Scalable Monitoring of Student Interaction Indicators in Exploratory Learning Environments

Alex Poulovassilis
Knowledge Lab, Birkbeck, University of London

Joint work with Sergio Gutierrez-Santos, Stefano Capuzzi, Ken Kahn (UCL IoE), Sokratis Karkalas

WebSTEd 2016 at WWW
April 2016

Knowledge Lab
Birkbeck
UNIVERSITY OF LONDON

Outline of the talk

1. Motivation
2. Case Study : the MiGen system
3. System implementation
4. Performance evaluation
5. Conclusions and future work

1. Motivation

- Much research and development work focusses on open-ended interactive educational applications that encourage students' experimentation within a knowledge domain
- 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
- Has led to research and development of intelligent techniques for providing adaptive support to students in order to foster their productive interaction with ELEs
- The role of the teacher in an exploratory learning setting is that of a ‘facilitator’ or ‘orchestrator’
- Supporting the teacher in this role poses several challenges, compounded by a learning setting in which students are each working with their own computer or handheld device

Motivation

- Due to the open-ended nature of the tasks that the students are working on, teachers can only be aware of what a small number of students are doing at any one time as they walk around the classroom

- It is therefore hard for the teacher to know which students are progressing with the task set, who is off-task, and who is in need of additional guidance from the teacher beyond the feedback being provided by the ELE itself

