Visualisation and Analysis of Students' Interaction Data in Exploratory Learning Environments

Manolis Mavrikis
London Knowledge Lab
UCL Institute of Education
m.mavrikis@lkl.ac.uk

Zheng Zhu
London Knowledge Lab
Birkbeck, Univ. of London
zheng@dcs.bbk.ac.uk

Sergio Gutierrez-Santos
London Knowledge Lab
Birkbeck, Univ. of London
sergut@dcs.bbk.ac.uk

Alexandra Poulovassilis
London Knowledge Lab
Birkbeck, Univ. of London
ap@dcs.bbk.ac.uk

ABSTRACT

Log files from adaptive Exploratory Learning Environments can contain prohibitively large quantities of data for visualisation and analysis. Moreover, it is hard to know in advance what data is required for analytical purposes. Using a microworld for secondary algebra as a case study, we discuss how students' interaction data can be transformed into a data warehouse in order to allow its visualisation and exploration using online analytical processing (OLAP) tools. We also present some additional, more targeted, visualisations of the students' interaction data. We demonstrate the possibilities that these visualisations provide for exploratory data analysis, enabling confirmation or contradiction of expectations that pedagogical experts may have about the system and ultimately providing both empirical evidence and insights for its further development.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

Keywords

exploratory learning environments, analysis, visualisation

1. INTRODUCTION

In recent years there has been much research and development work focusing on open-ended interactive educational applications that encourage students' experimentation within a knowledge domain. These applications range from simple games to complex simulators and microworlds [3]. These types of systems are becoming increasingly popular due to new forms of interaction afforded by hand-held devices and increasing ease of creation through related authoring tools. In parallel with their increased adoption, there

has developed an appreciation that in order for students to benefit from interaction with such *Exploratory Learning Environments* (ELEs) there is a need for explicit pedagogical support to be provided to students [4]. This has led to research and development of intelligent techniques for providing adaptive support to students in order to foster their productive interaction with ELEs [3].

The design and improvement of such adaptive ELEs is not a trivial task. Following a principled, evidence-based approach needs to rely on data gathered from students' interactions with the ELE, in order to help educationalists understand how students are interacting with the system and technical experts to develop enhanced or new intelligent support features. However, typical log files from ELEs contain such large quantities of data that their interpretation for researchers, teachers and systems designers a difficult and expensive task (cf. [6]). In addition, it is not possible to always know in advance what data are relevant for analytical purposes and therefore an exploratory analysis of the data gathered may be needed. Lastly, the logging of students' interactions typically takes place in a manner that is optimal for recording and supporting students' ongoing activities within the ELE but not necessarily for subsequent analysis and decision-making.

In this paper, we describe our approach to transforming students' interaction data from an operational online database into a data warehouse, thereby enabling data visualization and exploration based on online analytical processing and other visual tools that increase the speed of data retrieval and can allow stakeholders who have little technical expertise or training to explore such data. The main objective of this approach is to support designers, researchers (and also potentially teachers) to conduct exploratory analysis of the data that includes visualising it in different ways and enabling decision-making.

Our case study is the MiGen system (www.migen.org), an intelligent ELE that fosters 11-14 year old students' learning of algebraic generalisation [5]. In MiGen, students undertake tasks in a microworld called *eXpresser* — see Figure 1. These tasks ask students to create models consisting of 2-dimensional tiled, coloured patterns constructed from one or more building blocks. Firstly specific instances of such models need to be constructed and then generalised versions in which one or more of the numbers in their construction are replaced by so-called "unlocked" numbers, i.e. by variables.
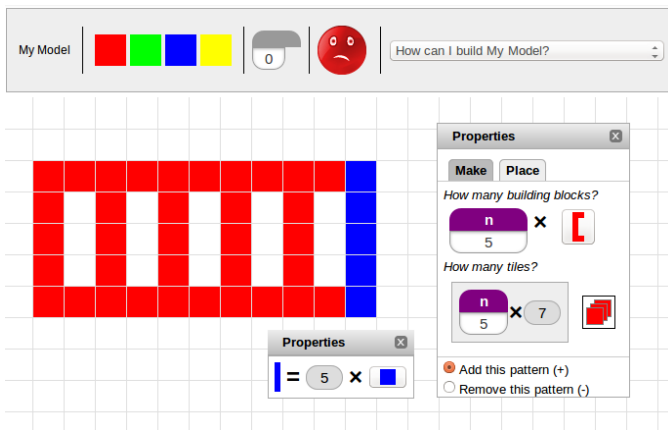
Figure 1: The eXpresser: Students create building blocks and use them to construct patterns, which they subsequently colour. They need to use variables to specify how many tiles of each colour are needed in a generalised pattern.

