Similarity-based Grouping to Support Teachers on Collaborative Activities in Exploratory Learning Environments

S. Gutierrez-Santos[a], M. Mavrikis[b], E. Geraniou[b], A. Poulovassilis[a]
London Knowledge Lab, University of London
([a] Birkbeck College, [b] Institute of Education)
{sergut,ap}@dcs.bbk.ac.uk, {m.mavrikis,e.geraniou}@ioe.ac.uk

Abstract

This paper describes a computer-based tool that helps teachers group their students for collaborative activities in the classroom, the challenge being to organise groups of students based on their recent work so that their collaboration results in meaningful interactions. Students first work on an exploratory task individually, and then the computer suggests possible groupings of students to the teacher. The complexity of the tasks is such that teachers would require too long a time to create meaningful groups. The paper describes the design of the tool, the algorithms and metrics used for generating the groups, the evaluation of the tool, and the pedagogical context in which the tool was designed.

Keywords: collaborative learning; teacher support; similarity-based grouping.

1. Introduction

Research in collaborative learning has long recognised that working in groups does not necessarily enhance learning. It is well-established that careful planning and structuring of tasks and strategic formation of groups has the potential to promote learning [2, 19]. The advantages of encouraging students to examine different approaches to a problem, discuss the benefits and drawbacks of each, build on each other's ideas, and benefit from the reflection that results from interaction with others, have been widely identified (examples of relevant research in mathematics education, which is the domain in which work presented here is situated, include [19, 22]).

In real classrooms, however, the challenge that teachers face is to form *meaningful* groups i.e. groups that will provide opportunities for students to engage in productive discussions, enabling them reflect on their approaches to the problem, to justify and critique their solutions, and thus lead to deeper learning. Pragmatic and logistic constraints in the classroom can push teachers towards resorting to forming groups that do not take advantage of the pedagogic potential of a strategic grouping [2]. Digital technologies can be employed to enhance the potential of collaborative learning by supporting teachers in addressing this challenge.

The work presented in this paper has exactly this goal. We present a Grouping Tool, the primary beneficiaries of which are teachers in the context of a primary or secondary school who are using exploratory learning environments (ELEs) such as microworlds (e.g. for algebra or geometry) or simulators. In such a context, teachers who would like their students to undertake collaborative activities in the classroom face the following problem. On the one hand, they need to ensure that students reach their full potential and are challenged enough. One way of doing that during a lesson is to pair students who have finished interacting individually with the ELE with other students to discuss their solutions. This transition from individual work with the ELE to collaborative work and discussion with a peer needs to take as short a time as possible. On the other hand, work in ELEs results usually in a large set of approaches from the students and therefore creating strategically meaningful groups ends up being a complex task for the teacher. Students' constructions in an ELE typically involve several objects with many different attributes and/or a long trace of manipulations of objects; neither is evident to the teacher at a glance and both require deep inspection. Understanding what every student has done requires a non-trivial amount of time even for experienced teachers and therefore a tool to support this process is highly beneficial.

The use case of the Grouping Tool presented here is to support teachers with activities that utilise a mathematical microworld called eXpresser designed to help 11–14-year-old students develop algebraic ways of thinking [13]. The eXpresser microworld is one component of a larger system called MiGen which comprises also an Activity & Task Design tool for the teacher, a suite of Teacher Assistance Tools that aid the teacher in monitoring students' activities, and the MiGen Server which is responsible for the managing the interaction of the student- and teacher-facing tools with the MiGen database where all information produced and required by these tools is stored. The MiGen project team comprised computing, AI and pedagogical expertise, teacher educators, and teachers and students from several schools. The design and evaluation of the MiGen system involved several iterative cycles, the early stages of which involved the research team, teacher educators, teachers and students in one-to-one and small-scale activities, and the later stages of which involved whole-classroom trials in four secondary schools. A detailed discussion of this process is beyond the scope of this paper and we refer readers to [16] for a comprehensive account.

As motivated further in Section 3, the typical use of the eXpresser in the classroom consists of one or two tasks undertaken individually by students, followed by a collaborative activity in which students are paired to compare and comment on their different approaches, and whether they were equivalent or not [7]. There is an important problem with this approach however, namely the formation of meaningful groups. Comparing the different approaches from all the students can too time-consuming for the teacher: it would require the teacher to go through all the students' constructions in the microworld, note down their important characteristics, find complementary approaches between students and decide how to group the students. In the context of a demanding teaching schedule, it is difficult for teachers to allocate the necessary time for this, particularly during a classroom session. In pilot studies of the MiGen system in the classroom, empirical evidence suggests that for a class of around 30 students such an activity requires at least one hour to be performed accurately — and this is assuming that the students' constructions and their important characteristics are already printed on paper to allow for easy inspection by the teacher. It also requires the teacher to keep in mind several student constructions or having a very structured way of approaching the problem (e.g., in our pilot studies the important characteristics of the students' constructions had been already established and discussed with the teacher, the constructions were printed out and were spread on a table to allow shuffling them around until it seemed that meaningful groups were identified). In similar settings where grouping is required based on students' constructions' in an ELE (e.g., in dynamic geometry environments) the tendency is resort to random (e.g. location-based) or free-will groups.

This paper presents a tool that automatically suggests groupings of students to teachers that have a high probability of resulting in meaningful discussion during the collaborative task. Specifically, the tool suggests groups of students who have followed approaches that are different to each other (i.e. that have a low similarity): working together, students will the have the opportunity to explain to each other why they chose their approach, how the approaches are different, whether they are equivalent, and whether they are correct. In order to be trusted and used by the teachers, the tool has been designed to be as reliable as human experts and very fast, providing suggestions after a few seconds.

The rest of this paper is organised as follows. Section 2 reviews related work in group formation for collaborative learning and identifies the novelty of our approach. Section 3 introduces the problem and provides the context in which this research was performed. The next two sections present our technique for comparing different students' constructions (Section 4) and our heuristic for grouping students (Section 5). The fine-tuning process that resulted in the final version of our Grouping Tool is described in Section 6. Section 7 presents the user interface of the tool as it is used by teachers. The performance of the tool is evaluated in Section 8. Section 9 discusses the possible application of our techniques to different domains. The paper concludes with Section 10, giving our final remarks and discussing possible routes of future improvements and further work.