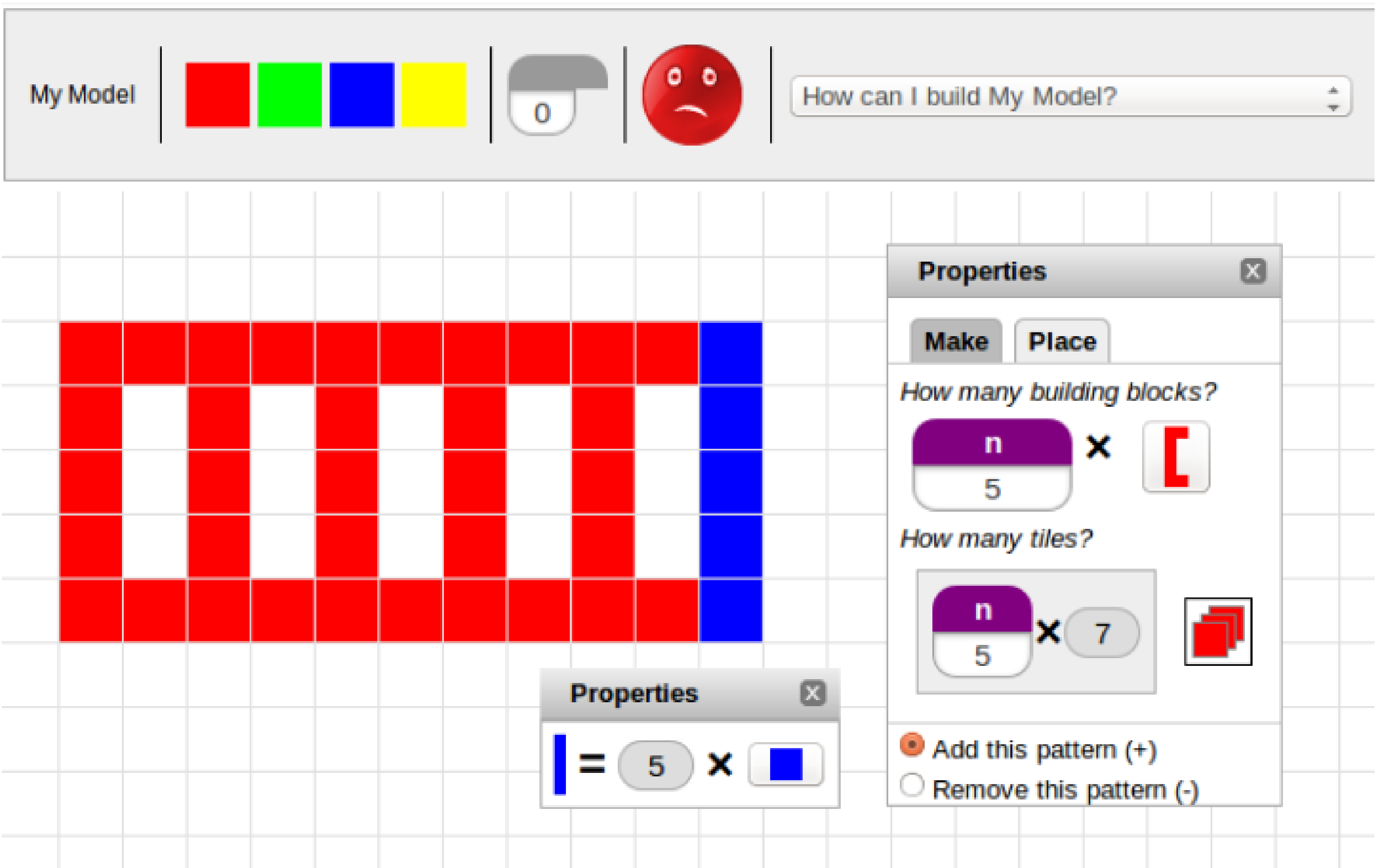
Motivation

- Has motivated our research into the design and deployment of **visualisation & notification tools** to be used by teachers in conjunction with ELEs
- The aim of such **Teacher Assistance (TA)** tools is to reduce the cognitive load on the teacher, increase the teacher's awareness of individual students' progress and of the classroom ‘state’ as a whole, and help the teacher to formulate his/her own timely guidance to individual students and to the class as a whole
- Earlier papers have described the design, prototype implementation, and evaluation of a suite of TA tools
- The present paper explores the **scalability** of provision of such tools, with the aim of providing them as **software-as-a-service**
- We present and evaluate a web-based architecture that can support such TA tools and demonstrate its scalability to allow concurrent usage by thousands of users

2. The MiGen system

- Our case study is the **MiGen** system (www.migen.org), an intelligent ELE that fosters 11-14 year old students' learning of algebraic generalisation
- In MiGen, students under take construction tasks in a microworld called ***eXpresser***
- These tasks ask students to create ‘generalised models’ consisting of 2-dimensional tiled, coloured patterns constructed from one or more building blocks
- 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

