Ounnas, A., Davis, H. C., & Millard, D. E. (2009). A Framework for Semantic Group Formation in Education. *Educational Technology & Society*, *12* (4), 43–55.

A Framework for Semantic Group Formation in Education

Asma Ounnas, Hugh C Davis and David E Millard
Learning Societies Lab, School of Electronics and Computer Science, University of Southampton, Highfield, Southampton, Hampshire, SO17 1BJ, UK // ao05r@ecs.soton.ac.uk // hcd@ecs.soton.ac.uk // dem@ecs.soton.ac.uk

ABSTRACT
Collaboration has long been considered an effective approach to learning. However, forming optimal groups can be a time consuming and complex task. Different approaches have been developed to assist teachers allocate students to groups based on a set of constraints. However, existing tools often fail to assign some students to groups creating a problem well known as "orphan students". In this paper we propose a framework for learner group formation, based upon satisfying the constraints of the person forming the groups by reasoning over semantic data about the potential participants. The use of both Semantic Web technologies and Logic programming proved to increase the satisfaction of the constraints and overcome the orphans' problem.

Keywords
Group Formation, E-Learning, Constraint Satisfaction, Collaborative Learning, Teams.

Introduction

Many approaches to learning and teaching rely upon students working in groups. Research in many disciplines has shown that learning within groups improves the students' learning experience by enabling peers to learn from each other. To form groups, students can be either allocated to groups randomly, self-select each other, or be appointed to a group by the teacher based on some criteria related to the collaboration goals. These criteria are usually expressed as a set of conditions, typically referred to as constraints, such as restricting the groups to be mixed in gender or skills (Ounnas, 2007b).

For the teacher, forming groups manually can be both difficult and time consuming. For this, researchers have been investigating several techniques for automating this process through the use of computer-supported group formation (CSGF). Similar to manual group formation, the challenges of CSGF lie in modeling the students' data, the teacher's constraints; and negotiating the allocation of students to groups to satisfy these constraints. However, existing tools often fail in allocating all students to groups, leaving some students unassigned to any group after the formation (Redmond, 2001), (Tobar, 2007). This problem is usually referred to as the orphan students problem.

In previous work (Ounnas, 2007b), we discussed some of the existing CSGF techniques in terms of the constraints and the selection of members in the groups. We also discussed the potential of using Semantic Web technologies (Berners-Lee et al., 2001) in providing meanings to the students' descriptions and constraints. In this paper, we propose a framework that is capable of efficiently automating the formation of students' groups by reasoning over the students' semantic data and the list of constraints specified by the teacher. We use the efficiency of both Semantic Web technologies and logic programming in modeling the problem of group formation as a constraint satisfaction problem (Kumar, 1992). The next section of the paper describes our motivation behind the research based on existing literature and the results obtained from a case study. We then describe the structure of the proposed framework and explain its components. Afterward, we discuss the evaluation of the proposed framework. Finally, we describe our future work for improving the performance of the framework and discuss some of the relevant issues with its evaluation.

Motivation

In order to understand the issues rising with forming efficient groups of learners, we analyzed the existing applications and efforts to automate this process. We identified the limitations of the existing tools and the need to design a framework that enables the delivery of well-formed groups based on constraints from a multidimensional space. To understand the nature of the possible constraints we carried an observational study with a class of undergraduate students at the University of Southampton.

Existing applications

In (Hoppe, 1995) Hoppe introduced an intelligent tutoring system that allows the learners to initiate a group formation when they have a problem (a learner-helper group). Based on the learners' models, the system displays a list of all potential peer learners that can help; the learner then selects a helper from the list, and the latter can accept or reject the invitation to help the learner. Parameters here are based on learning experience and competency criteria in the subject of the collaboration. In Mühlenbrock (2005) and (2006), context information such as the learner's geographical location from PCs, Phones, and PDAs were added to the model. Unfortunately, no evaluation of the application was provided by the authors.

A team from Osaka University in Japan (Ikeda et al., 1997) and (Inaba et al., 2000) introduce Opportunistic Group Formation (OGF) where an intelligent system detects the appropriate situation to start a collaborative learning session and sets up a learning goal for the learner. The system takes into account the modeling of learning goals for each learner. Based on individual goals as well as the whole group, the system negotiates with the agents of all the learners in order to come to an agreement and to form a learning group so that each member of the group can obtain some educational benefit. Unfortunately, there is no literature on the architecture of the developed systems or their evaluation.

In similar research (Soh et al., 2006), (Zhang, 2005), the authors introduce a multiagent intelligent system called I-MINDS where the instructor, each student and each group is represented by an intelligent agent. The student agent profiles the student and finds compatible students to form the student's "body group". The agents communicate, and form coalitions dynamically. For the group formation, each student agent bids to join its favorite group based on their previous performance in group activities. Therefore, the formation is constrained by the learner's previous performance. The collaboration, formation, and learner profiles updates are all processed in real-time. The student's profile in this research is built dynamically based on how active is the students during the real-time collaboration. At the beginning of this research (Soh, 2004), the author intended to provide a group formation based on positive interdependence and hence joint intentions where the students depend on each other for goal satisfaction, rewards, resources, division of labor, roles, and so on. However, when I-MINDS was introduced (Soh et al., 2006), the authors did not mention how this theory was put into practice, and only mentioned that their application considers the students' previous performance as a constraint to the formation. Soh et al. evaluated their IMINDS system by measuring its effectiveness in terms of its ease of usability by the instructor and the students. The group formation itself was evaluated against the performance of the teams, measured based on the teams' outcomes and their responses to a series of questionnaires that evaluates team-based efficacy, peer rating, and individual evaluation.

Also supporting Opportunistic Group Formation systems, in (Wessner and Pfister, 2001), the course author defines at which points in a distributed web based course a collaborative activity should occur. The system then uses knowledge about the collaboration context in real-time such as whether the student has performed this collaboration before, how often, and how fast in order to form appropriate groups. The formation here follows a self-selected approach. Although the authors did not present any results of evaluating their system, they mentioned that the comprehensibility of the group formation algorithms and the satisfaction of learning groups to be a key factor of the overall approach acceptance.