Figure 2: MiGen Architecture

In parallel, students are asked to formulate algebraic rules specifying the number of tiles of each colour that are needed to fully colour their models.

The MiGen system includes an intelligent component called the *eGeneraliser* which provides both unsolicited and on-demand personalised feedback to students, based on a three-layer architecture comprising Analysis, Reasoning and Feedback Generation sub-components (see [3]). This intelligent support for students was designed through a detailed process of cognitive task analysis and Wizard-of-Oz studies and it relies on a set of *task-independent* (TI) and *task-dependent* (TD) indicators being detected as students are interacting with the system. TI indicators occur when the eXpresser detects that specific actions have been undertaken by a student e.g. 'student has placed a tile on the canvas', 'student has made a building block', 'student has unlocked a number'. The detection of TI indicators is not dependent on knowledge specific to the task that the student is currently working on. TD indicators are detected by the analysis and reasoning sub-components of the eGeneraliser, based on the students' actions and on knowledge specific to the current task, e.g. 'student has made a plausible building block for this task', 'student has unlocked too many numbers for this task', 'student has achieved task goal n'. All the occurrences of TI and TD indicators are stored in the operational online MiGen database, leading to large volumes of such data.

Figure 2 (from [5]) illustrates the architecture of the MiGen system. Shown in white are the components comprising the Student software running on the students' computers, in light grey the components comprising the Teacher software running on the teacher's computer, and in dark grey the Server components. Also shown is the information flow between components. Each of the user-facing tools consists of a User Interface (UI) component and an Information Layer (IL) component. Each tool's UI is responsible for interaction with the user, while its IL is responsible for managing the data structures and computation required to support
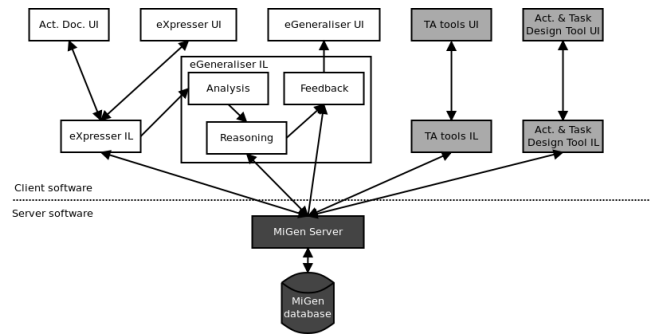
the UI and for communicating with the Server software. The Server software in turn provides access to the MiGen database. 'Act.Doc.' denotes 'Activity Document', within which construction tasks are presented to students and they record their reflections as they undertake these tasks using eXpresser. The Act. & Task Design Tool is used to create tasks and activity documents. The TA (Teacher Assistance) Tools assist teachers in monitoring students' activities, informing teachers' interventions in the class and review of students' achievements (see [2] for details of these).

In the initial implementation, all the tools were implemented in Java and integrated into a lightweight architecture based on REST, facilitating iterative prototyping and trialling. More recent development has ported the Student and Teacher tools to a Web environment (using XHTML, JavaScript and Ajax technologies) and the Server to Google's App Engine, for greater ease of adoption in schools and greater scalability. Performance studies have shown the system's ability to handle thousands of concurrent users (comprising concurrently running classes of students and teachers, at a typical ratio of about 30 students to 1 teacher), typically generating 50 to 100 indicator occurrences per student per hour.