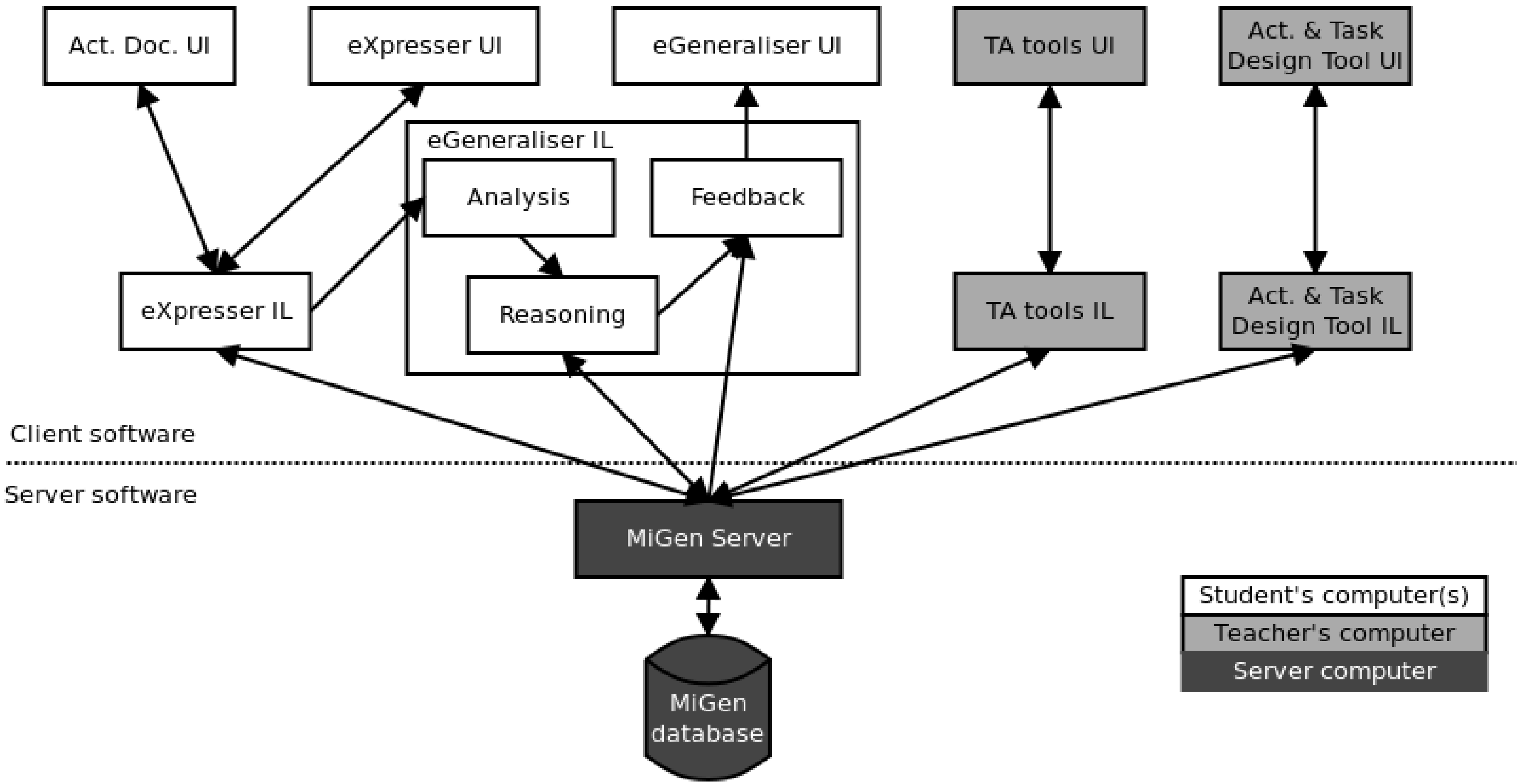
eXpresser microworld



Intelligent Support

- An intelligent component called the ***eGeneraliser*** provides both unsolicited and on demand personalised feedback to students
- This support relies on the detection of a set of ***task-independent*** (TI) and ***task-dependent*** (TD) ***indicators*** as students are interacting with the eXpresser
- Examples of TI indicators: student has placed a tile on the canvas, student has made a building block, student has unlocked a number
- Examples of TD indicators: 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
- TD indicators are detected by the eGeneraliser based on students' actions and on knowledge specific to the current task
- All the occurrences of TI/TD indicators are logged in the operational online MiGen database

