions. In the current version, Java RMI is used to publish and invoke each part of the API. However, to use the API from an application not developed in Java requires only building a module that translates parameters and results to and from Java. For instance, if the application is written in C/C++, JNI (Java Native Interface) can be used.

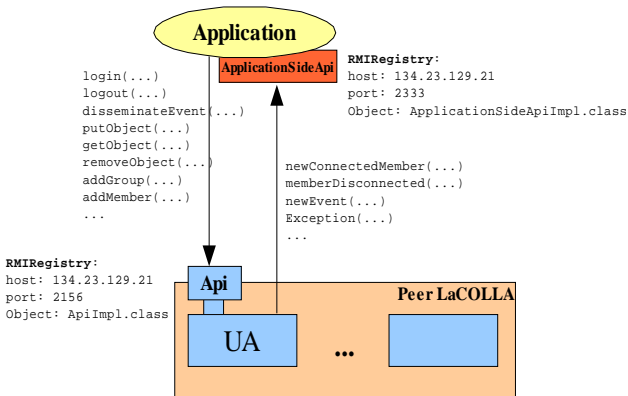


Fig. 3. LaCOLLA API has two parts. Applications use UA's API to ask LaCOLLA to do some action. The other API is provided by the applications to the UA and is used by LaCOLLA to notify events or information to applications. UAs invoke functions at ApplicationSideApi class. This class is provided with LaCOLLA middleware and must be extended by any application that wants to use LaCOLLA.

Table 1. API functions that User Agents offer to applications

Category	Function	Description
Presence	Login	Connects user to group.
	Logout	Disconnects user from group.
	whoIsConnected	Which members are connected to the group?
Events	disseminateEvent	Sends an event to all applications belonging to group.
	eventsRelatedTo	Which events have occurred to a specific object?
Objects	putObject	Stores an object in LaCOLLA.
	getObject	Obtains an object stored into LaCOLLA.
	removeObject	Removes an object stored in LaCOLLA.
Tasks	submitTask	Submits a task to be executed by computational resources belonging to group.
	stopTask	Stops a task.
	getTaskState	In which state is the task?
Instant Messaging	sendInstantMessage	Sends a message to specified members of the group.

Groups	addGroup	Creates a new group.
	removeGroup	Removes a group.
	modifyGroup	Modifies the properties of a group.
	getGroupInfo	Gets information about the properties of a group. (Look at groupInfo function)
	getGroupInfoSync	Gets information about the properties of a group in a synchronous manner. This function does not return until the operation is completed and a result is available.
Members	addMember	Creates a new member.
	removeMember	Removes a member.
	modifyMember	Modifies the properties of a member.
	getMemberInfo	Gets information about the properties of a member.

Table 2. API functions that UA invokes on applications

Category	Function	Description
Presence	newConnectedMember	Notifies that a new member has been connected.
	memberDisconnected	Notifies that a member has been disconnected.
Events	newEvent	Reception of an event occurred in the group.
Tasks	taskStopped	Notifies that the task has been stopped nicely.
	taskEnded	Notifies the ending of a task.
Instant Messaging	newInstantMessage	Reception of a new instant message.
Groups	groupInfo	Reception of the group information.
Other functions	exception	Notifies that an internal exception or anomalous situation has occurred.
	appsAlive	UA queries the state of the application. Used to know if application is alive and connected to group.

4 Validation

LaCOLLA middleware implements the functionalities presented in this paper. We also adapted and implemented some collaborative applications (an instant messaging tool, an asynchronous forum, and a document sharing tool) that benefit from LaCOLLA. These realistic applications helped us to improve the architecture and implementation of LaCOLLA. We have done limited tests with a number of ad-hoc users. All these tests confirm the usefulness of LaCOLLA. The next step is going to be to extend the functionalities of the applications we developed and use them in regular university courses at UOC.

Before implementing LaCOLLA middleware, a simulator was implemented to validate the proposed architecture under several realistic scenarios. The simulator implemented the UA, RA and GAPA components, virtual synchronism and the internal mechanisms necessary to prove that LaCOLLA behaves, as expected, in an autonomous, decentralized and self-sufficient manner.

