# WWG: A DISTRIBUTED INFRASTRUCTURE FOR LEARNING IN GROUPS

*Joan Manuel Marquès[1], Leandro Navarro[2], Atanasi Daradoumis[3]*

***Abstract* — This paper describes the experiences of learning and working in groups on the Computer Science studies at Virtual Campus of the Open University of Catalunya. Two communities have used a web based shared workspace tool to a) coordinate the task of tutors of a computer architecture course, and b) a problem based collaborative learning project with software engineering students. These experiences have lead to the design of WWG: a distributed infrastructure for distance learning in groups at Internet Scale. WWG is based on the distribution of events about the information produced by people. Every participant can be notified and thus be aware of the actions, changes, and progress of the groups he belongs to: synchronous awareness for asynchronous work. There are three classes of components that work together: user agents for collecting and presenting actions to people, meta-information agents to distribute event information, and repository agents to provide distributed storage for group objects.***

*Index Terms — CSCL environment, event distribution, Internet scale collaboration, PBCL.*

## INTRODUCTION

Group learning and in general, group work is an activity that is increasingly being perceived as beneficial and necessary for a more active and better learning process. The period of intensive learning that occurs at the school and the university is a great opportunity to acquire team work skills. These skills will be key to success in the professional activity.

In addition, the learning process is increasingly influenced and mediated by computers, and a fast growing number of universities are beginning to offer virtual campuses to support distance learning, because many students and sometimes professors are in remote locations, they have temporal restrictions due to overlapping activities, or self pace learning is preferred. Furthermore, students engage in projects or courses with a few participants from diverse locations interested in very particular topics.

The challenge of supporting cooperative learning and group work in this large-scale distributed context is the motivation of this work.

This is the experience of the authors at the "Open University of Catalonia" (UOC) (http://www.uoc.es), a virtual university providing university education to the Catalan and Spanish speaking world; also at UPC (http://www.upc.es), an established university giving in person and half-distance university engineering education.

Since the beginning of the Computer Sciences studies in the UOC (1997) learning through working in group has been a main issue. Several experiences [4] [5] have enlightened the design of our infrastructure. In one hand, we realized that to most people computer mediated group learning is a brand new way of learning and they need first to get used to it. On the other hand, learning in group needs some extra awareness information to know what the other members of the group are doing, information not needed in individual learning. That awareness information is not well provided by present applications because the infrastructure they use was designed to support isolated work. Our proposal is focused to provide that awareness information as the key design aspect.

The role of the tutor on those environments differs a lot. It goes from very active involvement where he organizes the interactions, observes, acts as another member of the group and at the end assesses the overall work, to the case in which the tutor only supervises the group and grades the students. In any case, the tutor needs information about what every group member is doing and has done. It requires special computer support to refine and abstract what is happening in the group.

The rest of the paper describes two experiences collaborative learning experiences that have demonstrated the importance of awareness information about group activity, and the need to support tutors to observe patterns of activity that require their intervention, without being overloaded by the large number of events. The extension from campus scale to Internet scale defines several issues, translated into requirements, with focus on event distribution as a basis. Then, the WWG architecture is described. The paper ends with a discussion on propagation of events and conclusions.

## COLLABORATIVE LEARNING EXPERIENCES

In this paper we present two kinds of virtual collaborative experiences with students and tutors of Computer Science Engineering we have done at UOC. The first experience is the coordination activity of all the tutors of a Computer Architecture subject. The second experience is a Problem-based Collaborative Learning (PBCL).

This work has been partially supported by the CICYT- Spain.

