

Group Prediction in Collaborative Learning

R. Messeguer, E. Medina, D. Royo, L. Navarro

Department of Computer Architecture
Technical University of Catalonia
Spain

J.P. Juárez

Division of Economic-Administrative Sciences
Department of Organizational Studies
University of Guanajuato
Mexico

Abstract— We propose an approach for predicting group formations, to address the problem of automating the incorporation of group awareness into CSCL applications. Contextual information can enable the construction of applications that effectively assist the group members to automatically communicate in synchronous and collocated collaborative learning activities. We used data traces collected from the study of students' behavior to train and test an intelligent system. Results have shown that context-information can be effectively used as a basis for a middleware for a dynamic group management. Inferring group membership is technically viable and can be used in real world settings.

Group prediction; Computer-Supported Collaborative Learning (CSCL); Context-awareness; Intelligent system

I. INTRODUCTION

Collaborative learning is an instruction method based on students working together in small groups to accomplish shared learning goals [1]. Group members do not work individually, but remain engaged in a shared task that is constructed and maintained by the participants. In addition, the students work and cooperate among themselves, helping each other to achieve the learning goals. The cooperative learning model is student-centered and encourages students to cooperate and collaborate with each other in order to achieve their learning outcomes. Among many collaborative learning methods, also considered as group learning approaches and used in both classroom-based and web-based environments, we have focused on the Project Based Learning and the Jigsaw techniques. These methods entail a dynamic setting where multiple parallel groups of students join and disband rapidly to form new groups.

In addition, beyond paper and pencil, students may use mobile devices – such as laptops, smart phones or PDAs, with wireless network connections– combined with computer applications, which provide support not only in the creation and manipulation of data, but also facilitate and encourage collaboration among the students. However, in that scenario there is a potential barrier on the integration of the real-world objects with the computer-based objects. Bridging that barrier by sharing contextual information makes interaction across real-world and computer-based objects invisible to the user [2].

Computer-Supported Collaborative Learning (CSCL) applications are used in collocated cooperative learning settings to support group tasks -in planned and unplanned situations-. Such applications require up-to-date contextual

information about the groups that are participating in the collaborative activity. Since this information is changing continuously, its manual introduction may stress the already overloaded teacher and introduce delays and an additional burden to any participant in a real-time collaboration. Thus, there is a necessity to automate the detection of changes in the real-world environment. An intelligent environment can use sensors to detect real-world changes -changes in places or in the state of every significant entity in the workplace (people or artifacts)- and learn about the users behavior to automatically detect the group's membership.

In this paper, we explore how intelligent environments can detect automatically the arrangement of participants in groups for unplanned and planned situations. That is, if students are required to participate in collaborative activities -either within the classroom or around campus-, the system could proactively and automatically set up a certain group collaborative environment before members explicitly request for it, or even before any kind of communication between them has been established.

To provide support to the students' activities, an intelligent system could take into account contextual information and estimate group membership. This information can enable the automatic construction of applications that effectively assist the group members in sharing, communicating and coordinating as they move and reorganize in synchronous and collocated collaborative learning activities.

The substantial contributions of this paper are:

- We define a list of requirements on basic contextual information about the entities that any intelligent system needs to automatically detect the creation of groups. (Section II)
- We propose a methodology for training an intelligent system and also for the frequency in performing group' membership estimations. (Section IV)
- We demonstrate that predicting groups' membership is technically viable and that it can be incorporated in CSCL applications. (Section V)

II. CONTEXT AWARENESS FOR DYNAMIC GROUP CREATION

We have used the framework for CSCL/CSCW system awareness proposed in [4] as starting point. This framework

classifies the contextual information into five main categories: (1) people and groups, (2) scheduled tasks, (3) the relationship between people and tasks, (4) the environment where the interactions take place and (5) the tasks and activities that have been completed.

In order to achieve our work's goals, we have considered information related to the first category, the identification of individuals and groups enriched with time and place information. This context information can be static -acquired only once- or dynamic, obtained periodically (every time it changes). Both static and dynamic information is always associated to the user Id.