Several experiments were done with synthetic workloads with different degrees of dynamism (failures, connections, disconnections or mobility), with different sizes of groups (from 5 to 100 members) and with different degrees of replication (number of RA and GAPA). All components were affected by dynamism.

Simulations had two phases. The first phase simulated a realistic situation. In that phase all internal mechanisms were operative. During this phase members' activity was simulated and components connected, disconnected, moved or failed. The second phase was called repair phase and only internal mechanisms were active. This second phase was used to evaluate how long LaCOLLA required achieving: a) self-organization: all connected components have consistent the information about all internal mechanisms, b) virtual synchronism: all connected components have all events and have consistent the information about available objects, c) presence and location: all connected components have consistent the information about presence and location.

Experiments showed that, in spite of the dynamism and the autonomous and decentralized behavior of components, LaCOLLA required short amount of time (with respect to the rate of changes) to update the information referring to internal mechanisms in all components. Experiments also showed that members knew what was happening in the group and that they had access to the latest versions of objects in a time they perceived as immediate [4].

5 Conclusions and future work

Asynchronous collaborative applications have to adapt to group idiosyncrasy and interaction style and support the formation of ad hoc collaborative environments for people willing to cooperate using only their own computers, without any additional computing resources (i.e. servers). This requires the autonomy and self-sufficiency that peer-to-peer networks can only offer. We have identified groups as units of resource sharing, by which several individuals dispersed through Internet may spontaneously start to collaborate by just sharing their own computers to form an independent ad hoc community.

In this paper we have described the general characteristics and properties of LaCOLLA, a decentralized, autonomous and self-organized middleware for building collaborative applications that operates with resources provided by their members, that adapts to the idiosyncrasy and to the interaction nature of human groups, and that allows execution of tasks using resources belonging to the group. We also presented the details of current LaCOLLA middleware implementation, paying special attention to its API.

From both building collaborative applications that use LaCOLLA and from using the developed applications we obtained valuable ideas and improvements to introduce in the next versions of LaCOLLA. These new versions will pay special attention to security issues, which are at its mi-

nimum expression in the first version; and to introduce new components and mechanisms that will allow mobile devices (PDA, mobile phone, sensors, etc.) become LaCOLLA peers.

We are also planning to use LaCOLLA in real collaborative settings. In that sense, we are planning to use collaborative applications that use LaCOLLA middleware in some collaborative learning practices at UOC. UOC is a virtual university that mediates all relations between students and lecturers through Internet. We think that this kind of collaborative environments where participants never physically meet one to each other will benefit from approaches like the one provided by LaCOLLA, specially for the degrees of autonomy and self-sufficiency that can be achieved. These real experiences will be of great value for us to further refine the architecture and adjust the implementation of the middleware.

References

1. Foster, I.; Kesselman, C.; Tuecke, S. (2001). *The Anatomy of the Grid Enabling Scalable Virtual Organizations*. Lecture Notes in Computer Science.
2. Navarro L., Marquès J.M., Freitag F.(2004). On distributed Systems and CSCL. The First International Workshop on Collaborative Learning Applications of Grid Technology (CLAG 2004). ISBN 0-7803-8431-8.
3. Bote, M., Dimitriadis, Y., Gómez-Sánchez, E. (2003) Grid uses and characteristics: a grid definition. Proceedings First European Accross Grids Conference.
4. Marquès, J.M. (2003). LaCOLLA: una infraestructura autònoma i autoorganitzada per facilitar la col·laboració. Ph.D. thesis, <<http://people.ac.upc.es/marquès/LaCOLLA-tesiJM.pdf>>
5. Golding, R.A. (1992). Weak-consistency group communication and membership. Doctoral Thesis, University of California, Santa Cruz.
6. Saito, Y.; Shapiro, M.(2002). Replication: Optimistic Approaches. HP Technical Report HPL-2002-33. <<http://www.hpl.hp.com/techreports/2002/HPL-2002-33.html>>.
7. Carter R. L. (1995). Dynamic server selection in the Internet. In Proceedings of the Third IEEE Workshop on the HPCS'95.
8. Verbeke, J.; Nadgir, N.; Ruetsch, G.; Sharapov, I. (2002) Framework for Peer-to-Peer Distributed Computing in a Heterogeneous, Decentralized Environment. Manish Parashar (Ed.): Grid Computing, Third International Workshop, Baltimore, USA. LNCS 2536, ISBN 3-540-00133-6. <<http://jngi.jxta.org/>>