[1] Universitat Oberta de Catalunya, Universitat Politècnica de Catalunya, Barcelona, Spain, jmarquesp@campus.uoc.es

[2] Computer Architecture Department, Universitat Politècnica de Catalunya, Barcelona, Spain, leandro@ac.upc.es

[3] Universitat Oberta de Catalunya, Barcelona, Spain, adaradoumis@campus.uoc.es

### Tutor coordination in a Computer Architecture course

As UOC is a virtual University, all the communication between members of the UOC community has to be done by electronic means. Students and tutors are from all around the country and they study or work from home. Those tutors' coordination experiences have been done during the last two semesters, involving 10 tutors each semester. The 10 tutors were responsible of facilitating the learning to around 650 students each semester. Due that at the end of the semesters all the students have to do the same exam with an anonymous correction, all the tutors have to coordinate and collaborate:

- Coordination to agree upon the: contents of the course; pace of study; partial examinations; evaluation criteria; kind of activities and exams.
- Collaboration to: prepare exams and partial examinations; prepare new activities; decide changes on the materials (as we decided to update the contents).

To achieve all those coordination and collaboration tasks, they had to deal with:

- Explicit communication: discuss; interchange documents; negotiate; reach agreements, making them explicit
- Implicit communication: know what are doing other group members. Know if all members are participating. Know when there are new documents or participations in a discussion.

All that information has to be organized; be accessible to anyone anytime; and it should be in a shared environment.

For all those reasons, we have used BSCW (Basic Support for Collaborative Work) [2], a collaborative tool that allow members of a group organize their work in shared workspaces.

Some tasks were done by subgroups. The basis of the communication within the group was the shared workspaces, even though for some specific tasks they had used person-to-person communication (i.e. email, phone, chat, face to face meetings). In those cases the resulting documents or summary of the discussions was placed in the shared workspace.

The experience has shown that shared workspaces complemented with awareness information about group and individuals activity makes collaborative and coordination work more effective, as reported by the participants. One of the most appreciated BSCW functionalities was the daily reports -an email summarizing the events done by the members of the group during the day.

### PBCL in a Software Development course

As in real life software is developed in teams, we wanted that our students had the opportunity of learning by experimentation. That's the reason why 17 students of Software Development Techniques -divided in 4 groups of four to five students- were involved in a collaborative realization of a project using BSCW as collaborative tool.

Our conceptual model of PBCL [5] integrates both synchronous and asynchronous communication, though at the present collaboration is mainly based on the later due to the characteristics of our distance-learning students. Its organizational structure is planned in three consecutive layers: initiating, forming and performing.

The first layer serves to initiate the students into the new experience and get to know each other. The second is used to carry out the rather complex task of group formation and organization and thus establish an accepted group normative. The third layer is where the different PBCL groups realize the project. To incite and promote cooperative interactions and make collaborative work and learning possible, we lead students through a guided process that involves achieving several learning objectives. The time period for realizing the project is of four months.

Tutors need to keep track of how learning evolves and knowledge is built as well as how the learning process is affected by the following factors: the type of interaction (contributions) made, the type of cognitive and social actions performed, the social dynamics, roles and interrelations developed among the group members, as well as the role the tutor has to take in supervising and guiding the learning-process of the students.

Moreover, for effective collaborative learning to occur, both actions and interactions need to be well organized in the shared workspace of the group. In particular, on the one hand, we need to provide the means for the management and structuring of the student interactions; on the other hand, we should care for the efficient organization and management of the information and knowledge produced when interactions and collaborative or individual actions take place.

### Student tracking and awareness

Student tracking and awareness influence a great deal not only the successful completion of the project but also the whole collaborative learning process. Our experience shows that the tutor needs to apply a careful and well-studied strategy in order to relieve the overload and difficulties that this process represents. To that end, we settled an active tracking based on several objectives, such as:

- Remind the students continuously that they are not alone "fighting" against a project that ends up with thousands of code lines
- Be aware of increasing difficulties as the project evolves and hence propose the students ways out when the project gets stuck due to technical difficulties
- Be aware of certain undesirable situations within the group and hence propose the students reasonable yet

human solutions to critical situations concerning the group functioning

- Provide group dynamics incentives so that students get more and more interested in the project.
- Observe individual member skills and give adequate orientations.