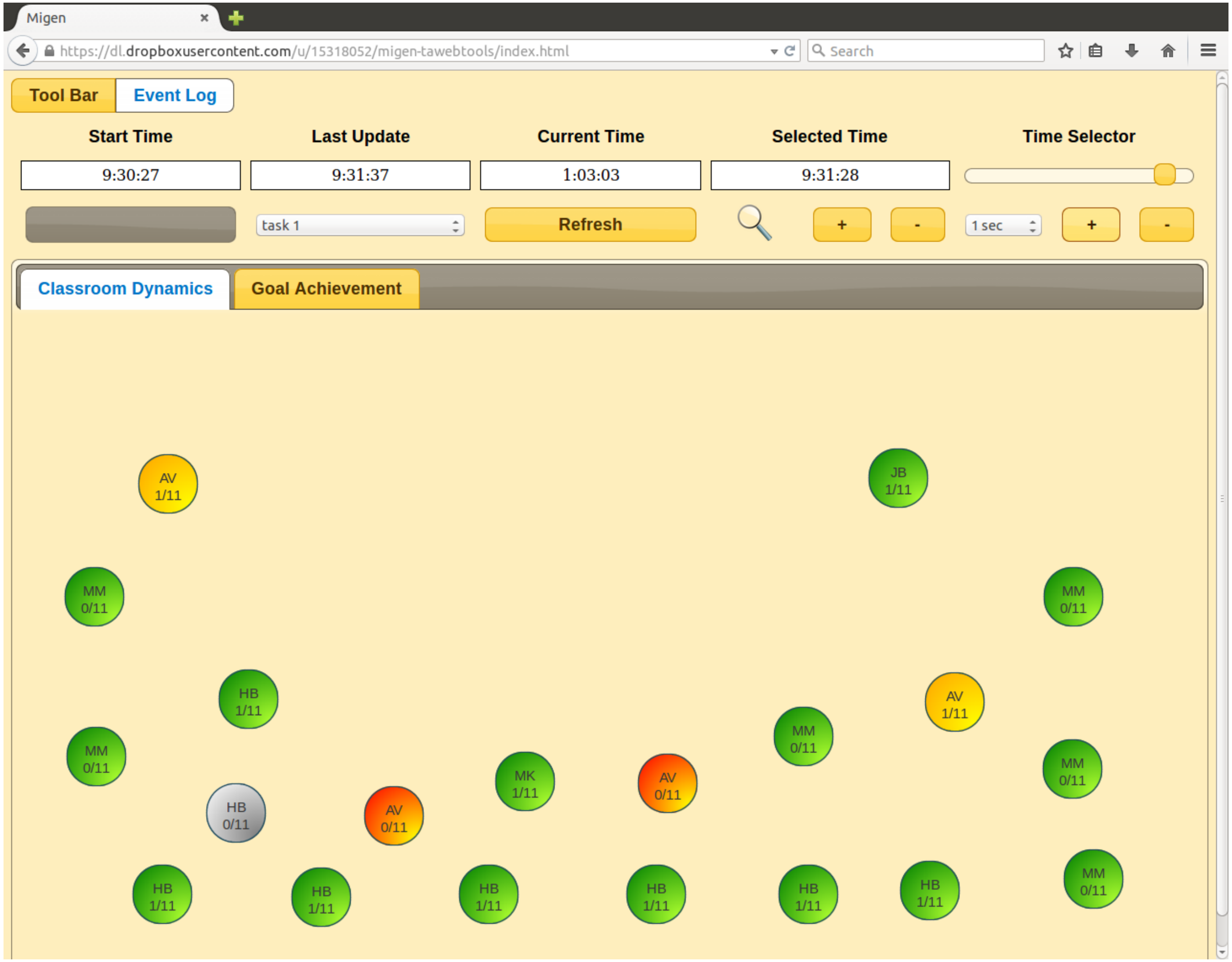
MiGen Logical Architecture



Teacher Assistance Tools

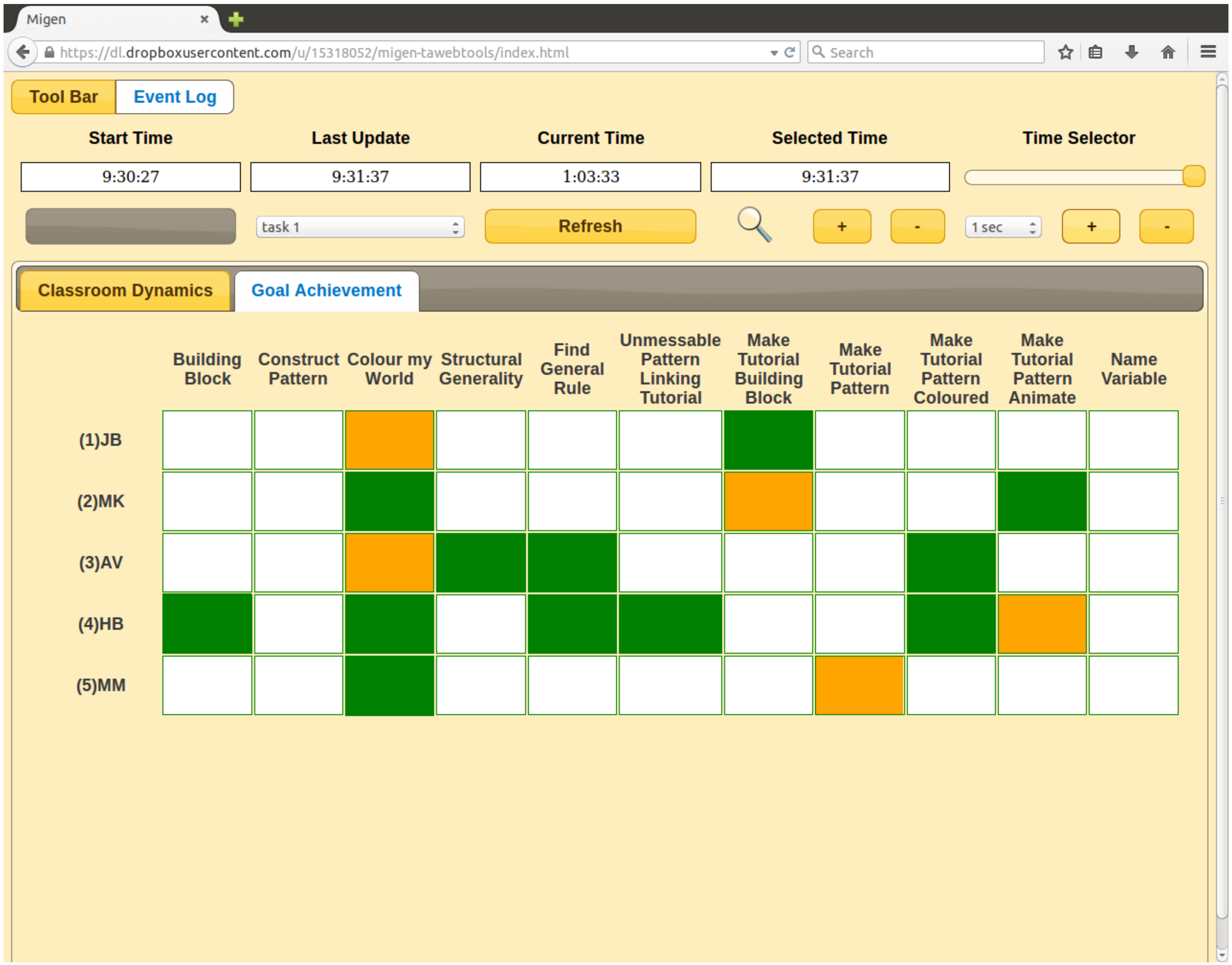
- The TA Tools receive real-time information from the MiGen Server relating to occurrences of TI and TD indicators for each student
- Each TA tool presents visually a selection of this information to the teacher
- MiGen's suite of TA tools includes
  - Student Tracking (ST),
  - Classroom Dynamics (CD) and
  - Goal Achievements (GA) tools
  - and also a Student Grouping tool

CD tool



Migen
https://dl.dropboxusercontent.com/u/15318052/migen-tawebtools/index.html
Search
Tool Bar
Event Log
Start Time
Last Update
Current Time
Selected Time
Time Selector
9:30:27
9:31:37
1:03:03
9:31:28
task 1
Refresh
+
-
1 sec
+
-
Classroom Dynamics
Goal Achievement
AV 1/11
JB 1/11
MM 0/11
MM 0/11
HB 1/11
AV 1/11
MM 0/11
MM 0/11
MK 1/11
AV 0/11
MM 0/11
HB 0/11
AV 0/11
HB 1/11
HB 1/11
HB 1/11
HB 1/11
HB 1/11
HB 1/11
MM 0/11

GA tool


Migen
https://dl.dropboxusercontent.com/u/15318052/migen-tawebtools/index.html
Search
Tool Bar
Event Log
Start Time
Last Update
Current Time
Selected Time
Time Selector
9:30:27
9:31:37
1:03:33
9:31:37
task 1
Refresh
+
-
1 sec
+
-
Classroom Dynamics
Goal Achievement
Building Block
Construct Pattern
Colour my World
Structural Generality
Find General Rule
Unmessable Pattern Linking Tutorial
Make Tutorial Building Block
Make Tutorial Pattern
Make Tutorial Pattern Coloured
Make Tutorial Pattern Animate
Name Variable
(1)JB
(2)MK
(3)AV
(4)HB
(5)MM