On recommending expert collaborators, Vivacqua and Lieberman (2000) introduces Expert Finder, an agent that automatically generates user models by classifying both novice and expert knowledge of the participants. The agent autonomously analyzes documents created in the course of routine work to rank the experts for recommendation to the learner who initiated the expertise search. For evaluation, the authors compare the results (the formed groups) generated from the system to manual generated results of the same participants' sample. This technique is frequently used in recommender systems (McDonald, 2001).

Redmond (Redmond, 2001) introduces a computer program to aid the assignment of students projects groups. This technique is used to form instructor-based group formation for all (part time and distance) learners in the class simultaneously. The students are grouped based on the time slot they prefer to do the group work in, and then allocate the projects to the groups based on the members' preferences in the group. The group formation is processed using a greedy algorithm where the program starts with the tightest constraint – the student with the fewest time slots rated highly – and tries to find a compatible group for them. This process repeats until all students have been assigned to a group. The formed groups are manually checked for even distribution of grades, and the students who are left unassigned are manually allocated to groups. To measure the efficiency of the program, the author introduced an evaluation formula that calculated the rating of group assignments by subtracting an unassigned penalty representing the program failure in assigned some students from the sum of all formed group overall rating.

In Christodoulopoulos (2007), the authors presented a web-based group formation tool that supports the instructor to automatically create homogeneous and heterogeneous groups based on up to three criteria and the learner to negotiate the grouping. The tool employs a clustering algorithm (Fuzzy C-means) for homogeneous grouping while heterogeneous groups are generated using Random Selection algorithm. For each student, the clustering algorithm gives the probability of the student belonging to each group. This helps the instructor to manually adjust the formation since the generated clusters may not be of same size. The probabilities enable swapping students who are unsatisfied with their allocation. The preliminary evaluation of the tool was satisfactory although it was tested on only 18 groups with one criterion (constraint on Learning Styles).

In Tobar et al (2007), the authors introduced a rule-based tool that aims at reducing the time teachers spend creating groups for learning. The tool takes into account the students' characteristics that are required by the rule (hard constraint that can not be violated). The characteristics available in this tool are taken form the IMS learner specification (Wilson, 2002). The results returned from the tool can be manually modified by the instructor. Unfortunately, there was no evaluation of the performance of this tool.

In Lugano et al (2004), the authors studied data from self-rated questionnaire together with statistics of the learners' real activity in a collaborative learning environment called EDUCOSM. The authors considered the students' motivation (learning goal orientation) and social skills (social group roles). The results of analyzing self-perception in actual behavior showed a low correlation of pre-test results with learning outcomes. The authors explained this observation by the high initial expectations being lowered later by factors such as high workload or technical problems with the system.

In de Faria et al (2006), the authors introduced an approach of forming groups for collaborative learning of computer programming. The groups were formed based on the students' programming style generated by a tool implemented to automatically assess the style of the programs submitted by the students. Analyzing the students' programs assists in finding characteristics that evidence significant differences such as program quality, which would be relevant enough to motivate the students to discuss them.

In Graf et al (2006), the authors propose a mathematical model for building heterogeneous groups based on the students' personality traits (group work attitude, interest for the subject, achievement motivation, self confidence, and shyness), their level of performance in the subject, and fluency in the language of instruction, where each of these attributes is ranked on a one to three scale. The authors use the Ant Colony Optimization algorithm to allocate each student to the most appropriate group that would maximize the diversity of that group while keeping the deviation between the groups minimum. The authors show that their approach is scalable (around 500 student) despite the problem being NP-hard.

In Cavanaugh et al (2004), the authors describe Team-Maker, a web-based system that aims at reducing instructors' time in allocating students to groups. The system takes some students characteristics such as gender, skills, and students' schedules, and the instructor's criteria for the creating of homogeneous or heterogeneous groups, and applies a Hill climbing algorithm to get the optimal solution. The authors show that the system outperforms manual group formation, but does not mention the complexity of the system or how good it performs as the number of constraints (instructor's criteria) grows.

In Wang (2007), the authors introduce a computer-supported heterogeneous grouping system called DIANA. The system uses a genetic algorithm to form fair groups in terms of heterogeneous grouping such that all groups have the same level of diversity. The system uses the students' characteristics (thinking styles) collected from questionnaires. It takes up to 7 variables and allocates 3 to 7 members per group. The evaluation of the research on a class of 66 students showed that DIANA performs better than random allocations to groups. Although, the authors did not discuss the complexity or scalability associated with the application of the algorithm. Table 1 shows a summary comparison of the discussed Computer Supported Group Formation applications. We refer to the applications that are not provided with a name with their first author name.

Table 1. Existing CSGF applications in e-learning

Formation features	Approach		Principle		Algorithm	Modeled students characteristics
	Self-selecting	Instructor Based	Opportun-istic	Simultaneous for all students in the class		
Hoppe	✓		✓	×	Rule/inference based	Knowledge in a specific domain
Inaba	✓		✓	×	Multi-agent System	Learning goal
Soh	✓		✓	×	Multi-agent System	Performance in previous teamwork
Wessner	×	✓	✓	×	Multi-agent System	Knowledge on student's state within the designed learning
Vivacqua	✓		×	×	Profile Matching	Expertise in a specific domain e.g., Java Programming skills
Redmond	×	✓	×	✓	Greedy algorithm	Preferred time slots and Preferred projects
DIANA	×	✓	×	✓	Genetic algorithm	Psychological variables (thinking styles) – but can take up to 7 variables
Team-Maker	×	✓	×	✓	Hill Climbing	Any variable
Graf	×	✓	×	✓	Ant Colony optimization	Performance and Personality traits
Tobar	×	✓	×	✓	Rule based	IMS LIP
Christodoulopoulos	×	✓	×	✓	Fuzzy C-Means	Knowledge and learning styles

Limitations of existing applications:

From the literature, we see that in terms of constrained group formation complexity (Ounnas, 2007):

Modeling

- Most systems only model a fixed set of parameters, which does not allow for the formation of different types of groups, and hence the implementation of different collaborative activities (only supports some types of teams).
- None of the existing efforts discuss the performance of the relative application in handling the group formation when the data about the user is incomplete, for example, if a new student with no record in the university joins the collaboration activity.