We analyzed which contextual information was relevant for training a Machine Learning and we discarded information that did not improve recognition accuracy. For instance, we found a significant increase in accuracy by including in the contextual information all the Bluetooth devices (physical proximity) sensed for the user, instead of only the devices of the students that had previously collaborated with him. In the experiments, we confirmed that such information was crucial for the Machine Learning to be able to accurately identify the context (Table II).

We finally selected the following items as relevant for predicting group creation (Fig. 1):

- Time stamp: the time and the day of the week.
- User identifier: a unique Id based on the identity of the user and his mobile device. (e.g. username and Bluetooth MAC address).
- Place: based on the access point fingerprint [6], our system obtains information about where users are placed: classroom, cafeteria, library, etc.
- Neighborhood: list of sensed Bluetooth' MAC addresses.

III. CONTEXT ARCHITECTURE

The architecture of our system (Fig. 1) consists of three blocks; at the user side there is the Context Provider block - where information about the users' context is gathered-, and at the Intelligent System side we can find two other blocks: Context Collector -where contextual information is pre-processed- and Machine Learning -where contextual information is consumed-. We used XMPP as communication protocol between the user and the intelligent system [7].

At the user side, context information is gathered to generate a vector for that user and it is sent to the system periodically or every time a change occurs.

When the system receives new contextual information related to one user, the Context Collector pre-processes this information and generates inputs to the Machine Learning which proposes the users to take some actions (group creation) if it is capable to predict them.

The user triggers some actions related with group management: create, join, leave and destroy groups. All these actions are also sent to the Intelligent System, which uses this

data to learn about the user behavior -with who and when the collaboration is done-.

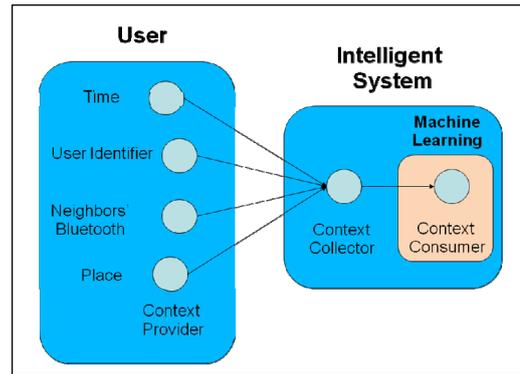


Figure 1. Context Architecture

A. Machine Learning for Group Prediction

In our effort to develop a mechanism to detect when collaboration among people starts in the real world and thus a group should be created in the computer-based environment, we trained a Weka workbench system to map contextual information into a group prediction. The Weka platform [3] is a framework where many different learning schemes are available, and where the results produced by each one can be assessed and compared. In addition, Weka provides a general API, so it can be easily embedded -as any other library- in our collaborative application.

We evaluated the results obtained with two different representative learning algorithms:

- Instance-based learning (IBL)
- Bayesian Network (BayesNet)

Both algorithms are very simple and have either few or no parameters to be tuned. They also produce classification models that can be easily interpreted. We chose IBL because it is a quite simple but robust learning algorithm, can tolerate noise and irrelevant attributes and can exploit inter-attribute relationships. On the other hand, we selected BayesNet as a baseline because it is a typical and well-known learning algorithm.

Instance-based learning (IBL) stores the training data. When a new input vector arrives a set of similar related instances is retrieved from memory and their corresponding outputs are used to predict the output for the new input vector [11].

Bayesian Network (BayesNet) is structured as a combination of a directed acyclic graph of nodes and links, and a set of conditional probability tables [12]. Nodes represent features or classes, while links between nodes represent the relationship between them. An estimation algorithm is used to create the conditional probability. We used the Simple Estimator algorithm, which estimates probabilities directly from the training data [13].

IV. EXPERIMENTS AND SIMULATIONS

This section presents the experimental setting of the simulations conducted in order to assess the automatic recognition of group formations.

A. Experimental setting

This study has been done at the EPSC campus of the UPC, an engineering school designed to use the collaborative learning and Project Based Learning models, so the classrooms are equipped with tables and chairs that facilitate teamwork. In addition, each student has a laptop equipped with WiFi and Bluetooth cards so they can interact with one or more peers to solve a given problem.

We studied the behavior of the students during a semester in order to identify the students' activities and the relevant contextual information needed to represent such activities. We recorded the time of each of the students' actions and some data regarding to the nature of such actions: the place, the list of students involved in a certain group activity, the subject they were attending at that moment, etc. We gathered information to develop a model that describes the students' activities and their characteristics.