It's worth mentioning here that we established two kinds of tracking, namely, individual tracking and group tracking. At individual level, the objective was to closely track the contributions of individual group members, while group tracking aimed to trace the group activities. In this way we were able to orientate correctly both individual and group work.

We have observed that the tracking process has much to do with the communication flows settled in the collaborative framework. We distinguished the following three main communication streams: (a) between the tutor and individual members; (b) between the tutor and different groups; (c) between members of the same group. A major difficulty with the tracking and awareness process is the rather large amount of information to be processed which might lead to a saturation of the tutor's possibilities. In our framework, we could find a trade-off by directing part of the communication flow to the coordinators of the groups.

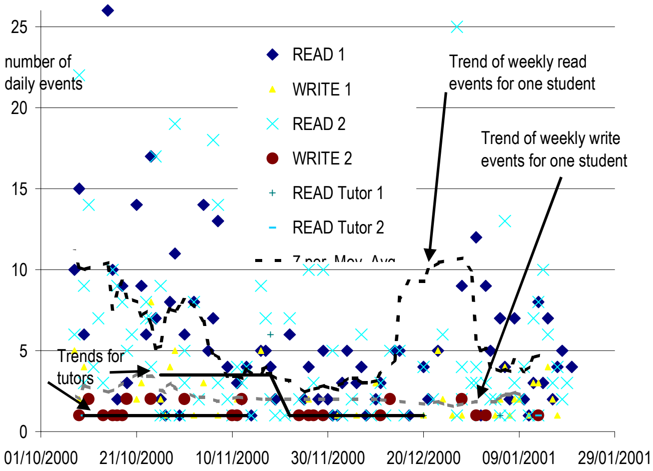

FIGURE. 1
DAILY READ, WRITE EVENTS FOR A PBCL GROUP

The UOC Virtual Campus and BSCW have been useful to support groups of students, but they are not designed to facilitate tutors the supervision of the learning process. The tutor needs information on the evolution of every group, in the right level of abstraction for the task and volume of information to avoid being overloaded. It needs information about the progress of groups not only to evaluate students, but also to improve their learning process: know the status of every group, clarify doubts, detect and support delayed groups, etc.

From that PBCL experience we concluded that a good metaphor to represent the groups evolving is a participation metaphor. The result of analyzing the contributions to the group and the effects that those contributions has had on other contributions, can help the tutor to understand the group idiosyncrasy. To do that, it is required to collect the maximum amount possible of events and to distribute those events to other members and the tutor.

## LEARNING IN GROUP AT INTERNET SCALE

The extension of learning activities from a campus scale to Internet scale presents for participants several problems of cultural and environmental diversity, feeling of isolation, lack of context, and for the support infrastructure problems of scalability and inter-operability between heterogeneous systems.

On the positive side, it opens up new opportunities for participating in learning experiences anywhere, creating focused learning communities scattered across the world, not viable in traditional virtual campuses.

This suggests the need for a cooperative learning infrastructure to support a large number of groups spread over the Internet. This is complementary to ARIADNE [1], IMS [6] or IEEE Learning Technology Task Force [8].

The following issues have been considered the most influential to our design:

- Multiplicity: people may belong to several groups at the same time. The degree of involvement may differ heavily. Users need facilitation to handle the complexity of multiplicity+diversity.
- Group membership may be relatively small, even though there may be large groups.
- Awareness: effective group work requires that members must be aware of the progress of the group: up-to-date and rich information about what others are doing, at low cost, at a glance.
- Quality of service: Clients will be offered the most accessible server from the set of currently available.
- Mobility: one person may connect from different environments: work, home, mobile, etc. The view of the groups must be the same from any location.
- Degree of connectivity: many group activities do not require to be always connected to the rest of members.