Constraint satisfaction

- Many systems use Opportunistic Group Formation (Wessner and Pfister, 2001), (Hoppe, 1995), (Soh et al., 2006), (Inaba et al., 2000), which does ensure satisfaction of the participants in the group through negotiation, but does not discuss the efficiency of the negotiation if all students in the class are grouped simultaneously. In addition to that OGF is usually more beneficial in short-term groups. In addition to this, these systems are based on self-selecting group formation (Hoppe, 1995), (Inaba et al., 2000), (Vivacqua and Lieberman, 2000), (Soh et al., 2006), which is not the most efficient approach in forming teams for learning, as it does not ensure balanced grouping.

- As observed in using existing group formation tools, another common problem in forming groups is "the orphans problem"; these are the students who remain unassigned to any group at the end of the formation. In existing tools, such as (Redmond, 2001) and (Tobar, 2007), this problem remains unsolved. Instead, most tools return the names of the orphans for the instructor to allocate them manually to some group, or rearrange the formation by swapping the orphans with other members, the fact that decreases the efficiency of the automated formation.
- Based on the reported results, most applications can only take a small fixed number of constraints. So far, DIANA seems to handle the highest number of constraints, which is currently limited to 7, and only for homogeneous grouping. We hypothesis that is fact is related to the limitation and complexity of the algorithms implemented in the tools and therefore rises issues on scalability of the systems.

Evaluation

- In addition to the luck of providing results on the performance of the applications in some of the literature, a limitation of most group formation applications is the exclusive reliance on the groups' performance measures indicators such as members' responses to questionnaires or post-tests to draw inference about the group formation system performance. From a learning viewpoint at least, group formation efficiency is clearly a multi-dimensional concept, which implies that multiple efficiency indicators besides perceived performance need to be employed. While different formation constraints might result in different formulas for calculating efficiency, these constraints can be related to group formation efficiency in a more abstract way. If so, consideration of defining this relation together with other group formation related measures is required.

Observational study

To match the growing need of forming groups with higher flexibility, we started analyzing what constraints do teachers consider when forming groups. We studied the possible students' features that can be relevant to forming different types of groups by investigating the available literature on collaborative learning theories (Ounnas, 2007b), and asking teachers what constraints they employ for different educational goals.

As a case study on group formation, we conducted an observational study with 67 undergraduate students taking a software engineering group projects course (SEG) in the School of Electronics and Computer Science at the University of Southampton. The students were manually grouped by the course organizers into 11 groups of 5 to 6 students, based on the following constraints:

- All groups have to be balanced in terms of the students' previous grades to ensure that all groups have an equal opportunity in performing well in the project.
- To avoid minorities, a female cannot be allocated to an all-male group to prevent her from being cast away by the members.
- International students from the same country can't be all members of the same group.

The module organizers used a script to allocate the students based on their marks, then manually swapped some of them to redistribute females and international students. To analyze the dynamics of the groups and how other criteria affect them, we distributed two questionnaires to the class:

Questionnaire (1): at the beginning of the course, we asked the students to fill in a form to get information about their previous experience in software engineering, teamwork, their gender, nationality (to detect minorities), and Belbin team roles to check which role can each student play within their group. Belbin roles are typically used in industry and training activities to discover the best roles a participant can play in a group (Belbin, 2004). There are 8 Belbin roles, and according to these roles; a balanced team is composed of:

- One leader: *Coordinator* (CO) or *Shaper* (SH), and not both in the same group to avoid conflicts,
- A *Plant* (PL): to stimulate ideas and insure creativity,
- A *Monitor/Evaluator* (ME) to maintain honesty,
- One or more *Implementers* (IM) to executed actions, *Team Worker* (TW) to ensure cooperation in the group, *Resource Investigator* (RI) to explore opportunities and secure resources, or a *Completer/Finisher* (CF) to ensure all tasks are completed on time.

Each person usually plays more than one Belbin role within the team. However, a member usually scores high in only one or two roles. In our study, we collected both the first and second roles for each student.

Due to some students dropping out of the course and others not filling in the questionnaires, we collected data from 9 groups out of the whole 11. Table 2 illustrates the results collected from questionnaire (1) showing Belbin roles in each group. The numbers in the cells demonstrate how many members in the group have that role as their strongest role, where the distribution of the latter is the average of the first and second strongest roles.

Table 2. Results of observational study (distribution of Belbin Roles)

Group	IM	CO	SH	PL	RI	ME	TW	CF
1	1			1	1	2	1	
2	3						1	
3	1		2		1	1	1	
4	1	1	1				1	1
5	1		1	1			3	1
6	4		1			1		
7	3	2	1					
8	1	1	1				2	
9	3			2	1			
Total	18	4	7	4	3	4	9	2

Questionnaire (2): at the end of the course, we distributed a 17 questions based evaluation form where the student is asked to rank the key elements that measure their group performance, dynamics, and the individual satisfaction with the group work on a 1 to 6 scale. In particular, we analyzed creativity, motivation, leadership, group cohesion, satisfaction with contribution of members and the group output.

Given that in some groups, only one or two students returned the questionnaire, we were only able to use the data from groups 1, 3, 4, 5, 7, and 8. Table 2 shows these groups in shaded color. The results showed that the majority of the groups were satisfied with the group output (the software), and no members (minorities) were isolated which can be related to the fact that the teams were formed to be balanced in terms of grades and gender. However, constant conflicts were reported in the groups that had no leader or more than one strong leader (groups 1, 7 and 8). The groups that did not have a plant member such as groups 3 reported a lack of innovation, while groups with a Plant responded well (group 1 and 5).