The question we address in this paper is: how might large volumes of interaction data from ELEs such as MiGen be visualised and explored in order to determine the effectiveness of the intelligent support being provided by the system to students and to improve this support as necessary? We have investigated several possible visualisations and we here report on those that we have found to be the most useful: (i) multi-dimensional data visualisation and exploration using online analytical processing tools, in Section 2; and (ii) additional, more targeted, visualisations of the frequency of occurrence of different types of indicators and the transitions between them, in Section 3. Section 4 gives our concluding remarks and directions of future work.

2. MULTI-DIMENSIONAL VISUALISATION

Our first approach towards visualisation and analysis of the data being gathered by MiGen as students interacted with eXpresser was to investigate multi-dimensional mod-
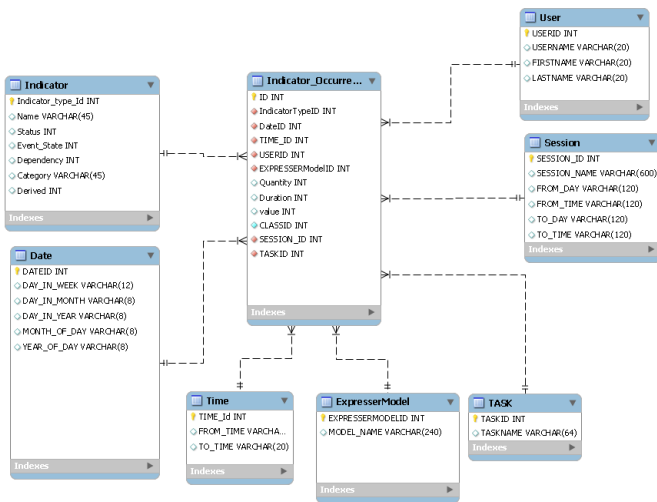
Figure 3: The Schema of the Data Warehouse

Figure 4: Indicators broken down by Status

elling and visualisation from the data management field, known as *Online Analytical Processing* (OLAP). The aim of OLAP systems is to provide interactive tools for applying filtering and aggregation operations to multi-dimensional data and viewing the results in a tabular or graph form. We used the Oracle data warehouse and Analytic Workspace Manager[1].

The first step required developing scripts to extract and transform the data from the operational MiGen database holding the students' interaction data into a *data warehouse* — this is the so-called Extraction, Transformation, Load (ETL) process and has the aim of transforming day-to-day operational data from its original format into a form that is more efficiently and easily visualised and analysed by domain experts. A data warehouse is typically designed to collect information on one or more numeric *measures*, each measure depending on a set of *dimensions*. Each dimension is described by a set of *attributes*. Data warehouse schemas thus typically consist of one or more *star schemas*. A star schema comprises a *fact table* and a number of *dimension tables*. The fact table contains information regarding one or more measures and its primary key is a concatenation of the primary keys of the dimension tables. Each dimension table contains information about the attributes of one of the dimensions relating to the measure(s) recorded in the fact table.

In the case of the MiGen interaction data, we identified two measures — the *quantity* and the *duration* of the indicator occurrences, and seven dimensions relating to each occurrence of an indicator: the Date and Time it occurred, the ExpresserModel and Task it relates to, the User who generated it, the Session in which it was generated, and its Indicator_type (e.g. 'tile placed', 'pattern made', 'number created', 'number unlocked', 'value of unlocked number changed', 'correct local rule created', 'inactive student' etc. — see the Appendix for the full list of indicator types).