2. Related Work

A significant effort in the research on Computer-supported Collaborative Learning has traditionally focused on setting the right context, analysing the interactions of students, and on the creation of tools to moderate the collaborative activity such as scripts [18, 5]. The development of tools to generate meaningful groups for such activities has received comparatively less attention from the community, even though several authors have highlighted its importance [4].

One work that explores the importance of generating appropriate groups and looks at a way of forming the right groups is [25]. However, in contrast to our approach, this work is based on information about the collaboration context (e.g. size of group, type of collaboration) and not on the actions or the strategies of the students. The works of [10] and [17] are similar, but is based on an ontological description of the learning goals, the collaborative situation, or the constraints of participants. Our work groups students based on their own interactions with an exploratory learning environment in the context of an open-ended task. Modelling open and unstructured interaction is an important challenge, and enabling meaningful collaborative activities is paramount to benefit from the introduction of exploratory learning environments in the classroom.

Other works propose different techniques for maximising the heterogeneity of the groups, taking as given a description of the students (e.g. by a level of performance) and viewing the problem of grouping as an optimisation problem where heuristics such as ant-colony optimisation [8], particle-swarm optimisation [20], fuzzy clustering [3], or evolutionary algorithms [26] have showed good results in other domains. In contrast, the main contribution of our work is the description of a new technique to compare the interaction of different students in an exploratory learning environment. To the best of our knowledge, this is the first attempt at modelling exploratory learning interaction in order to compare students' action; this is an open question for exploratory environments that cannot be taken for granted. Once the actions of students are compared, we have used a simple heuristic to group them. Although the results seem to be good enough, other heuristics can be explored in the future once we have defined a technique to compare students' interactions with the environment.

Interesting and complementary research to ours is presented in [12] analysing why collaboration does not always work in collaborative learning settings, paying particular attention to social and psychological aspects. Their conclusions are relevant to our work and their guidelines for creating good groups (i.e. groups that will collaborate and learn together) are useful for any teacher who uses our tool and who has a tacit understanding of why some groupings are better than others. Another work that can be considered as complementary to ours is [1], which studies how different learning styles have an effect on groups of students learning together.

Outside of the classroom, some researchers use technology to support collaborative learning activities in the open, using wireless technology and context information [27, 14] to enable quick changes to the groups. Our work can be seen as complementary to such initiatives, by helping to generate more meaningful groups based on the similarity of different students' approaches to the learning activity.

3. The problem

The MiGen project focuses on activities in which the goal is to identify relationships that underpin figural patterns (figures in which one of more parts is repeated, such as the ones illustrated in Figures 1 and 3). Similar activities are often found in the UK National curriculum, and have the potential to emphasise the structural aspect of patterning rather than the purely numerical. As explained in more detail in [15] this is a key difficulty that students face.

In general, MiGen tasks consist of a generalisation question, an associated construction, and a goal expression. The generalisation question relates the construction and the expression, and involves generalisation thinking. Students are expected to undertake an activity by constructing a figural pattern in the eXpresser microworld and to identify an algebraic expression that underlies it. It is important to note that the target group for the activities with the eXpresser microworld are primary and early secondary-school students who have not yet developed advanced algebraic thinking abilities; therefore, the microworld is designed to scaffold the development of algebraic thinking by the process of constructing the figural pattern [13]. In other words, the process of construction helps students to see the inherent structure of the task pattern, and scaffolds their advancement towards the final expression. An example of such a task is shown in Figure 1. Section 3.1 presents eXpresser in more detail and in Section 3.2 we outline the collaborative tasks that can be undertaken.

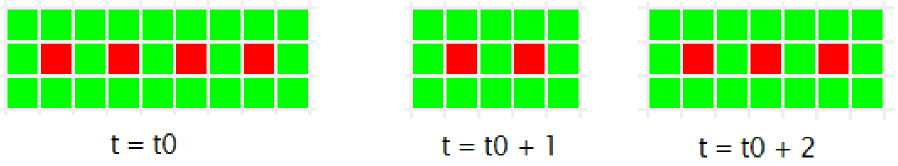

Figure 1: Example of task in MiGen: dynamic pattern. The pattern is shown in animation and different instances are shown as time moves forward. The figure shows three instances of the pattern at three consecutive seconds. One task question might be “If you know how many red tiles there are in the pattern, what is the rule for finding the number of green tiles?”

3.1. The eXpresser microworld

The eXpresser microworld allows students to construct figural patterns and corresponding expressions appropriate to each task. This microworld grants a lot of freedom to students, who may construct their patterns in a multitude of different ways. MiGen generalisation tasks are designed by the pedagogical team so as to exploit the functionalities of the eXpresser and to allow different students to express their different approaches.

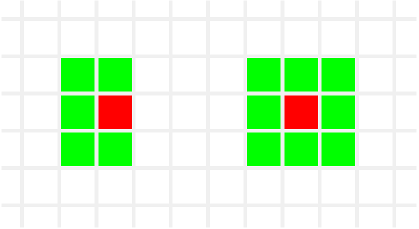

Figure 2: Two possible building blocks for the pattern shown in Figure 1: a c-shape and an o-shape.

Square tiles. The construction canvas consists of a grid of square tile-holders where different coloured tiles can be placed, moved around and combined into building blocks in different ways. Tiles can overlap and can also be subtracted (e.g. to correct an unintended overlap). This allows for a wide variety of generalisation tasks to be designed.

Numbers and expressions. Numbers can be created at any point. Expressions are algebraic combinations of numbers using addition, subtraction or multiplication; a single number is also an expression. Expressions can be dragged and dropped onto any expression placeholder, e.g. for defining the attributes or the colouring of a pattern (see below).

Patterns. Patterns are created by the spatial repetition of a *building block* which can be formed either by single tiles or another pattern. This recursive definition of patterns affords the creation of patterns of great complexity in different ways.

The repetition of a building block has three attributes: (i) number of repetitions, (ii) horizontal displacement after each repetition, and (iii) vertical displacement after each repetition. Positive horizontal displacement is left to right, and positive vertical displacement is top to bottom. For example, the pattern in Figure 1 can be created by repeating the c-shaped building block in Figure 2 four, two, and three times, moving it two spaces to the right and zero spaces vertically and adding a vertical bar at the end, or doing a similar operation with the o-shaped building block in Figure 2 and removing the overlaps by subtracting some tiles. Figure 3 shows how different patterns can be combined to form equivalent constructions.