J.M. Marquès, X. Vilajosana, A. Daradoumis,
DSCL Research group (<http://lacolla.uoc.edu>)

Open University of Catalonia

Technology Watch

This section presents Technologies, Specifications and Standards related to the e-Learning and GRID world. A brief description will be given together with a set of references to "must read" articles and documents.

This month, in line with the theme of the issue, we will look at platforms, middleware and frameworks that can be used to develop collaborative learning applications on the Grid.

There are two possible approaches to developing such systems:

- (1) Integrate already available technologies to create a coherent infrastructure for Grid-based learning, or*
- (2) Develop new Grid-based learning infrastructure from the bottom up.*

It is not yet clear which of these approaches will turn out to be the best, and so here we have chosen to review two recent projects that together show both approaches in action.

SELF – a Semantic Grid-based E-Learning Framework

Integrating existing technology to support heterogeneity

The SELF project [1] takes the first of the approaches above. It aims to identify the key enablers in a practical Grid-based learning environment and implement them by integrating existing technological solutions. The proposed framework should provide a scaleable e-Learning infrastructure with e-Learning specific application layers that run on top of semantic Grid-based support layers.

The framework inherently supports heterogeneity of tools, technologies, security policies, data formats and locations within the environment by creating service layers that facilitate seamless integration.

By integrating already available technology rather than building a monolithic system from scratch they believe they can maximise the capability for future integration with other systems and avoid the need for major technical reworking of already existing systems and software components.

The SELF project has already identified many of the key enablers necessary for effective Computer-Supported Collaborative Learning (CSCL) by studying some of the major CSCL projects around the world [2,3,4,5,6]. Furthermore, they have subjectively assessed the support that can currently be provided for these enablers by the WWW and found it to be lacking. This leads them to conclude that the enablers should instead be mapped on to the features of a Grid infrastructure incorporating the semantic Grid (i.e. the use of semantic web technologies in a Grid computing environment).

They propose a semantic Grid-based framework (SELF) with a service-oriented semantic Grid middleware at the core. The SELF service stack is to be deployed at each participating (Virtual) organisation, and consists of the following layers:

1. *E-Learning applications layer.* This will consist of end-user applications which are dependent on user requirements.
2. *E-Learning services layer.* This will consist of generic application-level services, including intelligent search, matching and inference support provided by a small set of generic inference services. They believe that integration of the inference services will give increased interoperability and thus strengthen information retrieval functionality in the heterogeneous Grid environment.
3. *Semantic Grid layer.* This core component of the SELF stack addresses issues of heterogeneity and low standardisation across a conventional (i.e. non-semantic) Grid.

The next stage of the project is to implement these layers through the integration of existing technologies and software. A collection of candidate open tools and technologies has been identified as being suitable for this integration, including collaborative tools, personalisation and inference services, software agents, Grid components and services, ontologies, semantic web technologies and security systems.

The wide array of existing suitable technologies gives multiple options for implementation of the SELF system. The integration of so many components into a single framework will be a challenging task, and the Learning Grid community should keep a close eye on progress – if it is successful the SELF project could provide an incredibly flexible framework, based on open and already existing technologies, for constructing learning applications on the Grid.

LAGrid – Distance Learning Assessment Grid

A service-oriented Grid middleware platform for collaborative applications

The development of the Learning Assessment Grid (LAGrid) [7,8] has taken the second of the approaches described above: this service-oriented Grid middleware platform for collaborative applications has been developed from scratch and has been successfully deployed in a large-scale cross-organisation environment.