The model used to perform our experiments considered a group of 30 students. These students represent a set of all the students in the Campus that can collaborate with each other in any different situation. Only 20 of these students follow the same degree course and the rest are others who can form a group out of the classroom context.

Six different places were selected as potential locations where the students could collaborate: classroom, library, group study room, cafeteria, vending machines and university grounds. Some of these places are not typical for a learning meeting but are daily meeting places for students where collaboration can occur spontaneously.

It was also identified that a particular student can be part of an average of 6 different groups during an academic semester. Of these groups, 4 are planned for regular work and the other 2 are spontaneous (unplanned) groups formed unexpectedly when the students have an opportunity to collaborate.

The regular class schedule of the considered set of students comprises six daily hours on mornings beginning at 8 o'clock. Moreover, the students have 6 different subjects (which are taught in the same classroom) but only 4 of them require teamwork.

Out of the lectures hours, students meet in different places in other to carry out the tasks requested by their teachers.

B. Training and Testing Data

We use data traces collected from the study of the students' behavior to train and test our system. This data represent the contextual information of the activities of the selected group of students during two weeks while they are collaborating. For example, when the users join a group, the traces contain the list of neighbors -sensing Bluetooth MAC addresses-, the place where they are located -deduced from the list of access points

sensed-, the date and time and the activity that is being done (group or non-group identifier).

TABLE I. CONTEXT VECTOR

Time stamp		User		Place	Neighbors	
Week day	Clock	Username	Bluetooth MAC		Bluetooth MAC	...
Monday	9h30	Anna	00:FF...	Cafeteria	00:FF...	...
Monday	9h45	Anna	00:FF...	Cafeteria	00:FF...	...

We transform the traces into input and output vectors (Table I) for our system. The inputs include the contextual information of a given activity from the point of view of one of the students, and the outputs are the group or non-group identifiers associated to such activity. This information is used to train the IBL and BayesNet systems which store this training data and use it later to predict the output at the arrival of a new contextual event.

The training data is a set of input vectors and the corresponding outputs. The items of an input vector are: a numerical value representing the time, a class corresponding to the day of the week, another class for the possible locations where collaboration could take place, and finally, a set of class items corresponding to the number of students that participate in the collaborative activity. These last items are Boolean values reflecting the presence or absence of Bluetooth signal from the students' devices.

The data used to train the IBL and BayesNet algorithms includes essentially the information sensed by the student from the moment he creates or joins a particular group until the time when the group is destroyed. Moreover, the Context Collector keeps historic data and can also generate inputs for the Context Consumer, for example, just before a group's creation and after its destruction.

The testing data is similar to the training one. It has similar input vectors but it includes all the contextual information that the student senses at any time. That is, not only in the time the student joins or leaves a given group. Moreover, the testing data has some changes in contextual information, such as the time of creation and destruction of the groups.

In our simulations we collected data during two weeks. We gathered a total number of 132 vectors during the first week for training. On the other hand, we collected another set of 214 vectors during the second week which was used for testing.

C. Training Process

The action of creating a group by a user is the start signal for the collection of training data (Fig. 2). This data is collected according to the following pattern of events:

- Group creation action: Upon arrival of this action, the training data is formed by the previous context event $C[n_0-1]$ associated with non-group state and the current context event $C[n_0]$ associated with the group created (Group X).

- Group operation period: During that time, a sequence of training data formed by the context events $C[n_0+1]$ associated with that group are generated periodically.
- Group destruction action: Finally, at the time of arrival of this action of closing the group, the training data of the end of the group is formed by the last context event $C[n_1]$ associated with the group and the next context event $C[n_1+1]$ associated with “non-group” state.