Colouring attributes. Patterns are not coloured automatically, the student must provide the appropriate allocation of colour for them. Technically, this is also a pattern attribute (one of each colour in the pattern). This colouring has to be defined by an expression: if the expression corresponds to the correct number of tiles in the pattern, the pattern is painted in the appropriate colour (e.g. a pattern of a building block of 7 tiles as in the first pattern in Fig 3 requires $7 * x$ tiles when it is repeated x times). Colouring, therefore, provides the necessary scaffolding to connect constructions and expressions in the eXpresser as the underlying rule of a pattern can be found by adding it constituents patterns (e.g. see more examples in Fig 3).

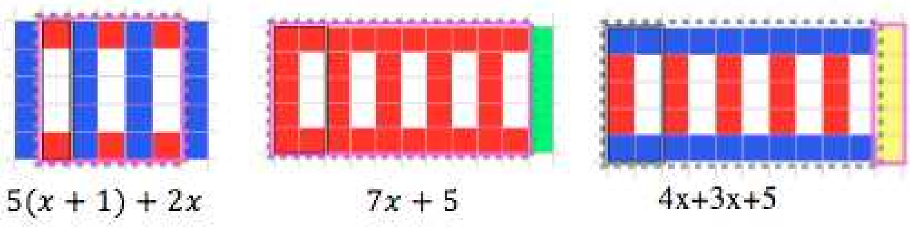

Figure 3: Example of different task solutions in MiGen. Different constructions of the pattern lead to different (but equivalent) expressions.

Variable numbers and variables. A crucial affordance of the system allows the student to specify that a numerical entity acts like a variable, making it possible (either for the student or the eXpresser) to vary its value, providing in this way a rationale for generality. Although in the project team (and in this paper) we use the term "variable" to refer to these entities, the metaphor for students — who have not been introduced to variables yet — is that of an "unlocked" number that can change. This is designed to act as a bridge between specific numbers and algebraic variables. The possibility of change is relevant because of the important implications of the eXpresser being able to vary an unlocked number: only constructions that have been made in a general way will always remain coloured.

From the point of view of this paper, the main idea to keep in mind is the flexibility that eXpresser provides for creating different patterns that are all equivalent to a figure provided to students at the outset of the task. Different pattern structures lead to different expressions, but ultimately they are all equivalent (if they are correct). Understanding this is an important step in the development of algebraic ways of thinking to express generality in maths and science, and this is where collaborative activities play an important role.

3.2. Collaborative activities with eXpresser

In addition to the generalisation tasks, the MiGen pedagogical team designed activity sequences comprising first one or two individual generalisation tasks to be undertaken using the eXpresser, and concluding in a reflective and a collaborative phase, with the aim of improving students' justification and generalisation skills [7].

The motivation behind the design of these activities is provided from a long research tradition in collaborative learning with computers in mathematics education. Early work from [21] and [9] highlighted the importance of expressing ideas in words and the opportunities that are provided by the computer artifacts to become a common focus of attention for students that can support discussions and argumentation around epistemic (rather than just pragmatic) aspects of the common artifact (e.g. [11]). Engaging with, or talking about, the product of their work and the opportunities for building on each others' ideas, students benefit from the reflection that occurs from their interaction with others (e.g. [22, 19]. Accordingly, the benefit of using eXpresser for collaborative activities is that instead of listening to the teacher saying that there are several ways of expressing the same general mathematical truth, students can see by themselves that there are other approaches to achieve the same result (see Figure 3). Discussion with peers and analysis of the characteristics of different approaches make students reflect on their own views, understand how they are equivalent to the views of others (and how they differ), and ultimately gain a deeper understanding of the domain.

However, as mentioned in the Introduction, these activities will only be fruitful if the collaboration has a purpose and for this to happen the groups need to be carefully formed so that students are grouped with other students who have different views to their own. Given the complexity of the constructions possible in eXpresser, this would be a cumbersome and lengthy process if done by hand. The tool described in the following sections makes the process fast and automatic, allowing the teacher to concentrate on other important tasks in conducting the lesson and managing the classroom.

4. Comparing students' approaches

In the first stages of design of the Grouping Tool we tried to clarify the limits of the task, namely what are the characteristics of the the best group and the worst group in the context of working with eXpresser. Although it is obviously hard to reach an agreement about these, all teachers and educators agreed that grouping together two students who have created exactly the same construction (i.e. used the same approach for the task) would not lead to much discussion as there is nothing to compare. Therefore, the first step was the determination of the definition of equality of two constructions. In collaboration with the pedagogical team, we agreed on the following definition:

> Two constructions are equal from the point of view of collaborative discussion if they have the same number of patterns, the patterns have the same building blocks, the building blocks are displaced by the same amount on each iteration, and any expressions used in their attributes are related using variables in the same way. It is not relevant if the patterns use the same colours, or even if they are coloured at all. The number of times a building block is repeated is not relevant either, i.e. two "footpath" shapes as seen earlier are considered equal (assuming all the former criteria are met) even if one has three steps and the other has five steps.

This definition allows us to know when two students should not be put together in the same group. More importantly, it also provides an indication of the factors that determine when two constructions are different: different building blocks, different attributes or variables that relate attributes in different ways (e.g. two patterns one of which is repeating a building block only to the left whilst the other pattern is repeating it diagonally). This allowed us to differentiate three types of characteristics and therefore three different metrics to compare patterns: equality, numerical difference, and set difference. These three kinds of metrics are explained in the next three subsections, respectively:

4.1. Building Blocks

The building blocks comprising a pattern are compared for equality tile-by-tile with one proviso: two building blocks that are a mirror image of each other are also considered equal. This is because the patterns created using them lead to equivalent algebraic expressions; in other words, they are functionally equivalent and there is very little to discuss in a potential collaborative activity.

The *building block distance* between two patters is defined as 0 if the building blocks comprising the patterns are equal and 1 if they are not.

4.2. Numerical attributes

Every pattern in eXpresser has several numerical (integer) attributes, as explained in Section 3, e.g. number of iterations, displacement to the right, displacement downwards. For every pattern, we build a vector with all the attribute values and we define the numerical distance between two patterns p and q as

$$nd_{pq} = \sum_i w_i (a_i^p - a_i^q)^2$$

where a_i^p is the ith attribute of pattern p, a_i^q is the ith attribute of pattern q, and w_i is a weight in the range [0,1] such that the sum of the weights is 1. The values of w_i are fine-tuned (see Section 8) to give more weight to the most important attributes, or even zero for those numerical attributes that should be ignored.