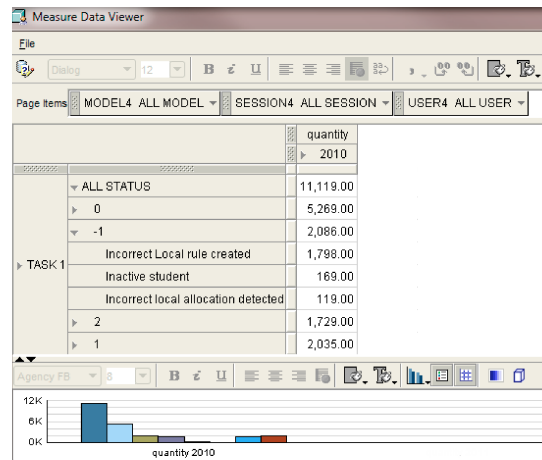Figure 3 illustrates the data warehouse schema that we designed, showing the table Indicator_Occurrence as the cen-

tral fact table and the seven connected dimension tables. Additional data relating to an indicator occurrence is stored in the 'value' attribute of the Indicator_Occurrence table. We note that the fact table is able to join with each of the dimension tables (because it contains their primary key attributes in its attribute list) but the dimension tables do not join with each other. Hence the 'star' shape of the schema.

In the Indicator table, each type of indicator has associated with it a Status. This may have value -1, 0 or 1 indicating, respectively, that an occurrence of this type of indicator is regarded as showing that the student is making negative, neutral or positive progress towards achieving the task goals. An additional Status value of 2 indicates that this indicator type relates to an intervention generated by the system: this may be an explicit request for feedback by the student, or the generation and presentation of feedback by the system.

There are several interesting visualisations provided by standard OLAP tools that, with little training, can be made accessible to non-technical domain experts. For example, Figure 4 shows a breakdown of the number of indicators occurring by Task ID, Status and Year. Task ID is an attribute of the Task dimension, Year is an attribute of the Date dimension, and Status is an attribute of the Indicator_type dimension. In Figure 4 there has also been some filtering of the data, so that data relating only to year 2010 and Task 1 has been selected to be viewed. There has also been a 'drilling down' into the data relating to Status -1, to break it down into the specific indicator types that have this Status value (namely, Incorrect Local rule created, Inactive student, Incorrect local allocation detected). This kind of view allows us to see what kinds of positive, neutral and negative behaviours are commonly occurring as students are undertaking a task, in order to gauge, for example, students' level of engagement and productive interaction with the task.

Closing up again the Status -1 row and breaking down the data into the Models to which the indicators relate, we obtain the view shown in Figure 5, which allows us to see the relative numbers of positive, neutral and negative indicators occurring during the construction of each model, in order to gauge, for example, the relative degrees of difficulty of different construction tasks.

---

[1]http://www.oracle.com/technetwork/database/enterprise-edition/index-087067.html

Figure 5: Indicators by Status and Model

The Proportion of Status of Indicator Type (Mean) in Sessions

Figure 6: Proportion of negative (-1), neutral (0), positive (1) or intervention (2) indicators

3. MORE TARGETED VISUALISATIONS

In our efforts to analyse students' interaction data and to identify which aspects of MiGen's intelligent support require further research and design work, the above multi-dimensional visualisations allowed the MiGen team to interact with each other and with other experts providing insights to support our decision-making. However, beyond these standard visualisations provided by OLAP tools, we have also developed several other ad-hoc visualisations that help us to further explore students' interactions. We present here two of the most useful ones, relating to the frequency at which different indicator types occur and to the transitions between different types of indicator.

3.1 Frequency of indicator type occurrences

For analysing frequency of occurrence of different indicator types we have found visualisations such as the one shown in Figure 6 very useful. The heights of the bars correspond to the conditional relative frequencies of indicator types with Status -1, 0, 1, 2 in each of the classroom sessions being analysed (the analysis here relates to three successive classroom sessions: Session 1, Session 2, Session 3). The widths of the bars correspond to the relative frequencies of indicator occurrences between the sessions.

Having a way to visualise this information quickly for researchers and teachers is useful. In this example, we can see that the number of indicator occurrences grows with each successive session and that the frequency of occurrence of negative indicator types is decreasing with each successive session. This may be because students are becoming more familiar with using the system — a hypothesis that would warrant further investigation.