LAGrid has been developed as part of the *Grid-based Collaborative Platform for e-Learning* project, which is investigating the use of web services technologies to support collaborative learning in a Grid environment, and which aims to integrate the learning, software and teacher resources from over forty autonomous organisations, distributed in location and network, in China Central Radio & TV University (CRTVU). The platform so far seems to be aimed more towards supporting teachers in working collaboratively, but there is no reason why, in future, the infrastructure and middleware cannot also be used to support collaborative learning amongst students.

The LAGrid architecture has five layers:

1. *Infrastructure layer.* This provides the basic networking environment.
2. *Basic service-oriented architecture layer.* This provides the runtime environment for implementation of various web services protocols, providing connectivity, interoperability and reliability for the layers on top of it.
3. *Grid middleware layer.* This is the core of the Grid architecture, which addresses issues of distribution and dynamicity.
4. *CSCW layer.* This provides collaborative work and collaborative learning services over the Grid middleware.
5. *Domain-specific services and applications layer.* This layer supports domain-specific collaborative applications.

The Grid middleware layer is crucial to building a Grid environment, and consists of a set of “universal services” that together provide basic Grid application functionality. Some of the core services here are the Message-Oriented Middleware (MOM) that provides publish/subscribe event-driven messaging, the Service Aggregator (itself implemented as a compound web service) that performs composition of several different services into a new service, the Grid Information Service (GIS) that is a directory service manag-

ing four kinds of Grid entity metadata (metadata of Grid nodes, metadata of organisations, domains and departments, metadata of service interfaces and metadata of services), and a Bulk File Transfer service.

The CSCW layer in LAGrid is currently focussed on monitoring large-scale collaborative processes and on providing “team space” for groups of different sizes. Based on the Grid middleware, the functionality of this layer can clearly be extended in the future to provide complex interaction analysis and to give more advanced collaborative learning services.

Perhaps the most impressive aspect of LAGrid is that it has been successfully deployed to support all learning assessment-related services in CRTVU in five cities. It provides a single entry-point Grid portal allowing users to log in and request learning services from all over the Grid.

This is good evidence that LAGrid’s service-oriented middleware provides an effective platform for the development of collaborative applications, definitely making it one to watch.

References

- [1] Abbas Z., Umer M., Odeh M., McClatchey R., Ali A. and Ahmad F., *A Semantic Grid-based E-Learning Framework (SELF)*. 5th IEEE International Symposium on Cluster Computing and the Grid, 2005. <http://arxiv.org/pdf/cs.DC/0502051/>
- [2] The Access Grid global community. <http://www.accessgrid.org/>
- [3] Space Online Universal Learning (SOUL). <http://www.soul.hkospace.org>
- [4] Jin H., Yin Z., Yang X., Wang F., Ma J., Wang H., Yin J., *APPLE: A Novel P2P Based e-Learning Environment*. Proceedings of the 6th International Workshop on Distributed Computing (IWDC 2004), Springer LNCS, Volume 3326, Jan 2004, pp. 52 – 62.
- [5] Guang-zuo C. et al., *OntoEdu: Ontology based Education Grid System for e-Learning*, GCCCE2004 International conference, Hong Kong. 2004.
- [6] Microsoft ConferenceXP – wireless classrooms, collaboration, distance learning. <http://www.conferencexp.net/>
- [7] GuiLing W., YuShun L., ShengWen Y., ChunYu M., Jun X., MeiLin S., *Service-Oriented Grid Architecture and Middleware Technologies for Collaborative E-Learning*. Proceedings of the 2005 IEEE Conference on Services Computing (SCC’05), IEEE, 2005.
- [8] Learning Assessment Grid (LAGrid). <http://www.lagrid.cn/>