In the case of eXpresser, we know from our definition of equality discussed earlier that both the number of iterations and the number of coloured tiles for each colour are not relevant to determine differences between patterns, so the weight of these attributes is set to 0. Therefore, only two numerical are left with a non-zero weight: displacement to the right and displacement downwards.

4.3. Relations between attributes

Variables in eXpresser allow students to relate attributes to one another, e.g. the number of tiles that the building block moves to the right on each iteration can be twice the displacement going downwards. These relations between attributes of the same or different patterns is an important factor in determining differences between patterns. For example, two “footpath” patterns may look the same but one of them may relate the amount of colour needed with the number of iterations (updating both at the same time so that the pattern always looks coloured correctly) while another may use two different variables (so the pattern only looks coloured if there is a lucky coincidence in the values of both attributes).

For each attribute of a pattern that contains at least one variable we define its *attribute relationship set* as a set containing a pair $(pattern, attribute)$ for every pattern and attribute that use the same variable(s).

For a given pattern, its *pattern relationship set* is a set that contains a set of pairs $(a, (p_i, a_i))$ where a is each attribute of the pattern and (p_i, a_i) denotes each of the pairs in its attribute relationship set.

The relationship distance between two patterns is defined as the Jaccard distance between their pattern relationship sets:

$$rd_{pq} = 1 - \frac{|S_p \cap S_q|}{|S_p \cup S_q|}$$

where S_i is the pattern relationship set for pattern i. The distance is 0 for patterns with the same relationships and 1 for patterns with no relationships in common. (If the patterns do not have any relationship, i.e. $S_p = S_q = \emptyset$, their relationship distance is defined as 0).

4.4. Putting it all together

Constructions made by students are composed of several patterns. For each pair of students' constructions, these patterns are compared pair-wise, using a greedy algorithm to put together pairs of patterns whose combined building block distance and numerical distance is minimal.

Once the patterns in the two constructions have been paired, the distance of each type between the two constructions is the sum of the distances of all pairs of patterns, and their overall similarity is defined as

$$s = K \times \left(w_{bb} \cdot \frac{1}{1 + bbd} + w_n \cdot \frac{1}{1 + nd} + w_r \cdot \frac{1}{1 + rd} \right)$$

where *bbd*, *nd*, and *rd* are the total building block, numerical, and relationship distances between patterns p and q, and the w_x are weights. These weights were initially set to 0.4, 0.3, and 0.3, following discussion with members of the pedagogical team, but were later modified and fine-tuned (see Section 8) to ensure that the calculations made by the tool were in line with the perceptions of teachers about similarity and dissimilarity between different students' constructions. K is a scale factor to take into account the possibility that constructions do not have the same number of patterns and is the ratio between the minimum and the maximum number of patterns in the compared constructions; one construction could have more patterns than another for a structural reason (e.g. a different construction approach) or a casual reason (spurious shapes that have not been removed from the canvas).

5. Pair Selection

Once all pairs of students' constructions have been analysed and their similarity calculated, there are $(n^2 - n)/2$ possible pairs. A selection of pairs must be made to present as a suggestion to the teacher; this suggestion should ideally minimise the overall similarity for the whole class. However it is impossible to make an exhaustive search of all available pairings. The number of possible pairings is

$$\prod_{i=1}^{N/2} (2i - 1)$$

where N is the number of students. For a typical class of 30 students there are more than $6 \cdot 10^{15}$ possible pairings.

We have developed therefore a simple heuristic that finds pairings with a low overall similarity. The heuristic is computationally cheap and provides results that are sufficient for our purposes. In our setting, a quick response to the teacher is more important than a more accurate selection as long as the selection is good enough (i.e. all or most of the suggested pairings are of low similarity).

For the sake of completeness, we describe the algorithm below. Figure 4 depicts it in pseudo-code.

We start from the set of all possible pairs of students, with the already computed measure of their similarity.

On every iteration, the n pairs with the lowest similarity are chosen for inspection (n being half the total number or 10, whichever is lower).

For each of these pairs, we calculate the average similarity between all possible pairs among the remaining students as if the students of the pair had been removed, i.e. not taken into account.

The pair that results in the lowest average similarity for the remaining students is chosen, i.e. those students are paired. Please note that this is not necessarily the pair with the lowest similarity because the similarities of their members to all other remaining students need to be taken into account. The process is repeated after all students have been removed, i.e. assigned to a group. The resulting selection of groups is then ready to be presented to the teacher on the tools' interface.

6. Fine tuning

It is crucial that the pairs suggested by the Grouping Tool make sense to teachers. Otherwise, teachers will not trust the results of the tool and spend a long time changing the groups or even abandoning its use. Although a study of the interaction teachers–tool in the classroom its effects (e.g. modifications of classroom preparation to increase the number of collaborative activities) is beyond the scope of this paper, we report here the process by which we ensured that the tool's recommendations could be trusted by teachers.

```
# First, all similarities are calculated
for every student st1 = 1..N
  for every student st2 != st1
    calculate similarity sim1_2
  end_for
end_for
# At this point we have N students to pair, N*(N-1)/2 possible pairs
n = min(10, N/2)
repeat
  chose n_min i.e. the n pairs with the lowest similarity
  for every pair in n_min
    remove both students in pair from set of unpaired students
    calculate average similarity in remaining unpaired students
    if the average similarity is lower than former minimum
      choose those students as a new suggested pair
    end_if
    reintegrate both students to the set of unpaired students
  end_for
  add chosen students to list of pairs of students
  remove chosen students from set of unpaired students
until all N students have been paired
```

Figure 4: Pseudo-code for student pairing

We evaluated the validity of the suggestions of the tool by a process of gold-standard validation. This consisted of an iterative process in which our team of pedagogy experts were presented with several scenarios, each of them containing some constructions as made in eXpresser: experts were expected to evaluate their similarity (Figure 5).