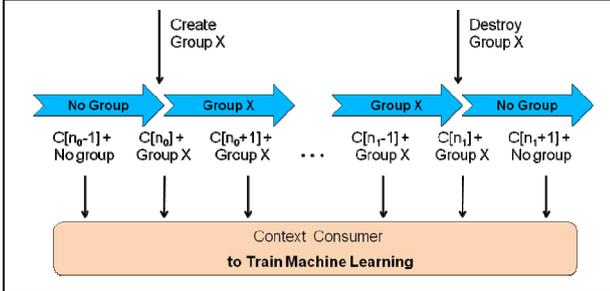


Figure 2. Training process

D. Estimating Process

The actions for performing groups’ estimations by the Intelligent System, follow the next pattern (Fig. 3): Since the last group estimation $E[n]$, a minimum and a maximum time are defined to make the next group estimation $E[n+1]$. New group estimation is performed depending on when significant contextual changes occur:

- In Fig. 3A, if one or more contextual changes $C[n]$ occur before the minimum time interval (five minutes), the estimation $E[n+1]$ will be just at this minimum time.
- In Fig. 3B, if contextual change $C[n]$ occurs between the minimum and maximum (15 minutes) times, the estimation $E[n+1]$ will be at that specific time.
- In Fig. 3C, if no change occurs, the estimation $E[n+1]$ will be at maximum time.

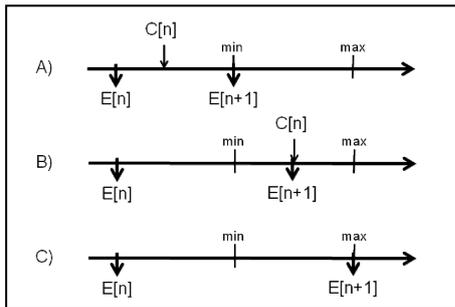


Figure 3. Estimating process

E. Experiment Evaluation

In order to make an initial assessment of the performance of our solution, the training data is used to create the learning model. Then, this data is used again for the system in order to evaluate its actual performance. The system computes the ratio

of correctly and incorrectly classified contextual events when its input is the same data used to construct the model.

Another method of evaluation was used: the cross-validation. This method uses only some parts of the training. Then other parts of the training data set are used to test the system.

Subsequently, we use a different data set to make a more realistic evaluation. It is used a testing data that has some differences respect the training data and that includes a wider context sensing. The training data is used to create the model and the new testing data is used to assess the performance of such model when it has to make a prediction based in new contextual information. Once again, the system computes the ratio of correctly estimated contextual events.

V. RESULT AND DISCUSSIONS

In this section, the results of our experiments are presented.

A. Results and Findings

Fig. 4 compares the accuracy in the prediction of the group formation for the BayesNet and the IBL algorithms. The IBL achieved the best results using the training and testing data sets, but when using the cross-validation testing method, both algorithms obtained very similar results.

Tests performed using the training data (Fig. 4A) are useful to confirm the validity of the constructed model. Due to the nature of the IBL algorithm and the great importance that it gives to the training data when estimating the output for a new input vector, it gets a 100% of success if it is tested using the training data. However, BayesNet gets the proper output in 85% of the events.

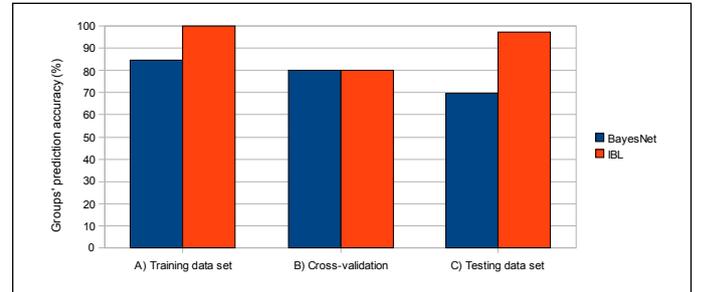


Figure 4. Accuracy of groups’ prediction of BayesNet and IBL

In the cross-validation method (Fig. 4B) the percentage of success usually decreases because the learning algorithm has less input information to train, in addition, the selected information might not be as relevant as the discarded one. From the cross-validation test results we can prove the importance of selecting appropriate time intervals for the collection of training data and thus justify the necessity to use the training process proposed in Fig. 2.

The results of the testing data set (Fig. 4C) show the accuracy of both algorithms when using a data set different from the training set. Using this evaluation method, we can see the actual performance of the system and the differences

between IBL and BayesNet. The obtained results show a clearly superior performance of IBL.