3.2 Transition of indicator type occurrences

The occurrence of sequences of indicator types can present patterns that may also provide insights. Standard sequence analysis, however, provides patterns that are difficult to inspect. In order to facilitate the involvement of pedagogical experts, we therefore investigated *transition matrices*, which are used to describe the transitions of a Markov chain:

Given a finite space of indicator types, $P_{ij} = P(j|i)$ is the probability of moving from indicator type i to indicator type j in one time step. Transition matrices can be normalised to quantify the transition probability from indicator i to any other indicator. We can also normalise the matrix to mea-

sure the incoming transition probability to indicator j from other indicators. In addition, we add artificial points to the system to capture the start and end of the student's interactions relating to each model. Accordingly, for each model, s indicates the first indicator before the student begins construction of the model and e the last indicator at the end of the model's construction.
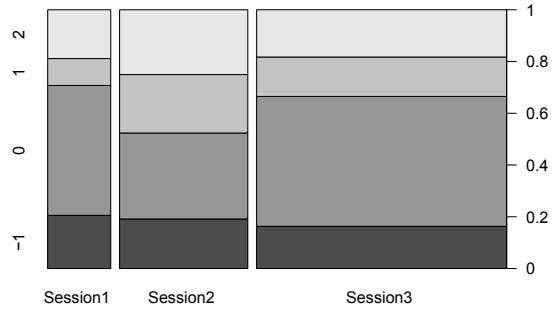
Transition matrices can be visualised using graphs such as those shown in Figures 7 to 10. Indicators shown with a circle round them indicate that there are transitions in the data where this indicator occurs repeatedly in succession. The thickness of each line or circle indicates the value of the transition probability: the thicker the line, the higher the probability. The red (light grey) lines are associated with a probability less than 0.2 and the black lines with a probability greater than or equal to 0.2.
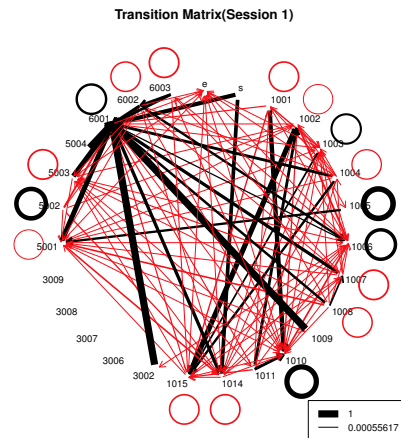

Figure 7: Incoming Transition Matrix (Session 1)

Figures 7 and 8 showing the transition matrices from Sessions 1 and 2 lead to interesting insights through visual observation. As an example, we focus on indicator 6001 — corresponding to an intervention generated by the system. There is a stark difference between the two Sessions in the transitions incoming to indicator 6001 from indicator 3002 — corresponding to a numerical answer being provided by the student. There is a black arrow 3002 → 6001 in Figure 7
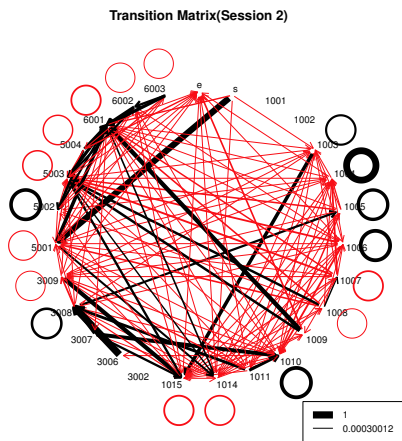
Figure 8: Incoming Transition Matrix (Session 2)

but no such transition in Figure 8, which indicates that feedback received from the system in Session 1 was carried over to students' interactions in Session 2. Such an observation helps us raise a hypothesis for more detailed analysis, or further student observations and data gathering: "are students internalising the system's feedback and thus avoiding the same error in subsequent sessions, or is this simply an artefact of their increasing familiarity with the system?"