Although our tool gives a numerical similarity (in the range (0–100]), it cannot be expected from human experts to quantify their intuitive notion of similarity in the same way. As humans are better at providing relative rather than absolute measurements, each scenario presents three “candidate” constructions to be compared against one “main” construction, and then asks for the “most” and the “least” similar. The experts’ answers were subsequently combined to obtain the gold standard; in our case, this is the candidate construction with the most votes. The outputs of the Grouping Tool were then compared against this gold standard.

The results of this evaluation of the tool are summarised in Figure 6. The orange (i.e. leftmost) bar shows the number of answers in agreement with the gold standard, averaged over all experts. The middle bar shows the level of agreement of the first version of the tool, and the rightmost bar shows the level of agreement of the final version of the tool, following several iterations of fine-tuning.

The level of agreement with the gold standard increased in later versions of the tool, up to a point where the final version was equivalent to that of the

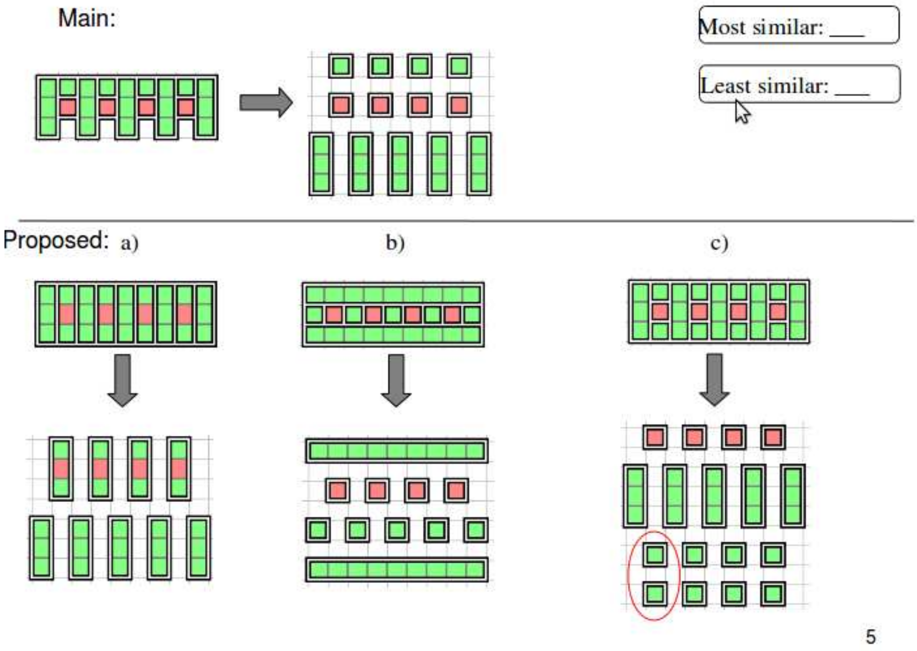

Figure 5: Gold standard evaluation. Pedagogical experts are asked which of the three proposed figures a), b), c) are the most and the least similar to the "main" one. Results are compared with the result of applying the metrics used in the Grouping Tool to answer the same question.

experts on average. At this point, the tool can be deployed in classrooms with confidence that its suggestions will be perceived as being as appropriate as those provided by a human expert.

7. User Interface

7.1. Basic interface

The interface of the Grouping Tool is illustrated in Figure 7. The tool can be used on a desktop computer or on a tablet PC, along with the other Teacher Assistance tools of MiGen. Overall, the MiGen system has a Client-Server architecture whereby the eXpresser runs on each student's computer, the teacher-facing tools run on the teacher's computer, and the MiGen server software runs on a separate server computer.

The grouping tool's UI allows teachers to request the automatic formation of pairings. The teacher can then change this selection — if needed — according to specific additional knowledge that the teacher may have about individual students.

The tool starts with a blank screen. When the teacher selects to "Create pairs" from a drop-down menu at the top of the screen, the tool connects to the

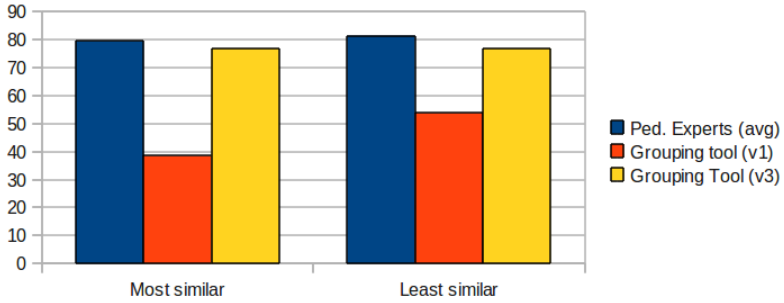

Figure 6: Gold standard evaluation of the tool's suggestions in MiGen.

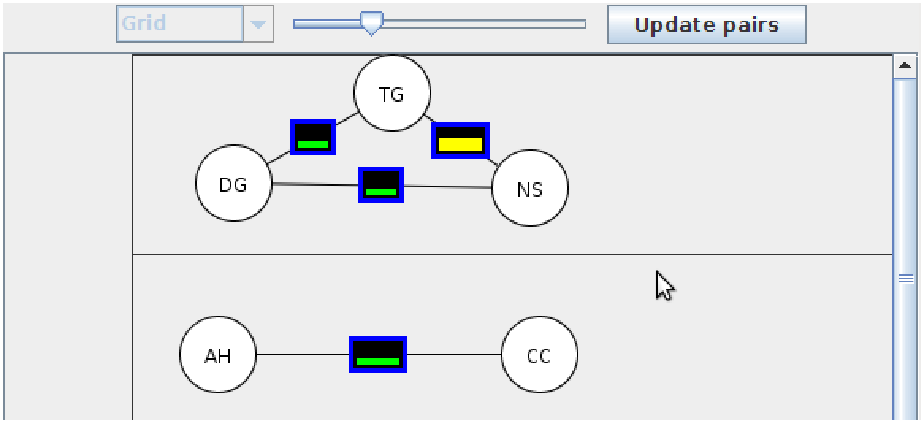

Figure 7: Partial view of the Grouping Tool UI, showing a pair and a triad. In classes with an odd number of students, the first group has three students.

MiGen server, which retrieves all the students' models from the MiGen database and returns them to the tool. Depending on the bandwidth of the network, we have observed that this *information download* phase is the one that takes the longer time, although it rarely takes more than a few seconds even on a limited Wi-Fi network composed of off-the-shelf components.