### Requirements

In the design of WWG we have addressed each of the previous issues. These issues have been translated into requirements that are briefly described in the following basic requirements for an infrastructure to handle easily and efficiently many learning/work groups at Internet scale:

- Information must be accessible at any time, and be managed transparently. The user does not have to worry about location, accessibility, replication of information.
- The user needs the appropriate amount of information produced by the group in form of documents, messages and events (awareness information).
- The system must be scalable: large number of participants, large number of events, participants distributed across large distance, decentralized.
- Group members must have information accurate, updated and consistent about actions being carried out by the rest of the group.
- Objects may be accessible from any location with an appropriate (interactive) response time.
- Adaptable to the needs of users: info should be where is more convenient to users. The user also requires availability, reliability and a good access time.
- Adaptive to the needs of the system: load balancing, balancing of storage, and minimizing the amount of information exchanged.
- Multiple access points: when a user moves to a new location, the system must adapt dynamically and provide a closer service access point.

Existing systems do not support the above requirements and issues. The goal of WWG is to provide an infrastructure for information management and propagation, without prescribing how information is represented or how applications operate.

### Event Distribution

Given that WWG is aimed at supporting learning and working in groups, the key factor is that group individuals should be informed immediately of whatever occurs within their groups. This is provided by the event distribution mechanism.

WWG is intended for situations where users get virtually synchronous information (equivalent to real-time information but relaxed to scale better and save resources) about the actions that occur on the system. In terms of system design, synchronous event distribution allows us to do the following assumptions:

- Consistency through events: virtual synchrony and consistent distribution of events can lead to a consistent distributed and replicated system. Consistency is possible because the system always knows where the latest version of every object is located.
- Events provide "sense of immediateness": event distribution provides information about what is happening now in the system i.e. in the groups of interest.
- Events provide "maximum information": when a learning or working activity is done in groups is of great importance to have the maximum amount of information about what are doing all participants. For us, "maximum information" means both the number of events received by a member and the amount of information that every event conveys.

Once we have decided that our system is based on event distribution, the next step is to design an architecture to guarantee a distribution of events that facilitates the achievement of these assumptions.

## The Architecture of WWG Middleware

Therefore, WWG is a distributed and decentralized infrastructure with the aim of supporting distributed group learning and team work, centered on the distribution of events, so that every participant can be notified and thus be aware of the actions, changes, progress of the groups he belongs to. WWG has been designed for situations where participants interact and work asynchronously, but receive synchronously information about the actions done in the group. This event distribution mechanism provides consistency, sense of immediateness, and awareness about what's going on.

Participants may need to transform (summarize, condense) events, to give the required information to group members with diverse degree or mode of participation. This is the case for tutors, teachers, professors, assistants, supervisors, moderators, evaluators, etc.

The following features guided the design of WWG:

- Open infrastructure: Application independent; extendable protocols
- Internet scale: participants can be anywhere; supports an arbitrary large number of groups; membership depends on the task and the degree of involvement of members.
- Synchronous distribution of events: Synchronous awareness
- Members of the group share objects asynchronously. Small objects can be embedded into events providing a virtual synchronism. E.g. a chat interaction.
- Interaction organized in groups: events and objects are restricted to the group. The group may be presented to the users as a group workspace.
- Some events are provided. Applications may extend existing events or define new application specific events.

### WWG Components

The **user agent** represents users in the system. It collects notifications of all actions done by the user. Once notified, the user agent has to interact with the rest of the system to get the action processed or to get the information about the action distributed to other members of the group, in form of an event. It is also in charge of receiving events about actions done by other members of the group and to provide this information to the user.

**Repository agents** are dedicated to the storage of the information generated by the group (documents, discussions, events, users, groups, folders, etc). To facilitate the availability and the accessibility to information on a potentially large scale, information may be replicated in different storage components depending on the needs of every group.

User agents and repository agents are interrelated by an intermediate layer in charge of the distribution of events. This layer is composed by **meta-information agents**, in charge of efficient distribution of information (events) generated by the users and the system. Meta-information agents efficiently route and distribute event information to interested agents; filter, aggregate and transform events; and suggest the best meta-information agent for each user agent, helping repository agents to decide the best location and number of replicas needed for an object.