We also evaluated the quality of the contextual information selected. For instance, The results (Table II) prove that it is very important to include in the contextual vector all the MAC addresses of the Bluetooth devices sensed by the user and not only the MAC addresses of the students' devices that have previously formed a group with the user -known group members-.

TABLE II. EVALUATION OF THE QUALITY OF THE CONTEXTUAL INFORMATION SELECTED.

	Correct	Incorrect
All sensed MACs	208 (97.2%)	6 (2.8%)
Only known MACs	145 (67.8%)	69 (32.2%)

We assumed our model inputs were accurate. However, in practice, inputs might be estimated through other, possibly error-prone methods. To address this and validate our results, we tested our model's robustness by introducing noise into the contextual vector that we use as input data for our system.

We define an error as a highly significant change or a missed value in the contextual vector. In order to assess the system's robustness, we tested its sensitivity to each of the contextual vector items. For each item in the vector, we substituted only one item in all the testing events and tested the behavior of the system according to this new data set.

Results show that the system presents an outstanding robustness and that it can easily deal with noise in the input data. Table III shows that the system is still having a good performance if there are errors in a single item. Even when noise is introduced in two items, the system has an accuracy of 74% (last column in Table II).

TABLE III. SYSTEM'S ROBUSTNESS TO ERRORS.

	Accuracy %
No errors	97
Week day	86
Bluetooth list	89
Time	85
Place	96
Week day & Bluetooth list	74

Table IV shows the confusion matrix in the worst case (errors in the time item of the contextual vector) result obtained in the testing process when introducing errors in one single item of the contextual vector. In this table, $G1$, $G2$, $G3$ and $G4$ represent planned groups, while $GS1$ and $GS2$ represent unplanned groups. Lastly, $No G$ represents the no group activity state. In this test, the ratio of incorrectly classified events was approximately 14.5%. The matrix shows the number of correctly classified events in the diagonal. The

values out of the diagonal are the number of events incorrectly classified. From these results, we can deduce that the worst results are obtained because sometimes IBL confuses the 'No group' state with some of the 'Groups' ones. However, some of the incorrectly classified items are not a major problem because although groups are not properly detected, they do not confuse or disturb the students while they are working.

TABLE IV. CONFUSION MATRIX OF GROUP FORMATION PREDICTION

		Estimated Groups						
		$G1$	$G2$	$G3$	$G4$	$GS1$	$GS2$	$No G$
Real Groups	$G1$	26	0	0	0	0	0	0
	$G2$	0	31	0	0	0	0	0
	$G3$	0	0	10	0	0	0	5
	$G4$	0	0	0	24	0	0	0
	$GS1$	0	0	0	0	11	0	0
	$GS2$	0	0	0	0	0	3	0
	$No G$	10	7	0	9	0	0	78

Average error rate: 14.5 %

B. Discussion and lessons learned

In our tests, we observed the importance of selecting appropriately the time intervals to collect the training data. In addition, it was also noticeable the necessity of including in the training data some events with the contextual information when no groups were established. If such events were not included, the ratio of correctly classified events suffers a dramatic decrease.

On the other hand, we observed a slight improvement when multiple events reinforcing the fact of the existence of a group were introduced.

The results show that there is a difference in accuracy in the estimation of the group identifier depending on the item on which errors were introduced. We can see (Table II) how the system has the highest sensitivity to changes in times and day of the week. In contrast, the system presents very little sensitivity to errors or changes in places.

We also noticed that a particular group could only be formed if all the participants were taking part in the collaborative activity. As a result, if a certain group's member is not at the meeting place or his computer is inactive, the group cannot be correctly established and it needs to be manually created.

VI. RELATED WORK

Although there are many tools for providing some degree of automation and support for activities within a group, we have not found specific tools for providing the applications with automatic group awareness based on environmental contextual data.

In a Context-Aware Ubiquitous Learning scenario where students use mobile devices, such as laptops, smart phones or PDAs, wireless network connection and aware of the

surroundings (context-aware). Such as [8], the authors present an approach to the classroom context by identifying the students' activity. The main goal is to acquire physical interaction identifying and obtaining context services. This proposal facilitates a service to a single user taking into account his contextual information but it does not provide service adaptation. Our system proposes a complementary approach. We intend to provide services to a group of students instead of just one student. In addition, we adapt the service provided to the user dynamically, learning from new situations.