After the information is downloaded, the pairs are calculated by the tool and displayed on the teacher's screen. Students are shown as circles containing the students' initials. Pairs of students are connected by a straight line with a square that indicates the similarity between the students' constructions. Showing all possible such connections would unnecessarily clutter the interface, so only the connections between the students in the same group (typically a pair, thus only one connection) are shown. The screen is split into a set of "cells" that represent groups of students. All students inside the same cell are considered to be in the same group, and pairs of such students are connected by lines showing the calculated similarity between their constructions. In the case of classes comprising an odd number of students, the first cell will contain three students.

Once the tool's initial suggestion of student groups has been displayed, the teacher has the option of changing the groups by "dragging" students (i.e. circles) on the screen. As a circle is moved from one cell to another, the corresponding student is automatically switched from group to group and their similarity with the other members of the new group (which has been precomputed and cached in memory) is immediately shown. This allows the teacher to fulfil two roles. First, when the computer suggests a group that is good in theory (low similarity of the approaches) but not in practise (e.g. due to interpersonal issues relating to the students) the teacher can change the groups easily to accommodate this contextual information about individual students that the computer does not know. Second, the fact that all the information that the grouping tool requires is in the memory of the teacher's computer, and no access to the MiGen server is necessary, makes the experience highly interactive for the teacher who can quickly see the similarities between the constructions of a group of students. There is no limit to the number of students who can be in the same cell apart from screen size limitations. Although teachers can zoom/unzoom to make the cells as big as they desire, we have observed in our pilot studies that it is rare for teachers to create groups with more than four students (i.e. 6 connections) at any time.

7.2. Model inspection

During our pilot studies, several teachers commented that they would like to have more fine-grained information about the groups suggested by the computer. Although the similarity information can be shown both as a category (i.e. low, high) or a number (i.e. 13%, 88%), some teachers said that they would like to see the models created by students to make a better informed decision when modifying the pairs of the computer. Therefore, we added additional functionality that allows teachers to view students' models by clicking on the circle representing a student.

When one of the student circles is clicked on by the teacher, a new window opens that shows the models of all students who are in that group. These models can be inspected e.g., in order to scrutinise their attributes and building blocks, and to examine their rules. Teachers can also look at past stages of a model's construction; historical information about a model's construction may comprise a large volume of data, so it is fetched from the server in an on-request basis so to minimise the initial delay when making up the pairs. On a local network, the delay to open the models of a group of three students takes less than a second.

Once the teacher is satisfied with the current grouping, she can tell the students what their groups will be for the collaborative task that is about to be undertaken by the class. In classrooms with an electronic whiteboard, teachers may want to project the groups' composition for all students to see. Students can then work in their groups, discussing their solutions, trying to convince each other about the correctness of their solutions, and deciding upon the equivalence or not of their approaches [7].

It is important to note that examining the students' pairs in detail is a time-consuming process. Based on teachers' comments from their usage of the Group Tool in classroom activities so far, the most common scenario is that teachers will follow the tool's suggestions as long as it is reliable — as discussed in Section 6— and fast — as discussed in Section 8. Although in the future we are planning to test how often and why teachers may need to revisit the suggested pairs, providing functionality that allows teachers to modify the pairs generated by the tool is important, particularly in the early stages of introducing the tool (when users may want to feel more in control) and also to accommodate those rare situations where the teacher may want to take into account interpersonal relationships between students.

8. Performance Evaluation

One of the main motivations for creating our Grouping Tool was the high cost (mostly in terms of time required by teachers) of forming groups for collaboration in a classroom. Therefore, one of the main measurements of fitness-to-purpose of the Grouping Tool is the time required by the tool to generate groups, or —in other words— the extent to which the tool saves teachers time in their preparation of collaborative activities.

In order to do this, we prepared an experimental set-up that is based on real classroom use of the tool. We have used the tool in the same way as it is used in the classroom: retrieving the information about students' constructions from the MiGen server, analysing these constructions and comparing them, and finally suggesting groupings for the collaborative activity. In order to do this, we have loaded the server with real constructions arising from several classroom trials of the MiGen system, with students working on the same eXpresser task. This is to ensure that the computational costs are realistic and similar to what would happen in the typical classroom. The computer that runs the Grouping Tool and the computer that runs the MiGen server are in the same local network; this has always been the set-up when we have used the MiGen system in real classrooms (usually over some low-cost variant of the IEEE 802.11 Wi-Fi family).

Figure 8 shows the time needed for different numbers of students in the classroom. It can be observed that the time required to generate the groups of students is quite low, even for high numbers of students. For the typical situation in which a class has between 20 and 30 students, the total time required to fetch their models from the server, calculate their similarities, and then propose a possible set of groups to the teacher takes a few seconds. Most of the time is spent fetching the models from the server, and the time spent in calculating similarities and pairing students is always below 5ms.

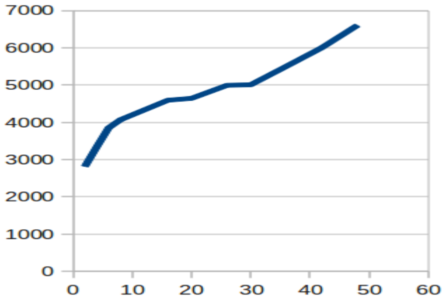

Figure 8: The x axis shows the number of students in the class. The y axis shows the time (in ms) taken to show suggested groups on the screen since the moment the teacher presses the ”Make pairs” button.

This small delay is more than adequate for use of the Grouping Tool in the classroom. Not only can teachers generate groups on the spot before the collaborative activity starts, they can do it at any point earlier in the lesson for a quick glance at what the possible groupings may be, or they can do it after the end of the lesson in preparation for a future lesson that will contain the collaborative activity, for planning purposes, etc.

It is important to remember that the tool suggests groups of students based on their interaction with the eXpresser microworld, and not based on personality traits, personal relationships between the students, or other contextual information that may influence the final grouping choice by the teacher. If the teacher needs to modify the pairs originally suggested by the tool, this is additional time to be spent before the collaborative activity starts. As has been discussed in former sections, the tool updates immediately the similarity information about students' models if the teacher changes a group manually, which means that the teacher immediately has a clear idea of whether the new group is suitable or not (i.e. there is low similarity between the students' constructions). The most common scenario is that any student can in principle be paired with any other student in the class, the exceptions will be few, and thus the overhead for the teacher low. In fact, in our studies to date with the tool, the teachers have never made any change to the original suggestion generated by the tool.