Kevin Keenoy

Birkbeck, University of London

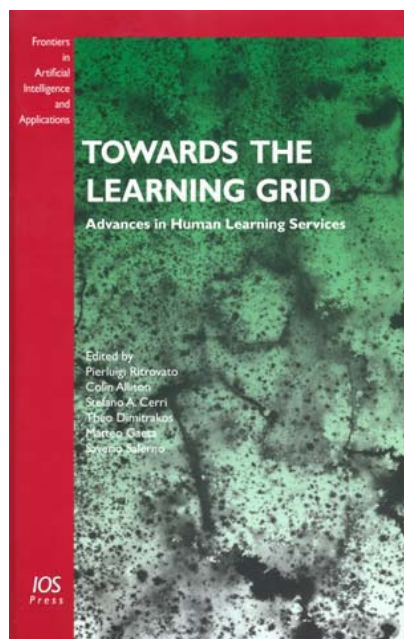
News

New Book: Towards the Learning Grid, Advances in Human Learning Services

November 2005

A new book on Learning Grid edited by P. Ritrovato, C. Allison, S.A. Cerri, T. Dimitrakos, M. Gaeta and S. Salerno has been published by IOS Press in Frontiers in Artificial Intelligence and Applications.

There is a paradigm shift in Informatics in general and in technologies enhancing human learning in particular. The debate between the "evolutionaries" – those that wish to optimize and refine current approaches – and the "revolutionaries" – those that support a fundamental change of approach – is quite actual. Within the Internet communities, the debate is hidden behind the words "semantic WEB" versus "semantic Grid"; within educational technologists between "content / resource centered" and "conversation centered" e-learning, or either between "teaching" and "pedagogy" on the one side, and "learning" and "communities of practice" on the other. In general, in Informatics, the shift from a product-page oriented to a service-conversation oriented view may possibly impact most if not all the foreseen applications, in e-learning, but also in e-science, e-democracy, e-commerce, e-health, etc.



Part A of the book is dedicated to Position papers: visions about what to do and why to do it in the next years. The remaining parts (B to D) offer partial answers to "how" to do it. Part B concerns what we called: Content-centered services, i.e.: a vision of learning systems that privileges knowledge and its structures, standards and their interoperability, storage and retrieval services. The subsequent part C has been called: Holistic services to refer to more mature and integrated solutions that address not only content but more generally the creation and management of human Virtual Communities connected on the Grid in order to offer and consume different services facilitating and enhancing human learning. Finally part D is concerned with new directions in learning services.

More information on www.iospress.nl.

Univa Ships Enterprise Globus Beta Release

20 December 2005

Univa Corp., the enterprise grid computing company launched by the founders of the open source Globus Toolkit, has announced the beta release of Univa Globus Enterprise (UGE) Release 4.0, the first shipment of its enterprise-ready, commercially supported software based on the de facto grid computing standard.

UGE includes new simplified installers and security configurations that facilitate rapid deployment of enterprise grids, and will be generally available early next year.

UGE adds new features that "deliver the simplicity, robustness and support demanded of commercial infrastructure software," Univa said. Based on the latest 4.0 release of the Globus Toolkit, UGE provides a core set of services that can be deployed and integrated into enterprise solutions based on commodity hardware and software.

UGE supports data management and execution services across a grid of hundreds or thousands of computing, storage and network devices. It allows resources to be shared securely across multiple sites; enables distributed data to be moved and replicated; facilitates distributed execution services across multiple job schedulers and workload managers; and provides monitoring information for managing the grid.

UGE is targeted at a broad range of applications in industries such as financial services, oil and gas, manufacturing and government, addressing the needs of geographically-dispersed organizations that want to optimize storage, compute, and network resources both within and beyond the corpo-

rate data center. IBM is among those licensing Univa's first release.

UGE Release 4.0 supports Linux and Unix running on x86, and IBM pSeries, zSeries and BladeCenter platforms and is offered on an annual subscription basis. For more information, visit www.univa.com/products/.

Source: GRID Computing Planet.

Grid Discovers Largest Known Prime Number

28 December 2005

Using an international grid of about 70,000 computers, researchers this month discovered the largest known prime number. The number, which is expressed as 2 to the 30,402,457th power minus 1, is a 9.1 million-digit figure. It was discovered by a team of researchers at Central Missouri State University.