### A unicast+multicast architecture

The WWG network is composed by coordinated computers running one, two or three of the following functions: user agent, meta-information agent, or repository agent.

Events have to arrive orderly to every destination interested in, as soon as possible (interactive delivery, virtually synchronously), keeping to a minimum the volume of messages exchanged (by aggregation of events).

The exchange of information between a user agent and a meta-information agent or between a repository agent and a meta-information agent is done using a unicast reliable transport protocol (TCP). A Multicast transport is optimal for event dissemination among meta-information agents. The combined use of unicast and multicast protocols allows an efficient use of the network

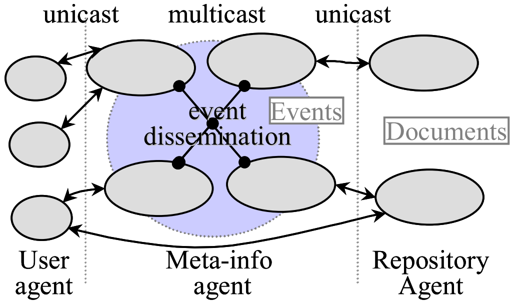

FIGURE. 2
WWG ARCHITECTURE

Repository agents cooperate to provide distributed and/or replicated network storage for objects. Group members should have transparent access to their objects with a reasonable quality.

Event information is very dynamic and abundant in any collaborative setting, and that is clearly useful for user agents to be aware of the progress of groups, but that is also useful for repository agents to decide where objects have to be located.

### The propagation of events

The experiences at UOC proved that users have a finite capacity of processing events [5]. A user, depending on the number of groups to which he belongs, on the activity of those groups and in his degree of involvement, needs to receive events with a different level of abstraction. For instance, in a group formed by three people writing a document, all the members may want to get all generated events; but a tutor responsible for six groups, with three members in each group generating events can be easily overloaded. In this case, the tutor needs fewer but more abstract events. These new inferred events are the combination of several related events.

The diversity of kinds of events can be classified under different points of view. A possible classification is about the kind of information that carries the event:

- User-actions events: events generated by the application (task-oriented awareness [9]) or the user agent (social awareness) for each user action. Examples of that kind of events are: read, create, delete, modify, copy, paste, undelete, etc. documents or messages.
- Inferred events: the virtual groups members need information of how the group evolves. The inferred events are particular interpretations about the group evolving. The user agent (or a client application) has information about the group and the actions done by the local user. With all that information, like an external observer, the user agent infers events about the group evolving. Those inferred events are perceptions: they are neither true nor false.

Events can also be classified thinking on the immediateness required by members of the group for the events:

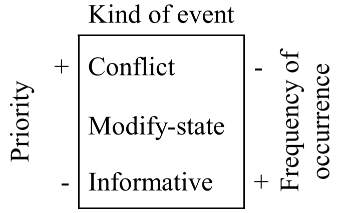

FIGURE. 3
PRIORITY VS. FREQUENCY OF KINDS OF EVENTS

- Conflict events: the events that inform that a conflict has occurred or the events that tries to solve the conflict. In asynchronous activities, the conflicts will be rare. In an environment such as WWG most of the conflicts can be avoided by choosing carefully some design alternatives. If the conflict cannot be solved automatically, the members of the group will be informed and someone will be responsible for the explicit resolution (as in [7]). Conflicts and conflict resolution has been studied in a separate report.

- Modify-state events: the events produced after an action that modifies the global state of the system: a new document, a delete action, a change of location, etc.
- Informative events: the events that inform about actions that don't modify the global state of the system. Those events include actions as read document or message and inferred events.

