A Framework for Semantic Group Formation

Asma Ounnas
School of Electronics and Computer Science
University of Southampton
ao05r@ecs.soton.ac.uk

Hugh Davis
School of Electronics and Computer Science
University of Southampton
hcd@ecs.soton.ac.uk

David Millard
School of Electronics and Computer Science
University of Southampton
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.

1. 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.

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 [1], [2]. This problem is usually referred to as *orphan students problem*.

In previous work [3], we discussed 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 [4] 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 [5]. The next section of the paper describes our motivation behind the research based on the results obtained from a case study. Section 3 describes the structure of the proposed framework and explains its components. Section 4 describes our future work in improving the performance of the framework and discusses some of the relevant issues with its evaluation.

2. Motivation

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 [3], 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, gender, nationality (to detect minorities), and Belbin team roles to check which role can each student play within their group [6]. There are 8 Belbin roles. 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), *Team Worker* (TW), *Resource Investigator* (RI) or *Completer/Finisher* (CF) to make things happen.

Table 1. 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

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 1 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.

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 1 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. Another common problem in forming groups (manually or using existing CSGF tools) is "*the orphans problem"*; these are the students who remain unassigned to any group at the end of the formation. In existing Computer Supported Instructor-based Group Formation tools [1], [2], this problem remains unsolved. Instead, the 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. The case study provided us with an initial understanding of the domain characteristics and relevant problems, which yielded various ideas for possible computer support in both modeling the constraints and evaluating the formation.

3. 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 [4], which can form a reliable dynamic learner profile [3]. 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)* [5]. The negotiation

process 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:

3.1. 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, and information about their course such as the modules they are taking. The students can update their data at any time.

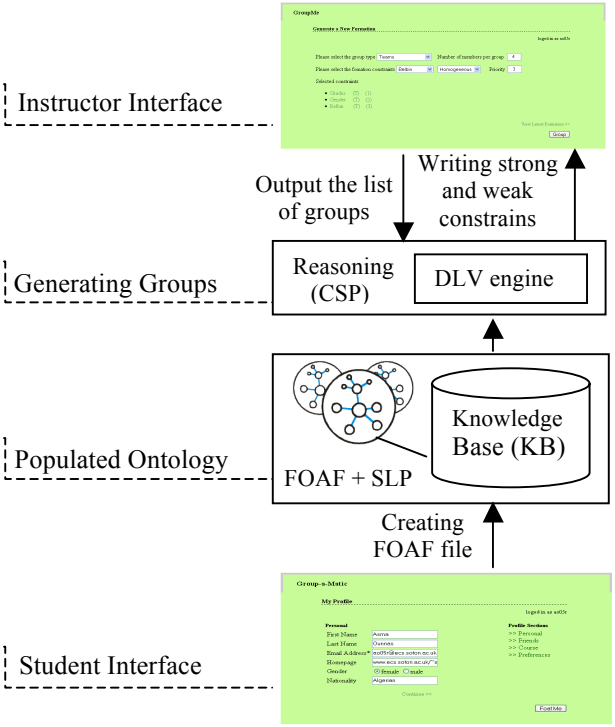

Figure 1. Semantic group formation framework

3.2. The Ontology

We created an ontology called *Semantic Learner Profile* (SLP[1]) that extends friend of a friend (FOAF[2]), 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. Therefore, the ontology describes a large range of student's personal, social, and academic data such as learning styles, preferred modules, topics, and collaborators [3]. 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).

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 [7], and instances of the ontology are then stored in an SQL 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

3.3. The Instructor Interface

Through this web-based interface, the instructor can select which constraints they care about for the formation they are initiating. They are 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.

3.4. 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*[3] extended with constraints, true negation and queries [8].

[1] http://users.ecs.soton.ac.uk/ao05r/slp.owl
[2] http://xmlns.com/foaf/spec/

[3] Datalog is a query and rule language for deductive databases that syntactically is a subset of Prolog