From the study, we observed the relation and effect of possible group formation constraints on the students' perceived satisfaction. However, despite the benefits of having a number of constraints in achieving the educational goal of the collaboration, negotiating the students' allocation to groups manually gets more complex and time consuming as the number of constraints grows, even if the teacher had the required data about the student.

The case study provided us with an initial understanding of the domain characteristics and relevant problems in forming groups, which support our findings from analyzing existing literature in the area.

Our analyses from both the literature and the case study yielded various ideas for possible computer support in both modeling the constraints and evaluating the formation of groups.

Semantic group formation framework

To overcome the complexity of allocating students to groups, we propose a framework to assist the teacher in forming groups based on their chosen set of constraints. The framework handles the group formation process based on the following concepts:

- Modeling the students' features: we model a large range of features that can be considered for different group formations using the concept of Semantic Web ontologies (Berners-Lee et al., 2001), which can form a reliable dynamic learner profile (Ounnas, 2007b). In this context, semantic modeling provides meaningful descriptions of the students and the relationships between them.

- Negotiating the group formation: we express the students' allocation problem as a *Constraint Satisfaction Problem (CSP)* (Kumar, 1992). The negotiation process of allocating each student to their most appropriate group can then be handled by a constraint satisfaction solver.

We emphasize that, in this research, we are not concerned with proving that any particular set of constraints leads to better results in terms of the performance of the groups; neither do we claim that any particular algorithm leads to best grouping. Figure 1 shows an overview structure of the framework, which is based on the following components:

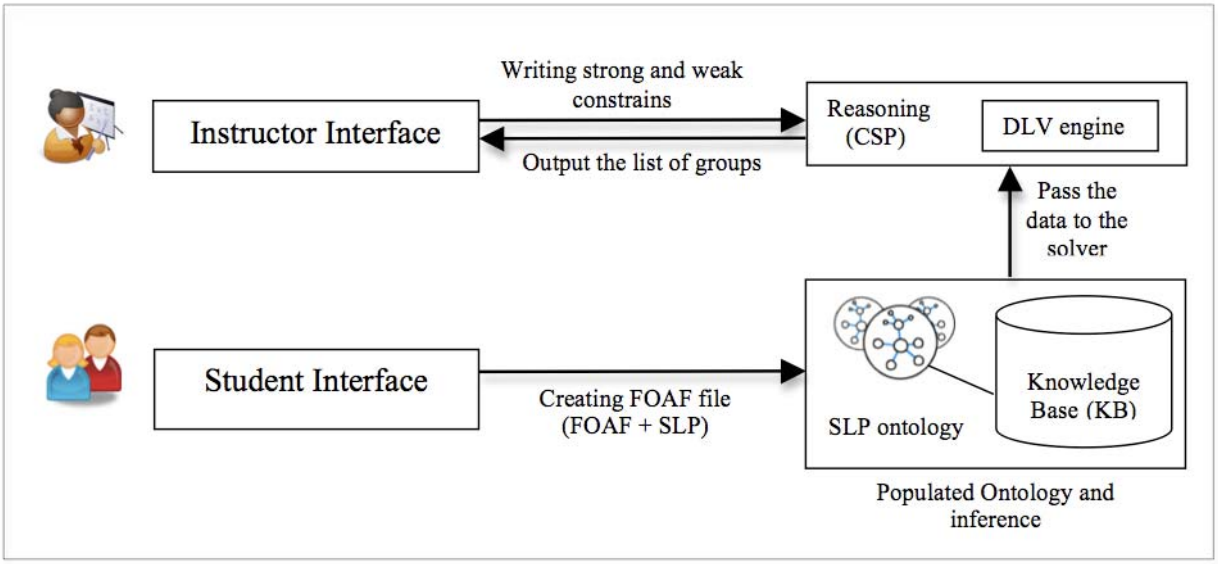

Figure 1. Semantic group formation framework

The Student Interface

The student can enter their data through a web-based form composed of four parts: the student's personal data, a list of their friends, their interests and preferences, information about their course such as the modules they are taking, and a list of their friends/colleagues taking similar courses. The students can update their data at any time. To avoid the form style in the interface, and to make it more desirable to students to update regularly, we aim to make it a Web 2.0 like interface such as the ones used by social networking sites.

The Ontology

We created an ontology called *Semantic Learner Profile* (referred to in this paper as SLP). The ontology is available at http://users.ecs.soton.ac.uk/ao05r/slp.owl. The ontology extends the Friend Of A Friend ontology (FOAF, available at http://xmlns.com/foaf/spec/), an existing ontology that describes people for building communities and social groupings. The learner's characteristics that the ontology describes were chosen based on a comparison of existing learner profiles such as PAPI, IMS LIP and eduPerson (Ounnas, 2006). Therefore, the ontology describes a large range of student's personal, social, and academic data such as learning styles, preferred modules, topics, and collaborators (Ounnas, 2007a). The semantic representation of these data, to which the instructor constraints can be mapped to, allows inferences to generate more data. This feature of using semantics enables the framework to handle incomplete data in a more effective way (this is explained in more details in section 4). Since the FOAF ontology is currently very popular (Ding et al., 2005), employing it would allow using data from any other ontology that can be mapped to FOAF.