"Working alone, it would take a brand new Pentium 4 computer about 4,500 years to find this number," says George Woltman, a retired computer programmer and founder of the Great Internet Mersenne Prime Search (GIMPS). "It took all 70,000 computers 10 months to make this find. Each computer on the grid tests a different number and, depending on the speed of the computer, it takes about two weeks to a few months."

Mersenne primes have, in many cases, been found by individuals, but this time the result came from a team that so far has contributed more processing time than any others -- the equivalent of 67,000 years running a 90MHz Pentium computer, according to GIMPS records. The Central Missouri State University team, led by professors Curtis Cooper and Steven Boone, used GIMPS software that ran on and off for about 50 days.

The new prime was independently verified in five days by Tony Reix of Bull S.A. in Grenoble (FR).

Source: GRID Computing Planet.

Academia Taking Standards Seriously to Improve Quality and Reduce or Contain Cost

12 January 2006

The Educational Technology Cooperative, composed of K-12 and higher education agencies in the Southern Regional Education Board's (SREB)

16 states, is working together closely to improve learning through the use of technology. Leaders in these agencies understand current and future issues including the adoption of standards to allow for more sharing and exchange. The leaders also provide strategic guidance to schools and colleges in their states. The SREB partnership with the Academic ADL (Advanced Distributed Learning) Co-Lab at the University of Wisconsin is proving invaluable to the work of state education leaders as several standards-based initiatives gain traction.

At a recent Cooperative meeting in Atlanta concerning online learning, the ADL Initiative presented information about repositories and registries -- critical components of the multi-state SREB SCORE (Sharable Content Object Repositories for Education) initiative. SCORE registry work is based upon work the Department of Defense has done with learning object repositories known as CORDRA.

A key SCORE document is the [SREB Technical Guidelines on Digital Learning Content](#) in which SCORM serves as an important consideration in development and acquisition of learning content. These guidelines were developed over several years as SREB worked closely with the Academic ADL Co-Lab to identify key issues and topics essential for effective use of digital learning content in teaching and learning.

SREB also partnered with other regional compacts and the Academic ADL Co-Lab through a national education group called the ATAlliance to obtain quality products and services for schools and colleges. A key focus this year has been on eLearning Management Solution companies -- companies that provide platforms for online courses. Learning Repositories were also evaluated. Standards were a key element identified in the request for proposal and the subsequent evaluation. Companies were judged in part by their level of use of standards to guide the development and implementation of their products.

Source: Advance Distributed Learning.

SCORM 2004 3rd Edition Impacts Summary Released

13 January 2006

The ADL Initiative has released the [SCORM 2004 3rd Edition Impacts Summary](#). The upcoming release of the SCORM 2004 3rd Edition will contain several changes that were influenced by various factors and events. These changes continue to make SCORM 2004 more stable and robust. This document provides a summary of those key

changes and the impacts to current SCORM 2004 2nd Edition products.

This document should not be treated as an exhaustive listing of all the changes found in SCORM 2004 3rd Edition, but rather as a guide to be used with the SCORM 2004 3rd Edition documentation suite. The document gives the reader an understanding of the changes from SCORM 2004 2nd Edition to SCORM 2004 3rd Edition and to assist in the determination of what changes are needed to SCORM 2004 products to migrate them to match the requirements defined in SCORM 2004 3rd Edition.

The document is being made available prior to the release of the SCORM 2004 3rd Edition to allow for organizations to review the types of changes. The ADL Technical Team encourages users to provide feedback if there are any additional questions or comments.

Source: Advance Distributed Learning.

New CERN Project, ETICS, to Improve Grid Quality

20 January 2006

The kick-off meeting for a new project called ETICS (eInfrastructure for Testing, Integration and Configuration of Software) is being held at CERN today. The goal of this project, which is coordinated by CERN and funded partially by the European Commission, is to improve the quality of Grid and distributed software by offering a practical quality assurance process to software projects, based on a build and test service. This is a first of its kind in Grid computing.