Equally interesting is the comparison of outgoing transitions. In Figure 9 relating to Session 1 we see a predominance of transitions from a system-generated intervention (6001) to the student's naming of a number (1009) and to other indicators relating to numbers (1006–1009). This behaviour is repeated in Session 2 (not shown here). In Figure 10 relating to Session 3, the likelihood of such transitions is less, indicating that such system-generated interventions are needed fewer times. This is confirmed by the increased likelihood of a direct transition between a number being unlocked (1007) and being named (1009) without requiring an intervention by the system (this was confirmed by further analysis and empirical observations in the classroom).

However, in Figure 10 there is a higher probability of indicator 1004 (corresponding to an unlocked number being changed) transitioning to itself. This is an example of an unexpected behaviour, and may indicate trial-and-error by students starting to take advantage of the affordances of the eXpresser (i.e. 'gaming' the system [1]) after having been prompted in the first Session. This behaviour is something beyond the intentions of the designers and demonstrates how our analysis and visualisation approach allows us to make informed decisions and identify areas of further development of the intelligent support provided by the system.

Of course, visual inspection is generally not sufficient and may indeed be misleading depending on the total number of indicators. Generally, we need to know whether there is a statistically significant difference between a transition across different sessions. To do that, we run the 2 sample t-test as follows: Given a transition from a to b, we count the number of times such transitions happen for each session and each student, denoting this by N_i^j, where i is the student number ($i = 1 \ldots n$) and j is the session number. We form one vector for each session j with i ranging from 1 to n. The

elements of the vector are the number of times such a transition happens for a student. Continuing with the examples before, and focusing on indicator 6001, we found that a significant difference occurs (among other places) in the transition $6001 \rightarrow 1014$ between Session 1 and Sessions 2 and 3 (but not between Session 2 and Session 3). For Session 1, the transition probability is 0.278, while for Sessions 2 and 3, it is 0.472 and 0.493, respectively. In this case, this indicates the effectiveness of the system-generated feedback in helping students to formulate a correct rule (indicator 1014).

4. CONCLUSIONS

The volumes of data gathered from students' interactions in Exploratory Learning Environments (ELEs) makes interpretation of such data a difficult and costly task. The data is typically logged and stored in a manner that facilitates users' interactions with the ELE, but not necessarily subsequent analysis and decision making. Moreover, domain experts may not know in advance which data is relevant for analysis and decision making. In this paper we have addressed the question of how large volumes of student's interaction data gathered by an adaptive ELE can be visualised and explored in order to determine the effectiveness of the intelligent support being provided by the system and to improve this support. We have presented several possible visualisations of such data. We have discussed insights derived from these and how they can inform decisions with respect to further research and development. Although developed in the context of the MiGen system and its eXpresser microworld, our approach and visualisations are generic and could be applied to any ELE in which key indicators relating to students' interactions are detected or inferred by the system.

Currently, our visualisations require the support of a technical expert in order to create them, using either standard OLAP tools or ad-hoc visualisations (mostly generated using R scripts). We plan to improve both their interactivity and their ease of use in order to allow stakeholders with less technical expertise to be able to create such visualisations for themselves [7], to explore the data from their perspectives, and to formulate hypotheses for further investigation.
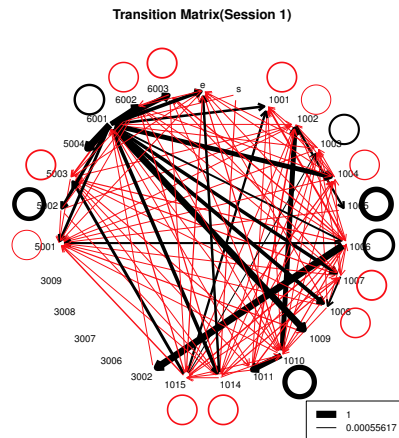
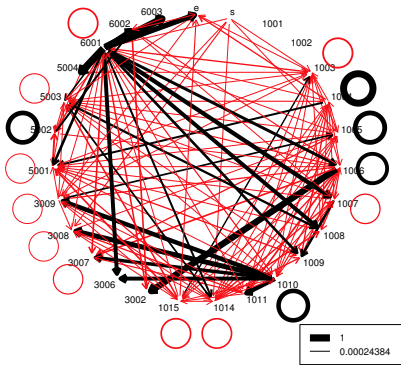**Figure 9: Outgoing Transition Matrix (Session 1)**