Once the student submits the profile data through the student interface, an RDF file is created (FOAF + SLP). The file is processed using Jena, a Semantic Web inference engine (Carroll et al., 2004), and instances of the ontology are then stored in a database. Figure 2 shows an example of a student's FOAF file extended with the SLP ontology. In this figure, the file holds information about the student's name, gender, Belbin role, preferred module, topics of interest, and friends (classmate).

```
<rdf:RDF
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:foaf="http://xmlns.com/foaf/0.1/"
     xmlns:slp="http://www.ecs.soton.ac.uk/~ao05r/slp.owl">
     <foaf:Person rdf:nodeID="asma">
          <foaf:name>Asma Ounnas</foaf:name>
          <foaf:gender>Female</foaf:gender>
          <slp:belbin>Implementer</slp:belbin>
          <slp:interest>e-Learning</slp:interest>
          <slp:interest>Semantic Web</slp:interest>
          <slp:preferredModule>CS1004</slp:preferredModule>
          <slp:classmateOf><foaf:name>Ilaria
                    Liccardi</foaf:name></slp:classmateOf>
    </foaf:Person>
 </rdf:RDF>
```

Figure 2. An example student FOAF+SLP profile

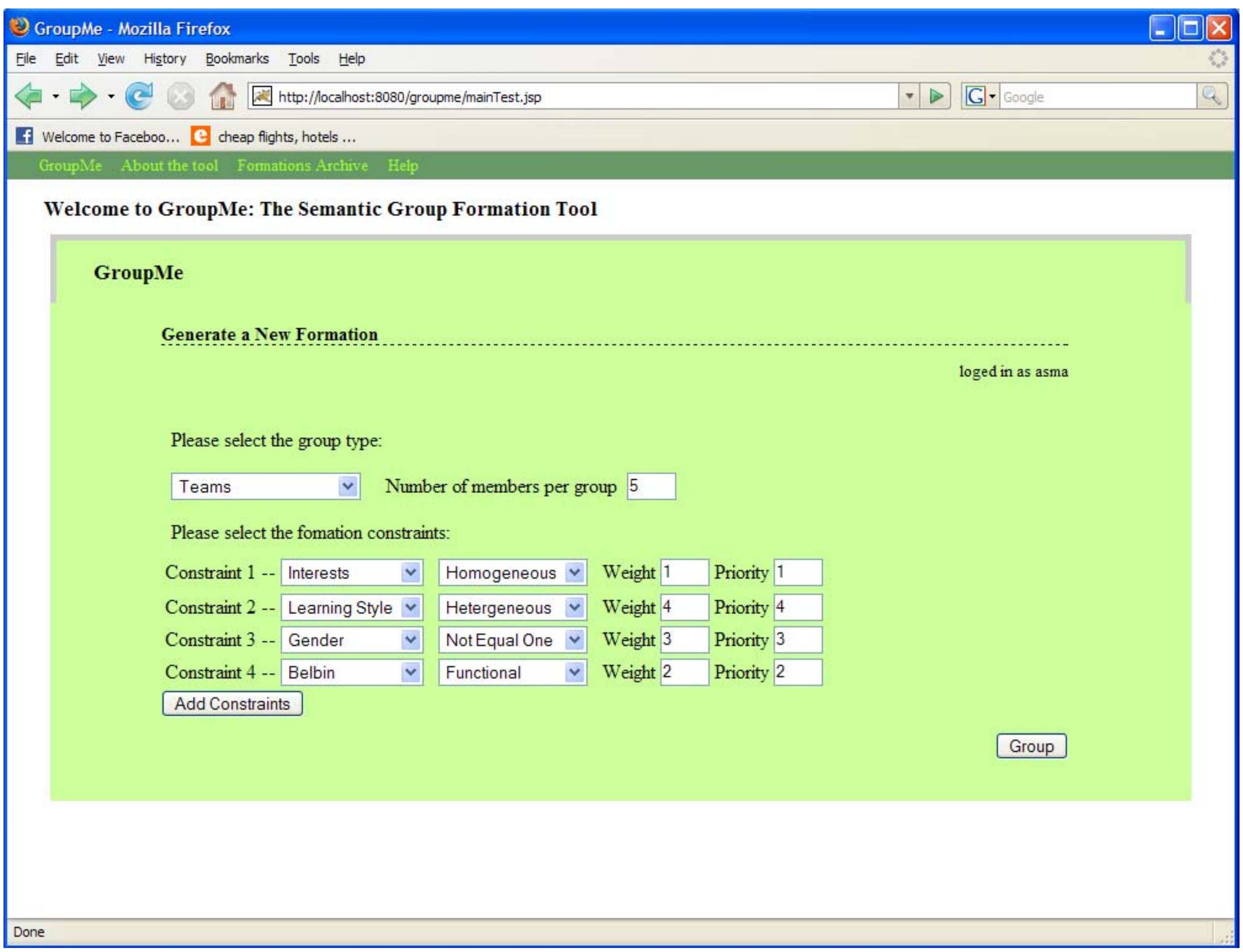

Figure 3. An example student FOAF+SLP profile

The Instructor Interface

Through this web-based interface, the instructor can select which constraints they care about for the formation they are initiating such as the student's gender, their team role, their learning style and so on. So far, it is possible to add constraints on the following characteristics: gender, nationality, age, previous marks (for competency), team roles, interests, and learning styles. Other characteristics such as preferences and trust values between participants will be implemented soon.

For each constraint, the teacher can select the condition on this characteristic, for example the groups are to be heterogeneous in gender and mark distribution, homogenous in learning style, and so on. They are also provided with an option that enables them to set a priority value for each constraint. Ranking the importance of the constraints to the group formation enables the application to manage compromises based on these priorities as explained in the next section.

The group generator

As the core component of the framework, the group generator is responsible for negotiating the allocation of students into groups. The generator is based on a DLV solver, an implementation of disjunctive logic programming, used for knowledge representation and reasoning. DLV's native language is *Disjunctive Datalog* extended with constraints, true negation and queries (Leone et al., 2006), where Datalog is a query and rule language for deductive databases that syntactically is a subset of Prolog.

DLV performs a simple forward checking algorithm (Kumar, 1992) on the data provided by the learners and the instructor in order to allocate students to groups. The students' data is automatically transformed from the SQL database to an *Extensional Database (EDB)* in the form of predicates that the solver can read as an input. Figure 4 shows an example of this knowledge base where predicates of the form "*student(name,role,gender).*" show the student's family name, Belbin role, and gender.