We finally note that the communication latency takes far longer than the computation of the groupings, even over a local network. This means that our algorithm is light-weight and fast. Use of the Grouping tool in a different set-up, e.g. with the MiGen server in a different network, or somewhere on the Internet, may take longer. Investigating this possibility lies beyond the scope of the present work, as our typical MiGen scenario involves the use of the system on a local network, but it is an interesting area of future investigation.

9. Discussion

The Grouping Tool presented in this paper was developed to help teachers in their preparation of collaborative learning activities in the context of learning mathematics for young learners, but the principles are general enough to be extended and used in other domains.

Exploratory learning activities are, by nature, the kind of activities that can benefit the most from collaboration, discussion, and peer review. There are open problems that can be approached in different ways to reach valid solutions in many domains. At the moment we are considering extending the ideas used for our Grouping Tool to other learning domains that are inherently exploratory, such as dynamic geometry, programming, or electronic design (in particular, using description languages like VHDL). In such domains, older students may also benefit from discussing their algorithms and data structures with their peers. For example, we have observed such behaviour happening naturally in university-level programming courses; the use of a tool like the one presented here could add some structure to this naturally emergent phenomenon and enhance its positive effect on learning. There are also additional possibilities that could be exploited in this context, such as showing students pieces of "good code" that are similar to what they have written so far, allowing them to learn by reading code written by experts that is not very different from their own code (i.e. that is in their Zone of Proximal Development in the Vygotskian sense [24]).

Initiatives like the use of peer-review among students can show students solution approaches other than their own, making them reflect on the similarities and differences with their own approach, and the fitness of each approach. One of the open problems in the peer-review field is finding the right match between reviewer and reviewee [23]. Our techniques could be applied in this scenario to calculate similarities between different approaches.

More generally, any task or domain in which an approach or strategy can be represented by a vector can be analysed with the tools presented in this paper. Our problem required the use of three different types of vectors and three different metrics (equality, numerical difference, and similarity of sets), but different problems may require a subset of the metrics we needed or maybe a new type of metric. For example,the description of simple programs might use the number of statements in a loop or the number of methods in a class as numerical attributes, while the use of private fields in different methods of a class may be a relevant relation to be taken into account, and there may be no need to test any attribute for equality. Finding the right representation (i.e. based on what teachers think is really important from a pedagogical point of view) is the most intellectually demanding aspect of the procedure. Obtaining the right set of traits that uniquely define an approach is highly challenging because teachers' perceptions are deeply ingrained in their own implicit knowledge, and finding the right set of characteristics usually requires several iterations of prototyping with teachers.

However, obtaining the right traits is only part of the solution. Finding the right weights to balance the different metrics is also crucially important if the teachers are expected to trust and use the tool in the classroom — the other crucial factor is speed. An iterative process of gold-standard validation is very important because it harmonises the results of the algorithm with what teachers expect. In our case, building blocks initially were assigned an average weight based on initial interviews with teachers, but it became gradually clear that teachers were giving more weight to the building block than to all other considerations put together, probably due to the epistemic relationship between the shape of the building block and the algebraic rule used to describe the pattern (see Figure 3). We have allowed the possibility of configuring the weights (using configuration files in advance of running the Grouping Tool) but have not so far provided a corresponding functionality in the user interface. Our top priority was to create a tool that could be used by teachers directly. Allowing teachers the possibility of modifying the weights themselves would allow teachers to see the effects of such modifications on the grouping produced by the tool on-the-fly. Although it would be technically straight-forward to provide such a facility, it is not clear how useful teachers would find it in practise and this is an area of future research.

Finally, we would like to stress that the application depicted here aims at minimising the similarity between members of the groups, because the goal in the MiGen project is to give students more possibilities for discussion. In our experience, this is the most common scenario in which our tool would help a teacher in the classroom. That is only one possibility, though. Other different applications aim at maximising the similarity of the groups instead. We have already suggested a the hypothetical scenario of "looking at good code" in a programming lesson. Another, real, scenario is taken from the recent EU-funded project Metafora project where eXpresser is integrated in synchronous and asynchronous computer-supported collaborative activities [6]. Apart from grouping students for discussions based on the constructions after the completion of the activity, the project team is experimenting with the benefits of providing suggestions to students about who of their peers is best equipped to help them. In this case, it is more useful to group together students who have taken *similar* approaches rather than different ones.

10. Conclusions and Future Work

We have presented a tool that groups students according to their different approaches to an exploratory learning task. The work has been undertaken in the context of a microworld for developing algebraic ways of thinking but could be extended to other domains. The tool allows teachers to easily create groups for collaborative activities that maximise the probability of meaningful discussions by putting together students whose approaches are very different (i.e. have low similarity). The tool can generate groups for all students in the classroom in a few seconds. In contrast, an experienced teacher, with the help of a teaching assistant, needs to spend at least one hour for a class of 30 students to do this manually.

Our tool enables teachers to group students into meaningful groups very quickly, thus facilitating the integration of collaborative activities into the classroom routine. Several teachers have been involved in the iterative design of our Grouping Tool to ensure that the tool's suggestions are in line with the teachers' own understanding of what constitutes a good pairing of students for discussion given the students' interactions with the exploratory learning environment. The tool is thus as reliable as human pedagogical experts and much faster, the two main requirements for a tool to be trusted and used by teachers in the classroom.

There are several directions for improvement and future research. The use of better algorithms for weight-tuning and pair-selection is one possibility (some alternatives were discussed in Section 2) but the current heuristic is good enough and there would be little to gain in terms of usability. A potentially more fruitful direction of research would be to let the Grouping Tool learn from the teacher's actions, e.g. which students should not be put together. Teachers' actions can be analysed to introduce constraints in the pairing algorithm so that future suggestions take that information into account, facilitating the teacher's role even further.

Another line of research that lies beyond the scope of this paper is to explore more deeply the interaction between the teacher and the tool from a human–computer interaction point of view (e.g. do teachers trust the suggestions from the tool? how often do they modify the grouping?), and how the introduction of a tool such as this influences teachers' choices of activities for their classroom (e.g. do they introduce more discussion activities in their classroom routine?).