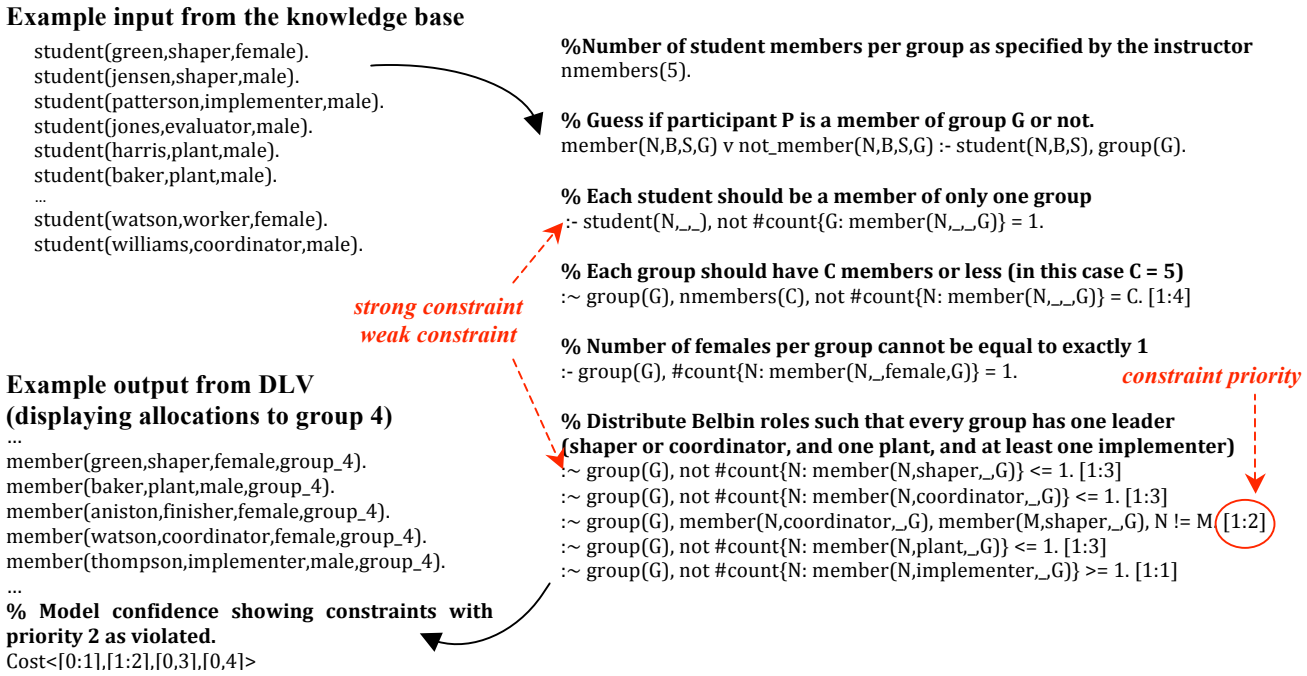

Figure 3. Example DLV program

DLV performs a simple forward checking algorithm [6] 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 3 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

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 3. Here, we use two types of constraints: *strong constraints* and *weak constraints* [9]. 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 3, 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 [1], [2], 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 3 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.

3.5. Evaluation

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 [10]. So far, we used the 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 allocated students to groups within two courses in the University of Southampton. However, since the instructors of these courses had only a maximum of three constraints (previous marks, gender, and international students), the framework returned a best model in both cases with violation of one constraint for only one group in both courses. To monitor the performance of the framework, more challenging evaluation scenarios are set as future work of the research.

4. Future work

To evaluate the proposed framework, we intend to run different scenarios on simulated classes of students. The simulated data will be based on the population statistics collected from our observational study. The framework will be tested with various constraints, different in content and number. 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).

5. Conclusion

In this paper, 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 CSGF tools, the approach we followed does not leave any student unassigned to groups. Instead, we employ strong and weak constraints to negotiate the students' allocation to groups and report the confidence of the generated solution. In our future work, we intend to 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.

6. References

[1] Redmond, M.A., A computer program to aid assignment of student project groups, In Proceedings of ACM SIGCSE, USA, 2001.

[2] Tobar, C.M., de Freitas, R.L. A support tool for student group definition. In Proceedings of the 37th ASEE/IEEE Frontiers in Education Conference. 2007.

[3] Ounnas, A., Davis, H. C. and Millard, D. E. Towards Semantic Group Formation. In Proceedings of The 7th IEEE ICALT'07, Niigata, Japan. 2007, pp. 825-827.

[4] Berners-Lee, T., J. Hendler, and O. Lassila, The Semantic Web, in Scientific American. 2001.

[5] Kumar, V. Algorithms for Constraint Satisfaction Problems: A Survey. In AI Magazine, Vol 13. Issue 1, 1992, pp. 32-44.

[6] Belbin, R. M. *Management Teams: Why They Succeed Or Fail*, Elsevier Butterworth-Heinemann, 2004.

[7] Carroll, J. J., Reynolds, D., Dickinson, I., Seaborne, A., Dollin, C., Wilkinson, K., Jena: Implementing the Semantic Web Recommendations. In Proceedings of ACM WWW'04, New York, USA, 2004, pp74-83.

[8] Leone, N., Pfeifer, G., Faber, W., eiter, T., Gottlob, G., Perr, S., Scarcello, F. The DLV system for knowledge representation and reasoning. In *ACM Transactions on Computational Logic (TOCL), Vol 7, Issue 3,* 2006, pp. 499 – 562.

[9] Buccafurri, F., Leone, N., Rullo, P. Strong and weak constraints in disjunctive Datalog. In Proceedings of the 4th International Conference on Logic Programming and Nonmonotonic Reasoning. In LNCS, Vol. 1265. 1997, pp. 2 -17.

[10] Ounnas, A., Millard, D. and Davis, H. A Metrics Framework for Evaluating Group Formation. In Prodeedings of ACM Group'07, Sanibel Island, Florida, USA, 2007. pp 221-224.