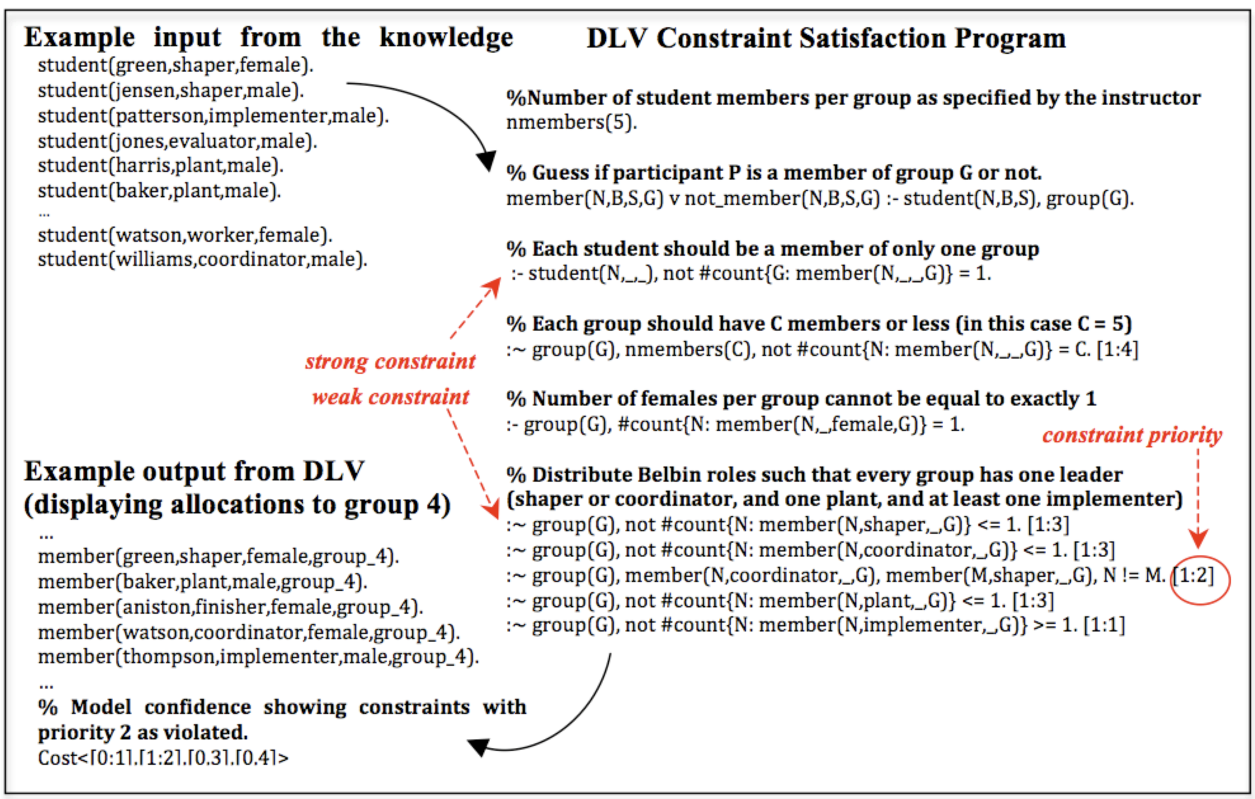

Figure 4. Example DLV program

Through the instructor interface, we feed the list of constraints specified by the teacher. The constraints are written into a DLV program, modeled as a constraint satisfaction problem as illustrated in figure 4. Here, we use two types of constraints: *strong constraints* and *weak constraints* (Buccafurri, 1997). The former are used to specify the conditions that have to be satisfied by the system in all cases. An example of these constraints would be that each student can be a member of only one group.

The weak constraints are used to specify the conditions that are preferably satisfied, but can be violated if the system would not be able to find a solution otherwise. These constraints are given a priority level according to their importance in the group formation through the instructor interface. For example, in figure 4, the instructor considers having only one leader (shaper or coordinator) in each group to be more important than having an implementer in each group by assigning these constraints priority levels 3 and 1 respectively.

Depending on the data provided and the constraints, DLV outputs more than one solution to the problem (i.e. more than one grouping of the students). Each solution is called *a model*. The optimal model is hence the best grouping of students in relation to the given constraints and input data. The best model is calculated as an objective function that minimizes the number of violated constraints.

Unlike other computer supported instructor-based group formation tools (Redmond, 2001) (Tobar, 2007), our approach does not leave any student orphans. Based on the negotiation of the constraints satisfaction through optimization, all students are allocated to some group, even if some constraints are violated. The best model is computed and the confidence of the computation (formation) is returned to the instructor: For instance, if the instructor wants only one leader per group to avoid conflicts, and gave the constraint priority 2, but the number of leaders is larger than the number of students; then some of the groups will have more than one leader. Here, a constraint of that priority is violated. Hence, the confidence of how good is the group formation is decreased. Together with the model, the confidence is computed in terms of violated constraints, and then returned as an output solution.

DLV outputs the model as a list of predicates. Figure 4 illustrates an example output predicates of the form "*member(name,role,gender,group)*" showing the students' family name, Belbin role, gender, and the group they are allocated to. This output data is then stored in an SQL database and then returned to the instructor through the instructor interface as a list of groups. To ensure a good practice interface design, the interface output can be dynamically manipulated by the instructor in case modifications or swapping students around the groups is preferred.

Evaluation

As mentioned in the limitations of current systems, a well-defined approach to evaluating group formation has not been introduced yet. Hence, to evaluate the quality of the generated group formation, we defined a set of metrics that measure how good the formation is in terms of the constraints satisfaction rate of all the groups of students (Ounnas, 2007c). So far, we used the group formation framework with a range of strong and weak constraints and a different number of variables (i.e. students' characteristics in which the formation is constrained). We used the framework to allocate students to groups within two courses in the University of Southampton including the software engineering course on the following year. However, since the instructors of these courses had only a maximum of three constraints (on previous marks, gender, and international students), the framework returned a best model in both cases with no violations in both courses.