Events related to conflicts must be sent immediately. Modify-state events must be sent as soon as possible. Different policies can be applied to informative events, which are the majority. It is not the aim of that paper to study those policies, nevertheless we present two possibilities: aggregation (when 10 actions occur in an object, a single event is sent indicating that 10 actions have occurred), grouping (when several events goes to the same destination, send all of them in the same message).

During the experiences at UOC, 23.566 events were generated. From those, 80% were informative events and 20% were modify-state events, and only 2.2% were events that may cause conflicts if not propagated immediately. This supports our intuition that most of the events generated in a system are informative events. That conclusion is even reinforced because, in the version of BSCW we used, a read event is only generated the first time a user reads a document. Successive are not recorded as events. In WWG we want to distribute all the events generated on the system to give the "maximum information" to the users. Then, all the read actions would need to be distributed.

### A WWG Prototype

A WWG prototype has been developed to prove the viability of the architecture. One component is a group browser, a Java application that uses the Swing user interface components to present events about the activity of other group participants, and an object workspace where people can express interest in several groups, browse through documents and folders in these groups, and publish new documents. As a result of user actions, events are passed to a meta-information agent and then propagated to other meta-information agents and finally to other group browsers interested in the same group. Objects are stored at the closest repository agent.

Siena [3] is used to provide event distribution among meta-information agents. Repository agents have been implemented in the same process to simplify development.

In the second phase of development, we are going to develop a shared argumentation tool for an experiment with several people collaborating to produce a report.

To improve interoperability events will be expressed as structured messages based on a XML vocabulary [10] with structured data eventually including related objects.

## CONCLUSIONS

Two virtual collaborative experiences have demonstrated the value of awareness information for group members and the need to support tutors in the supervision of the group progress and performance.

These experiments have informed the design of WWG, a distributed and decentralized infrastructure with the aim of supporting group learning and team work, based on distribution of events, in large scale networks.

The WWG infrastructure may be useful to extend existing centralized systems such as BSCW that give support for small to medium scale groups, but it may also be an important improvement for large scale groups now using primitive tools not adapted to collaborative learning such as mailing lists or Usenet News.

Initial work shows the viability of WWG, but work is under way to demonstrate and optimize their scalability, evaluate how awareness is supported, describe the operations, and test the prototype with one or several groups doing real work.

## REFERENCES

[1] Ariadne Project (Alliance of Remote Instructional Autoring and Distribution Networks for Europe), CE. 1996-2000.

[2] Bentley, R., Horstmann, T., Sikkel, K. and Trevor, J., "Supporting Collaborative Information Sharing with the World Wide Web: The BSCW Shared Workspace System", *Proceedings of the 4th International WWW Conference*, Issue 1, December 1995, pp 63-74.

[3] Carzaniga A., Ronsenblum D. S., Wolf, A. L., "Achieving Scalability and Expressiveness in an Internet-Scale Event Notification Service", in *Nineteenth ACM Symposium on Principles of Distributed Computing (PODC2000)*, July 2000.

[4] Daradoumis, T. and Marquès J.M. A Methodological Approach to Networked Collaborative Learning: Design and Pedagogy Issues. In: Proceedings of the 2nd International Conference on Networked Learning. Lancaster University, England, April 17-19, 2000.

[5] Daradoumis T., Xhafa F., and Marquès J.M. "A Methodological Framework for Project-based Collaborative Learning". *Internal report.*

[6] Instructional Management System, URL: http://www.imsproject.org

[7] Kistler, J.J., Satyanarayanan, M., "Disconnected Operation in the Coda File System", in *ACM Transactions on Computer Systems*, Feb. 1992, Vol. 10, No. 1, pp. 3-25 http://www.coda.cs.cmu.edu/

[8] IEEE Computer Society Learning Technology Task Force (LTTF), URL: http://lttf.ieee.org/

[9] Prinz, W., "NESSIE: An Awareness Environment for Cooperative Settings", in *Proceedings of the ECSCW '99 Conference.*

[10] W3C Architecture: Extensible Markup Language (XML), URL: http://www.w3.org/XML/