In the short term, the ETICS project will offer a build and test service via a secure Web interface and command line tools. In the medium term, distributed testing will be added, as well as a secure repository of components built by the ETICS service on a variety of platforms, with well-known and controlled external dependencies. The longer term goal of ETICS is to propose a certification process for Grid and distributed software (for example middleware, components, services and user level applications). The goal of this certification process is to promote practical quality assurance for Grid and distributed software, and to provide a quantitative measure of the quality level of software.

The quality assurance process offered by ETICS will help users developing the next generation of Grid software. In particular, ETICS will help projects developing a clear set of well behaved and high quality software components, services,

plug-ins and applications. Once the certification process is in place, ETICS aims to ensure that it becomes a globally recognized statement of quality for Grid and distributed software. This process should also contribute to improving interoperability between Grid and distributed software stacks.

For more information about the ETICS project, visit <http://eu-etics.org>. Source: GRID Today.

EUChinaGRID Project Kicks Off

25 January 2006

The EUChinaGRID (Interconnection & Interoperability of Grids between Europe & China) Project officially kicked off in Athens this week. The main goal of this European Commission funded project is to support the integration and interoperability of the Grid infrastructures in Europe and China for the benefit of e-science applications and worldwide Grid initiatives, in line with the support of the intercontinental extension of the European Research Area (ERA).

A second important objective for EUChinaGRID is the dissemination and training activity, which aims to improve the accessibility of the Grid infrastructure for new research groups and applications, promoting scientific and, possibly, industrial developments.

The main aim of the Project is to facilitate exchange and processing of scientific data supporting a pilot interoperable usage of the most relevant Grid Infrastructures in Europe (developed within the EU EGEE Project) and in China (managed by the CNGrid Project).

The Project will use the existing and planned infrastructure provided by the research networks like the Gigabit Pan-European Research & Education Network (GEANT), exploit the initiatives of high speed intercontinental network connections, like TEIN2 and ORIENT, and work in synergy with them for the optimization of network usage in agreement with the objectives of the Communication Network Development scheme.

Moreover, EUChinaGRID will use EGEE supported applications and the established common practices and policies to foster the migration of new applications on the Grid infrastructures in Europe and China by training new user communities and support the adoption of powerful Grid tools and services.

For further information about the EUChinaGRID project, please visit www.euchinagrid.org.

Source: GRID Today.

When	What	Where
13-16 February 2006	<p>GGF16 - The 16th Global Grid Forum.</p> <p>The 4 day meeting will take place at the Divani Caravel Hotel ideally situated in the heart of the city. GRNET, together with the GGF Community Council, will offer an exciting regional plenary program that will bring together Greek and other European experts in the field of grid and distributed computing. In addition to the regional program, GGF16 will provide an environment to continue the work of our Standards Groups through WG/RG sessions, BoFs and workshops. GGF's community program will continue throughout the week, emphasizing the applications and operation of grids in eScience and other research environments. The community program will consist of workshops, invited talks, technical tutorials, and demonstrations of standards-based implementations of grid software.</p> <p>More information at: www.ggf.org/GGF16/ggf_events_ggf16.htm.</p>	Athens, Greece
27-31 March, 2006	<p>Innovating e-Learning 2006: Transforming Learning Experiences.</p> <p>This conference is based around three topics: Designing for learning, Learner experiences with e-learning and Innovating e-Learning Practice and will be of interest to practitioners, e-learning coordinators and managers in further and higher education, researchers, staff developers and learning technologists.</p> <p>More information at: http://www.jisc.ac.uk/elp_conference06.html.</p> <p>For those who never participated in an online conference, see: http://www.jisc.ac.uk/index.cfm?name=elp_conference06_faq</p>	On the Internet
10-12 April, 2006	<p>Fifth International Conference on Networked Learning 2006.</p> <p>This conference is an opportunity to participate in a forum for the critical examination and analysis of research into networked learning ie learning and teaching carried out largely via the internet or web which emphasises collaborative and cooperative learning, online group work, interaction with e-learning materials and networked knowledge production.</p> <p>More information at: http://www.networkedlearningconference.org.uk/</p>	Lancaster University, UK
25-19 April, 2006	<p>20th IEEE International Parallel & Distributed Processing Symposium.</p> <p>The conference is presented by IEEE Computer Society Technical Committee on Parallel Processing in cooperation with ACM SIGARCH, IEEE Computer Society Technical Committee on Computer Architecture, and IEEE Computer Society Technical Committee on Distributed Processing. The conference will be hosted by the Research Academic Computer Technology Institute (CTI), Greece.</p> <p>More information at: www.ipdps.org.</p>	Rhodes Island, Greece
8-12 May, 2006	<p>Fifth International Workshop on Agents and Peer-to-Peer Computing (AP2PC 2006) to be held at the Fifth International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS 2006).</p> <p>The MultiAgent paradigm can be superimposed on the P2P architecture, where agents embody the description of the task environments, the decision-support capabilities, the collective behavior, and the interaction protocols of each peer.</p> <p>This workshop will bring together researchers working on agent systems and P2P computing with the intention of strengthening this connection. Research in Agents and Peer to Peer is by its nature interdisciplinary and offers challenges for several communities, such as distributed systems, networks and database systems.</p> <p>More information at: http://p2p.ingce.unibo.it/</p>	Hakodate, Japan