To monitor the performance of the proposed framework, more evaluation scenarios were set. We run different scenarios of forming groups on simulated classes of 100 students. The simulated data was based on the population statistics collected from our observational study and confirmed from the UK Higher Education Statistics Agency (HESA: http://www.hesa.ac.uk/). The framework was tested with various constraints, different in content, number, and priorities. From the evaluation, we observed that our system can take up to 11 constraints before the solver starts taking a long time to process the groups. Although the results are still more efficient than any of the existing tools, the maximum number of constraints can be enhanced with the addition of some heuristics to the solver's algorithm. With the heuristics implemented, the solver will be prevented from running out of time during the computation.

The group formation quality here is measured against the satisfaction of the constraints chosen by the teacher. This includes calculating the average of how many constraints have been satisfied for each group. The formation quality is then calculated in terms of and the average and standard deviation of the constraints' satisfaction of each group, and therefore for the cohort in general. Formulas for calculating the formation quality are detailed in our evaluation framework, as described in (Ounnas, 2007c). In this context, we don't measure the quality of the constraints themselves if they will lead to a good formation or not, neither do we take the students' satisfaction with the allocations. Since teachers are given the freedom to choose the constraints and their importance, we trust they will choose the constraints that best fit their students' needs and the collaborative task they are trying to achieve through the group formation. It is a part of our future work however, to evaluate the quality of different constraints in relation to different sets of data. Through many runs of experiments, the quality of the constraints can be measured as described in our evaluation framework (Ounnas, 2007c).

We also tested the system's performance with incomplete data, by deleting data at random with an equal distribution on each characteristic considered for that group formation. Results showed that the systems still performs well (for example from no violations to 2 violations) when the data is down to 50% incomplete for a formation with 3 constraints, a moderate number of constraints for creating learning groups. However, as the number of constraints grows, the performance decreases accordingly when the data is incomplete. Table 3 shows the results of forming groups with incomplete data with a different number of constraints. For each scenario (with a different number of constraints), we calculated the number of constraints violations (NCV), and the group formation quality (FQ) as explained in our group formation evaluation metrics framework in Ounnas (2007c). The cells with no results show the case where the solver runs out of time during the computation.

Table 3. Results of forming groups with complete and incomplete data

Data Completeness	**With 2 constraints**		**With 3 constraints**		**With 7 constraints**		**With 9 constraints**		**With 10 constraints**	
	NCV	**FQ**	**NCV**	**FQ**	**NCV**	**FQ**	**NCV**	**FQ**	**NCV**	**FQ**
100%	0	1	0	1	0	1	0	1	0	1
90%	2	0.91	2	0.94	3	0.83	3	0.9	5	0.83
70%	2	0.91	2	0.94	3	0.83	4	0.89	6	0.82
60%	2	0.91	2	0.94	3	0.83	5	0.77	7	0.80
50%	2	0.91	2	0.94	10	0.74	12	0.74	-	-
30%	3	0.86	6	0.74	11	0.73	-	-	-	-

These results were collected without using the semantic inference feature of the framework as it is still being implemented. We expect the results to improve, as the main use of the semantic inference is to handle incomplete data through the use of domain ontologies and deduction rules.

Future work

For future work of this research, we intend to test the framework with more challenging scenarios by increasing the number of students in the class, and the number and context of the constraints. Since groups can also be generated from social networks, a range of the constraints will be based on the social connections between the learners (established through FOAF relationships), such as forming groups of students who do not know each other directly, or forming groups of student who collaborated with each other in a previous task.

So far we only used the framework to run using complete data about the students. A more challenging task will be to generate groups from incomplete data. Since our framework is based on Semantic Web technologies, we intend to empower it to handle incomplete data using these technologies. For our future work, we plan to add a module to the architecture of the framework that mines data from web pages and connect it to the ontology and a set of deduction rules to infer the missing data from the knowledge base. In this case, if the student does not provide the data needed for the group formation constraints, then the system will substitute the necessary data and subsequently feed it to the DLV solver. For example, if the information about whether student John is a leader or not is missing, and we know from John's web page that he is a captain of the football team, then we can infer that John is a leader; or if we are grouping student by skills, and we don't know Sarah's skills, but we know that Sarah has a high grade in discrete mathematics and Sarah has a high grade in Logic then we can infer that Sarah will perform well in formal methods.

Once the framework is refined with deduction rules, the evaluation of its performance with incomplete data will be compared to its performance with complete data (and no deduction rules), and its performance with incomplete data (and no deduction rules).

Conclusion

In this paper, we discussed the challenges associated with the manual and automated formation of groups in Education by analyzing existing applications in the field, and running an observational study. Based on our findings, we proposed a framework for group formation based on Semantic Web technologies and constraint satisfaction optimization to assist teachers with effectively defining groups of students based on a set of constraints of their choice. Unlike existing Computer Supported Group Formation tools, the approach we followed does not leave any student unassigned to groups. Instead, we employ strong and weak constraints to negotiate the students' allocations to groups and report the confidence of the generated solution. We evaluate the group formation based on the quality of the generated groups in terms of constraint satisfaction, and the robustness of the formation in case of incomplete data. For future work, we intend to evaluate the potential of employing Semantic Web technologies in compensating incomplete data through inference in deduction rules and domain ontologies.

References

Belbin, R. M. (2004). *Management Teams: Why They Succeed Or Fail*, Elsevier Butterworth-Heinemann.

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web. *Scientific American, May*, retrieved September 1, 2009, from http://www.scientificamerican.com/article.cfm?id=the-semantic-web.

Buccafurri, F., Leone, N., & Rullo, P. (1997). Strong and weak constraints in disjunctive Datalog. *Lecture Notes in Computer Science, 1265*, 2 -17.

Carroll, J. J., Reynolds, D., Dickinson, I., Seaborne, A., Dollin, C., & Wilkinson, K. (2004). Jena: Implementing the Semantic Web Recommendations. *Proceedings of ACM WWW'04*, New York, USA, 74-83.

Cavanaugh, R., Ellis. M., Layton, R., & Ardis, M. (2004). Automating the Process of Assigning Students to Cooperative-Learning Teams. *2004 American Society for Engineering Education Annual Conference & Exposition*, June 20-23, 2004, Salt Lake City, Utah.