3. System implementation

- The initial implementation of the MiGen system was conceived to be installed and used locally in schools
- Each school has a copy of the MiGen server software installed on their own facilities
- Clients (i.e. students' and teachers' computers) connect to this server through the school's local area network

- The initial Java and REST-based implementation was sufficiently performant and scalable for this deployment scenario
- However, to allow larger-scale dissemination and uptake of the system in schools after the end of the MiGen project, we decided to re-implement the system as a cloud-based service

System implementation

- The MiGen software was ported onto Google's App Engine
- The eXpresser, eGeneraliser and TA tools became web browser applications
- The MiGen server and database became server software running in Google's cloud
- For the TA tools implementation, a 3-tier architecture was used, employing the Google App Engine at the back-end as a data store, and the cloud-based web service of Dropbox as the middle tier
  - the client tier is a relatively 'fat' browser-based application; it includes an in-memory database to cache information such as
    - the set of students whose progress is being monitored,
    - the tasks assigned to the students,
    - the indicator occurrences generated by these students' interactions with eXpresser
- See the paper for full technical details

System implementation

- The data format used for data interchange is XML
- The indicator data sent from the eXpresser web application to the MiGen server running in the Google cloud is stored in a NoSQL datastore
- A TA tools web application instance requests an update of the indicator data from the MiGen server every 5 seconds (by default – configurable)
- Updates are incremental:
  - the request is parameterised according to the time stamp of the previous such request and according to the classroom ID,
  - so only the data required by the specific TA tools application instance is sent by the MiGen server
  - this data is inserted into the local in-memory database and used by the TA tools application to instantly update its visualisations
  - the visualisation updates are partial, using Ajax

4. Performance Evaluation

- To test this new implementation in a scenario as close to reality as possible, we reused real data arising from a class of students working with the original MiGen system with their teacher in a school
- 15 students had interacted with the eXpresser in the context of a real Mathematics lesson for roughly one hour
- Approximately 1,000 indicator occurrences had been generated by these 15 students
- We downloaded this set of indicator occurrences from the MiGen database into a file and we replicated the indicator data as many times as needed, in order to simulate the activity of larger numbers of students