In the future, our approach to describing students' exploratory interactions (and subsequent pairing) could be used to group students working on exploratory tasks in different domains (like the aforementioned example of programming) over a *grid*, perhaps in the context of a massive online course. This paper can be seen as a first step towards that goal.

References

[1] Alfonseca, E., Carro, R., Martin, E., Ortigosa, A., Paredes, P., 2006. The impact of learning styles on student grouping for collaborative learning: a case study. User Modeling and UserAdapted Interaction 16 (3–4), 377–401.

[2] Blatchford, P., Kutnick, P., Baines, E., Galton, M., 2003. Toward a social pedagogy of classroom group work. International Journal of Educational Research 39 (1-2), 153–172.

[3] Christodoulopoulos, C., 2007. A group formation tool in an e-learning context. In: 19th IEEE Int. Conf. on Tools with Artificial Intelligence. pp. 111–123.

[4] Dillenbourg, P., 2002. Over-scripting CSCL: The risks of blending collaborative learning with instructional design. In: P.A.Kirschner (Ed.), Three worlds of CSCL. Can we support CSCL? pp. 61–91.

[5] Dillenbourg, P., Fischer, F., 2007. Basics of computer-supported collaborative learning. In: Zeitschrift fur Berufs- und Wirtschaftspadagogik. Vol. 21. pp. 111–130.

[6] Dragon, T., McLaren, B. M., Mavrikis, M., Geraniou, E., 2012. Scaffolding Collaborative Learning Opportunities: Integrating Microworld Use and Argumentation. In: Ardissono (Ed.), Advances in User Modeling: UMAP 2011 Workshops (pp. 18-30), Girona, Spain, July 11-15, 2011, Revised Selected Papers. Vol. 7138 of Lecture Notes in Computer Science. Girona, Spain, pp. 18–30.

[7] Geraniou, G., Mavrikis, M., Hoyles, C., Noss, R., 2011. Students' justification strategies on equivalence of quasi-algebraic expressions. In: Ubuz, B. (Ed.), Proceedings of the 35th. Conference of the International Group for the Psychology of Mathematics Education. Vol. 2. pp. 393–400.

[8] Graf, S., Bekele, R., 2006. Forming heterogeneous groups for intelligent collaborative learning systems with ant colony optimization. In: Intelligent Tutoring Systems. pp. 217–226.

[9] Healy, L., Pozzi, S., Hoyles, C., 1995. Making sense of groups, computers and mathematics. Cognition and Instruction 13 (4), 505–523.

[10] Inaba, A., Ikeda, M., Mizoguchi, R., Toyoda, J., 2000. Theoretical justification of opportunistic group formation with ontological engineering. In: Intelligent Tutoring Systems. pp. 282–291.

[11] Kieran, C., 2001. The mathematical discourse of 13-year-old partnered problem solving and its relation to the mathematics that emerges. Educational Studies in Mathematics 46, 187–228.

[12] Kreijns, K., Kirschner, P. A., Jochems, W., 2003. Identifying the pitfalls for social interaction in computer-supported collaborative learning environments: a review of the research. Computers in Human Behavior 19 (3), 335–353.

[13] Mavrikis, M., Noss, R., Hoyles, C., Geraniou, E., (2012). Sowing the seeds of algebraic generalisation: designing epistemic affordances for an intelligent microworld. Special Issue on Knowledge Transformation, Design and Technology, Journal of Computer Assisted Learning.

[14] Muehlenbrock, M., 2005. Formation of learning groups by using learner profiles and context information. In: Int. Conf. on Artificial Intelligence in Education.

[15] Noss, R., Hoyles, C., Mavrikis, M., Geraniou, E., Gutierrez-Santos, S., Pearce, D., Aug. 2009. Broadening the sense of 'dynamic': a microworld to support students' mathematical generalisation. The International Journal on Mathematics Education (ZDM) 41 (4), 493–503.

[16] Noss, R., Poulovassilis, A., an Sergio Gutierrez-Santos, E. G., Hoyles, C., Kahn, K., Magoulas, G. D., Mavrikis, M., August 2012. The design of a system to support exploratory learning of algebraic generalisation. Computers and Education 59 (1), 63–81.

[17] Ounnas, A., 2007. Towards semantic group formation. In: IEEE Int. Conf. on Advanced Learning Technologies. pp. 825–827.

[18] Stahl, G., Koschmann, T., Suthers, D., 2006. Cscl: An historical perspective. In: Sawyer, R. K. (Ed.), Cambridge Handbook of the Learning Sciences. Cambridge, UK.

[19] Swan, M., 2006. Collaborative Learning in Mathematics: A Challenge to Our Beliefs and Practices. National Institute for Advanced and Continuing Education (NIACE, London.

[20] Tambouris, E., Panopoulou, E., Tarabanis, K., Ryberg, T., Buus, L., Peristeras, V., July 2010. An innovative group formation approach for collaborative learning. IEEE Learning Technology newsletter 12 (3).

[21] Teasley, S. D., Roschelle, J., 1993. Constructing a joint problem space: The computer as a tool for sharing knowledge. In: Computers as Cognitive Tools. pp. 229–258.

[22] Vahey, P., Tatar, D., Roschelle, J., 2007. Using handheld technology to move between private and public interactions in the classroom. In: Ubiquitous computing in education: Invisible technology, visible impact. Lawrence Erlbaum, pp. 187–210.

[23] Verleger, M., Diefes-Dux, H., Ohland, M., Besterfield-Sacre, M., Brophy, S., 2010. Challenges to informed peer-review matching algorithms. Journal of Engineering Education 99 (4), 397–408.

[24] Vygotsky, L., 1978. Mind and society: The development of higher psychological processes. Harvard University Press.

[25] Wessner, M., Pfister, H.-R., 2001. Group formation in computer-supported collaborative learning. In: ACM SIGGROUP Conference on Supporting Group Work.

[26] Yannibelli, V., Amandi, A., 2011. Forming well-balanced collaborative learning teams according to the roles of their members: An evolutionary approach. In: IEEE 12th International Symposium on Computational Intelligence and Informatics.

[27] Zurita, G., Nussbaum, M., Salinas, R., 2005. Dynamic grouping in collaborative learning supported by wireless handhelds. Educational Technology & Society 8 (3), 149–161.