Christodoulopoulos, C. E., & Papanikolaou, K. A. (2007). A Group Formation Tool in an E-Learning Context. *Proceedings of the 19th IEEE ICTAI'07*, 117-123.

de Faria, E.S.J., Adan-Coello, J.M., & Yamanaka, K. (2006). Forming Groups for collaborative Learning in Introductory Computer Programming Courses Based on Students' Programming Styles: An Empirical Study. *Proceedings of the ASEE/IEEE Frontiers in Education Conference*, San Diego, CA, USA, 6-11.

Ding, L., Zhou, L., Finin, T., & Joshi, A. (2005). How the Semantic Web is Being Used: An Analysis of FOAF Documents. *Proceedings of 38th International Conference on System Sciences*, Hilton Waikoloa Village, Island of Hawaii.

Graf, S., & Bekele, R. (2006). Forming Heterogeneous Groups for Intelligent Collaborative Learning Systems with Ant Colony Optimization. *Lecture Notes in Computer Science, 4053*, 217-226.

Hoppe, H. U. (1995). Use of multiple student modeling to parameterize group learning. *Proceedings of the 7th World Conference on Artificial Intelligence in Education* (AI-ED 95), AACE, Charlottesville VA, 234-241.

Ikeda, M., Go, S., & Mizoguchi, R. (1997). Opportunistic group formation. *Proceedings of AI-ED 97 Artificial Intelligence in Education: Knowledge and Media in Learning Systems*, Amsterdam: IOS, 166-174.

Inaba, A., Supnithi, T., Ikeda, M., Mizorguchi, R., & Toyoda, J. (2000). How Can We Form Effective Collaborative Learning Groups? *Proceedings of the 5th International Conference on Intelligent Tutoring Systems*, 282-291.

Kumar, V. (1992). Algorithms for Constraint Satisfaction Problems: A Survey. *AI Magazine*, *13* (1), 32-44.

Leone, N., Pfeifer, G., Faber, W., Deiter, T., Gottlob, G., Perr, S., & Scarcello, F. (2006). The DLV system for knowledge representation and reasoning. *ACM Transactions on Computational Logic, 7* (3), 499–562.

Lugano, G., Nokelainen, P., Miettinen, M., Kurhila, J., Tirri, H. (2004). On the Relationship between Learners' Orientations and Activity in CSCL. *Proceedings of the IEEE International Conference on Advanced Learning Technologies* (ICALT'04), Los Alamitos: IEEE Computer Society Press, 759-761.

McDonald, D. W., & Ackerman, M. S. (2000). Expertise Recommender: A Flexible Recommendation System and Architecture. *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, New York: ACM, 231-240.

Mühlenbrock, M. (2005). Formation of learning groups by using learner profiles and context information. *Proceedings of the 12th International Conference on Artificial Intelligence in Education*, Amsterdam, The Netherlands.

Mühlenbrock, M. (2006). Learning group formation based on learner profile and context. *International Journal on E-Learning, 5* (1), 19-24.

Ounnas, A., Liccardi, I., Davis, H. C., Millard, D. E., & White, S. A. (2006) Towards a Semantic Modeling of Learners for Social Networks. I*nternational Workshop on Applications of Semantic Web Technologies for E-Learning* (SW-EL) at the AH2006 Conference, 20-23 June, Dublin, Ireland.

Ounnas, A., Davis, H. C., & Millard, D. E. (2007) Semantic Modeling for Group Formation. *Workshop on Personalisation in E-Learning Environments at Individual and Group Level* (PING) at the 11th International Conference on User Modeling UM2007, Corfu, Greece.

Ounnas, A., Davis, H. C., & Millard, D. E. (2007). Towards Semantic Group Formation. *Proceedings of The 7th IEEE International Conference on Advanced Learning Technologies* (ICALT 2007), Los Alamitos: IEEE Computer Society Press, 825-827.

Ounnas, A., Millard, D., & Davis, H. (2007). A Metrics Framework for Evaluating Group Formation. *Proceedings of ACM Group'07*, Sanibel Island, Florida, USA, 221-224.

Redmond, M.A. (2001). A computer program to aid assignment of student project groups. *Proceedings of the 32nd SIGCSE Technical Symposium on Computer Science Education*, Charlotte, NC, USA.

Soh, L. K. (2004). On Cooperative Learning Teams for Multiagent Team Formation. *Technical Report WS-04-06 of the AAAI'2004 Workshop on Forming and Maintaining Coalitions and Teams in Adaptive Multiagent Systems*, July 26, San Jose, CA, 37-44.

Soh, L. K., Khandaker, N., Liu, X., & Jiang, H. (2006). A Computer-Supported Cooperative Learning System with Multiagent Intelligence. *Proceedings of the 5th international joint conference on Autonomous agents and multiagent systems*, New York: ACM, 1556-1563.

Tobar, C. M., & de Freitas, R. L. (2007). A support tool for student group definition. *The 37th ASEE/IEEE Frontiers in Education Conference*, October 10-13, Milwaukee, WI, USA.

Vivacqua, A., & Lieberman, H. (2000). Agents to assist in finding help. Proceedings of the ACM Conference on Computers and Human Interface (CHI-2000), the Hague, Netherlands, 65–72.

Wang, D. Y., Lin, S. S. J., & Sun, C. T. (2007). DIANA: A computer-supported heterogeneous grouping system for teachers to conduct successful small learning groups. *Computers in human Behaviors, 23* (4), 1997-2010.

Wessner, M., & Pfister, H. R. (2001). Group formation in computer-supported collaborative learning. *Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work*, New York: ACM, 24–31.

Wilson, S., & Jones, P. R. (2002). *What is...IMS*- IMS Learner Information Packaging? standards briefings series. c. JISC.

Zhang, X., Soh, L. K., Liu, X., & Jiang, H. (2005). Intelligent Collaborating Agents to Support Teaching and Learning. *IEEE International Conference on Electro Information Technology*, May 22-25, Lincoln, NE.