Performance Evaluation

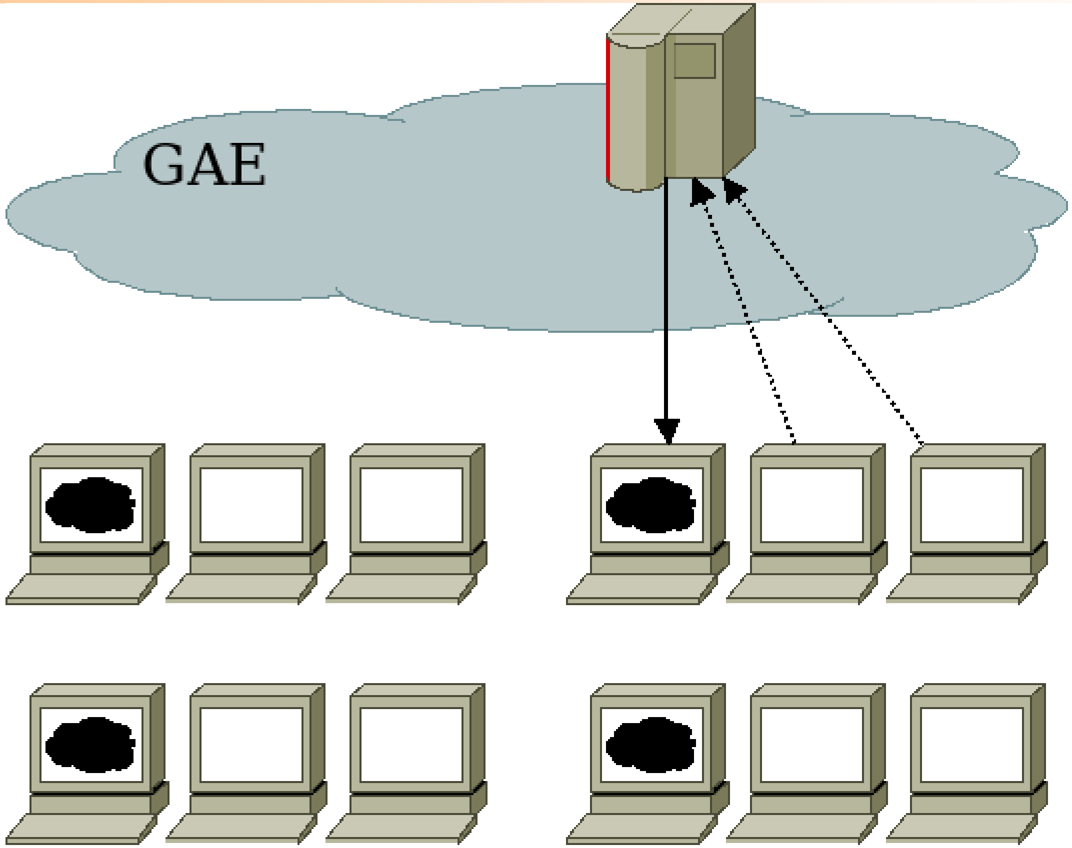
- We wrote a program to run on a client web browser that reads this indicator data file and generates the indicator occurrences contained in it at the specified times, just as if this was the MiGen client software running in the web browser
- From the point of view of the MiGen server software running in the Google cloud, all this activity looks like a set of students working with eXpresser concurrently, dispersed across their different classrooms

- We evaluated the scalability of the implementation by measuring the average ‘round-trip’ time taken by an indicator occurrence from the moment that it is generated by the client software until the moment after it is ‘pulled’ from the MiGen server by the TA tools application and processed so as to update the information being displayed by the TA tools UI