When	What	Where
16-19 May, 2006	<p>The 6th International Workshop on Global and Peer-to-peer Computing organized with the IEEE/ACM International Symposium on Cluster Computing and the Grid 2006 IEEE/ACM CCGRID 2006.</p> <p>Global Computing systems, Desktop Grids and Peer-to-Peer systems (P2P) aiming to harness Internet-connected resources at a global scale. These platforms can be built in a pre-defined, stable, organized fashion (more like today's Grids) or can be self-organizing, ad-hoc, decentralized, like in P2P systems.</p> <p>Supporting computation on such systems raises a novel set of questions. Because of size, autonomy and the high volatility of their resources, these platforms provide the opportunity to revisit major fields of distributed computing such as protocols, infrastructures, security, fault tolerance, scheduling, performance, services, applications, and incentives for cooperation. Moreover, new issues concerning the installation, maintenance and scalability of large-scale distributed systems become more relevant than ever.</p> <p>This workshop will describe current research and novel ideas in the area of Global, Desktop Grid and Peer-to-Peer Computing as well as experiences with the deployment of related technologies.</p> <p>More information at: http://GP2PC.lri.fr</p>	Singapore
3-6 July, 2006	<p>International Advanced Research Workshop On High Performance Computing and Grids (HPC 2006).</p> <p>The aim of the Workshop is to discuss the future developments in the HPC technologies, and to contribute to assess the main aspects of Grids, with special emphasis on solutions to grid computing deployment. The HPC Advanced Workshops in Cetraro have been well established and two of them (1992 and 1996) were sponsored by NATO.</p> <p>Workshop Topics are: General Issues in High Performance Computing; Advanced Technologies for Petaflops Computing; Emerging Computer Architectures and Their Performance; Programming Models; Parallel Languages; Parallel Software Tools and Environments; Distributed Systems and Algorithms; Parallel Multimedia Computing Technologies; Innovative Applications in Science and Industry; High Performance Computing for Commercial Applications; General Issues in Grid Computing; Grid Computing for Enterprise: security, system life cycle management, reliability, accountability. More information at: www.hpcc.unical.it/hpc2006.</p>	Cetraro, Italy
4-7 July, 2006	<p>Workshop on P2P Data and Knowledge Sharing (P2P/DAKS) in conjunction with The IEEE 26th International Conference on Distributed Computing Systems (ICDCS-2006).</p> <p>The aim of this half-day workshop, within ICDCS 2006, is to explore the P2P area of data and knowledge sharing [P2P/DAKS] for information intensive applications. The objective is to blend concepts and ideas from the database and AI domains in order to produce a sound basis for an effective P2P/DAKS facility.</p> <p>More information at: http://www.takilab.k.dendai.ac.jp/conf/DAKS/2006/</p>	Lisboa, Portugal