MCI-Supporter [9] is an application supporting collaborative learning methods in the classroom. It was conceived by first analyzing the best known collaborative learning practices trying to find out which are the real needs for mobility and face-to-face. Working groups have to be manually established by the teacher. In [10], the authors discuss on contextual information about groups (team learning context). They focus on workspace and social awareness and they even comment on team formation support: closed and open teams, joined and left manually and dynamic teams formed automatically by the system based on context and meta-information. . Our approach differs from the presented above mainly because we predict users' interactions taking into account the users past actions and our system does not require the users' intervention. Related works are less flexible, use only static information, do not learn and self-adapt depending on the circumstances, and they need the users to perform some actions.

VII. CONCLUSIONS

In this paper we presented the motivation and the problem of automating the incorporation of group awareness information into CSCL applications without shifting this burden to group participants or overloading the teacher.

In the light of this, we have proposed an Intelligent Environment System which based on contextual information from the students' laptops is able to automatically estimate group membership.

Finally, we evaluated the utility in terms of the rate of group recognition success. Simulations have shown that context-information can be effectively used as a basis for a middleware for an automatic and dynamic group management. Therefore we conclude that inferring group membership is technically viable and can be used in real world settings.

These results enable the construction applications that effectively assist group members in automatically share,

communicate and coordinate as they move and reorganize in synchronous and collocated collaborative learning activities.

ACKNOWLEDGMENT

We thank the Castelldefels School of Technology (EPSC) and the Distributed Systems Group (DSG).

REFERENCES

- [1] D. W. Johnson, R. T. Johnson, "An overview of cooperative learning", J. Thousand, A. Villa and A. Nevin (Eds), Creativity and Collaborative Learning; Brookes Press, Baltimore, 1994
- [2] G. J. Hwang, C. C. Tsai, S. J. H. Yang, "Criteria, Strategies and Research Issues of Context-Aware Ubiquitous Learning". Educational Technology & Society, Volume 11 (2), 2008.
- [3] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten, "The WEKA Data Mining Software: An Update"; SIGKDD Explorations, Volume 11, Issue 1. 2009
- [4] M. R. S. Borges, P. Brézillon, J. A. Pino, J. C. Pomerol: "Groupware System Design and the Context Concept", Proceedings of CSCWD Int. Con. Computer Supported Cooperative Work in Design (Selected papers) 2004
- [5] R. Messeguer, S. F. Ochoa, J. A. Pino, L. Navarro, H. Andrés Neyem, "Communication and coordination patterns to support mobile collaboration" Proceedings of CRIWG Int. Conf. Groupware: Design, Implementation, And Use, 2008
- [6] P. Bolliger. "Redpin - adaptive, zero-configuration indoor localization through user collaboration" Proceedings of MELT '08 Int. Workshop on Mobile Entity Localization and Tracking in Gps-Less Environments, 2008
- [7] J. Choi, S. Park, H. Ko, H. Moon, J. Lee, "Issues for Applying Instant Messaging to Smart Home Systems". Proceedings of ICCSA Int. Conf. on Computational Science and Its Applications: Part I, 2009.
- [8] J. Bravo, R. Hervás, G. Chavira, Ubiquitous Computing in the Classroom: An approach thorough Identification Process, Journal of Universal Computer Science, Vol. 11, No 9, 2005
- [9] N. Baloian, G. Zurita: "MC-Supporter: Flexible Mobile Computing Supporting Learning though Social Interactions", Journal of Universal Computer Science, Vol. 15, No. 9, 2009
- [10] A. Ferscha, C. Holzmann, S. Oppl "Team Awareness in Personalized Learning Environments", Proceedings Int. Conf. on Mobile Learning, 2004.
- [11] D. Aha, D. Kibler, and M. Albert, "Instance-based learning algorithms," Machine Learning, Boston: Kluwer Academic Publishers, pp. 37-66, 1991.
- [12] P. Langley G. H. John. Estimating continuous distributions in bayesian classifiers. Proceedings Conf. on Uncertainty in Artificial Intelligence, 1995.
- [13] R. Bouckaert, "Bayesian Network Classifiers in Weka", Technical Report, Department of Computer Science, Waikato University, Hamilton, NZ 2005.