Performance Evaluation

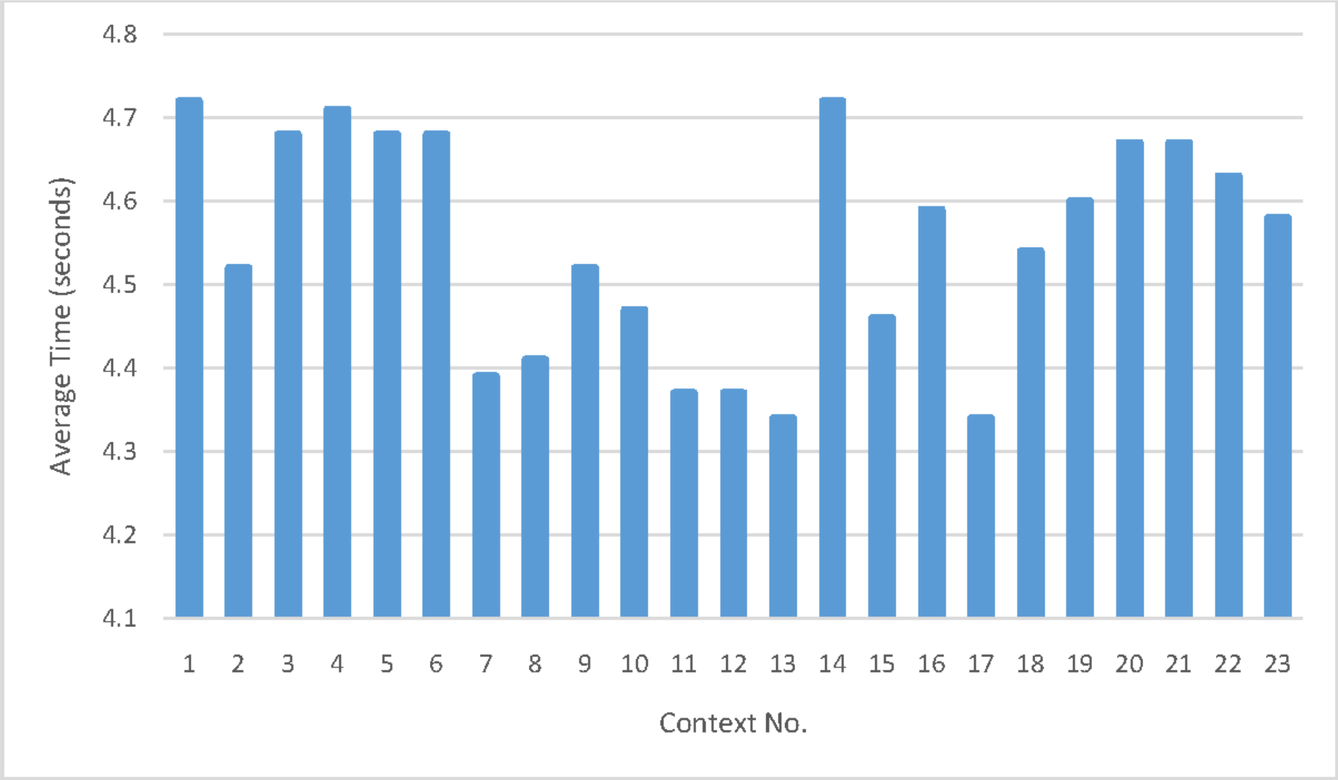
- Our initial set of simulations used one machine to generate all the indicator occurrences and one machine to run the TA tools application
  - reached an upper limit on linear performance at about 25,000 indicator occurrences over one hour (equivalent to the activity of a bit less than 400 concurrent students being monitored by one teacher, which is well above practical classroom sizes)
- We then used several web browsers in parallel to further test scalability, within the limits of the hardware resources available from our institution – a computer lab with 69 PCs, all connected to Google's cloud through the university network
- Two thirds of the PCs simulated 375 concurrent students each, while the remaining PCs each ran one instance of the TA tools application, drawing indicator occurrences from the MiGen server for the students being simulated by two corresponding ‘student’ computers

Performance Evaluation



Two ‘student’ computers simulate 375 students each, and the third computer runs an instance of the TA tools application, that retrieves indicator occurrences for all of these students. There were 23 of these triplets, 69 computers in total. The dashed lines represent the computers sending indicators as they occur in the simulation. The solid line represents the TA tools application requesting updates to its local cache of indicator occurrences every five seconds.

Performance Evaluation



The average round-trip time for the indicator occurrences in each one of the 23 'contexts', i.e. triplets of two PCs simulating students and one PC simulating a teacher, was measured. The average delay was under the 5-second upper bound (arising from the 5-second interval between indicator update requests) – an acceptable delay in the time that it would take for a teacher to be appraised of the current state of a classroom of students. The number of simulated indicator occurrences was approximately 1,150,000, corresponding to the activity of 17,250 students interacting concurrently with the MiGen system over one hour.

5. Conclusions and Future Work

- We have discussed the challenging role of teachers in supporting students who are working on exploratory learning tasks in the classroom
- Motivated the need to design **notification & visualisation tools** that support the teacher in being aware of the progress of individual students and of the classroom 'state' as a whole
- Presented a web-based architecture that can support such Teacher Assistance tools
  - described an implementation based on standard web and cloud computing technologies
  - demonstrated its scalability to allow concurrent usage by thousands of users, simulating concurrent classes of students and their teachers interacting with an ELE

Conclusions and Future Work

- Due to its usage of Google App Engine to store and retrieve the data relating to indicator occurrences, the scalability of our architecture is theoretically limitless

- Although developed and evaluated in the context of the MiGen ELE, our architecture is generic and could be used for monitoring students' progress in any ELE in which key indicators relating to student-system interactions are detected by the system as students work with the ELE

- Our work presented in this paper has demonstrated that it is technically feasible to provide such ELEs and accompanying Teacher Assistance tools at a national level as software-as-a-service.