Figure 10: Outgoing Transition Matrix (Session 3)

5. REFERENCES

[1] R. Baker, J. Walonoski, N. Heffernan, I. Roll, A. Corbett, and K. Koedinger. Why Students Engage in "Gaming the System" Behavior in Interactive Learning Environments. *Journal of Interactive Learning Research*, 19(2):185–224, Apr. 2008.

[2] S. Gutierrez-Santos, E. Geraniou, D. Pearce-Lazard, and A. Poulovassilis. Design of Teacher Assistance Tools in an Exploratory Learning Environment for Algebraic Generalization. *IEEE Transactions on Learning Technologies*, 5(4):366–376, 2012.

[3] S. Gutierrez-Santos, M. Mavrikis, and G. D. Magoulas. A Separation of Concerns for Engineering Intelligent Support for Exploratory Learning Environments. *Journal of Research and Practice in Information Technology*, 44:347–360, 2013.

[4] R. E. Mayer. Should there be a three-strikes rule against pure discovery learning? The case for guided methods of instruction. *American Psychologist*, 59(1):14–19, 2004.

[5] R. Noss and et al. The design of a system to support exploratory learning of algebraic generalisation. *Computers and Education*, 59(1):63–82, 2012.

[6] C. Romero, P. Gonzalez, S. Ventura, M. Del-Jesus, and F. Herrera. Evolutionary Algorithms for subgroup discovery in elearning: A pratical application using Moodle. *Expert Systems with Applications*, 39, 2009.

[7] J. Steele and N. Illinsky. *Beautiful Visualization: looking at data through eyes of experts*. O'Reilly, 2010.

APPENDIX

Indicator Type	Name
1001	Building block made
1002	Pattern made
1003	Tile placed
1004	Unlocked Number changed
1005	Model Rule modified
1006	Number created
1007	Number Unlocked

Indicator Type	Name
1008	Number Locked
1009	Number Named
1010	A shape deleted
1011	All shapes deleted
1012	Task done
1013	Start task
1014	Correct Local rule created
1015	Incorrect Local rule created
1016	Correct Model Rule created
1017	Incorrect Model Rule created
2001	All Local Allocations correct
2002	Model is animated
2003	Inactive student
2004	Model consists only of Single Tiles
2005	Model consists only of Patterns
2006	Model has at least one Pattern
2007	Model has at least one uncorrected overlap
2008	All negative tiles used
2009	All variables named
2010	One or more variable named
2011	One or more constant named
2013	Last shape modified
3001	Construction evaluation
3002	Activity document answer right
3003	Activity document answer wrong - unknown reason
3004	Activity document answer wrong - scaling
3006	Building block made - plausible
3007	Building block made - implausible
3008	Pattern made - plausible building block
3009	Pattern made - implausible building block
4001	Task pattern built with unit tiles
4002	Plausible building block in use
4003	Animated unmessable pattern
4004	Animated apparent solution pattern
4005	Rhythm detected
4006	Spurious Title
4007	Pattern structure general - no shape detected
4008	Correct general allocation
4009	Pattern coloured generally
4010	Apparent solution on canvas
4011	Right amount of unlocked numbers
4012	Too many unlocked numbers
4013	No unlocked number
4014	Incorrect local allocation detected
5001	Goal checked by system
5002	Goal checked by student
5003	Goal unchecked by system
5004	Goal unchecked by student
6001	Intervention generated
6002	Intervention shown
6003	Feedback requested