

Generic log files and algorithms developed for educational multiplayer games

B.M. Monjurul Alom^{*}, Claire Scoular and Nafisa Awwal

Assessment Research Centre, Melbourne Graduate School of Education, Level 8, 100 Leicester Street, The University of Melbourne, Victoria 3010, Australia

Received: 20-August-2017; Revised: 06-November-2017; Accepted: 08-November-2017

©2017 ACCENTS

Abstract

There is potential for online multiplayer educational games to play a vital role to measure cognitive and social abilities among students. Educational games can be developed to capture student responses or actions, both shared and unshared, within the game environment. This paper presents a data capturing process in online games with a focus on the structural components required to measure sufficiently. The first part of this paper introduces a suggested structure of the log files and the second part describes generic algorithms used to extract information from the log files to indicate any observable cognitive and non-cognitive processes of the students. The term generic algorithm refers the common applicability of the algorithms to log files across games and students to provide researchers with rich and meaningful data about student behaviours. A considerable number of generic algorithms were developed to extract salient student behaviour from student activity log files, including intent to achieve efficiency in both present and future development of such complex educational endeavours.

Keywords

HTML5, Game, Collaborative, Problem solving, Log file, Generic algorithm.

1.Introduction

The use of online games within education has advanced in recent years and the integration of log file technology and learning analytics has provided a platform from which rich data can be captured about student behaviours and used to measure student learning. Students can be evaluated in real time in an automated method which decreases the time and effort that would be essential for teachers to perceive student performance in the classroom.

The online educational games presented in this study were designed to measure collaborative problem solving. A pair of students is required to work together in order to successfully progress through the game. Common data events are developed across all the tasks that lead to generic log files for the developed games. Once the generic log stream captures data gradually across tasks common skills or behaviours can be recognized within and across tasks.

Therefore, in order to capture the common behaviour of the students through the game, generic algorithms are developed to extract information from generic log files.

The algorithms were developed to capture behaviours that are established regardless of the role and perspective that students were offered with at the beginning of the game.

In order to deliver a reliable experience for students crosswise all browsers, platforms, and devices multiplayer games were developed in this study by CONSTRUCT2 platform that were initially designed by Flash, as part of the Assessment and Teaching of 21st Century Skills (ATC21S™) as presented by Awwal et al [1] and later in HTML5 in another consecutive study that branched out from ATC21S as presented by Alom et al. [2]. To overcome the task specific events and data from HTML5 platform, generic or common events across all the games on CONSTRUCT2 platform are developed to produce generic log files.

CONSTRUCT2 is an HTML5 based two dimensional (2D) game editor which provides the fast creation of games in a drag-and-drop fashion using a graphical editor and a behaviour-based logic system. The games are all dynamic in that students can move objects within their own screen, and across to their partner's screen. These games allow two students to cooperate by sharing resources and communicate through an embedded chat box. To measure the skills

^{*}Author for correspondence

of students, this facility can be used through their responses or actions within the game environment. An example of one of the developed games (Warehouse) is presented in *Figure 1* and *Figure 2*.

The methodology established by Hesse et al. [3] describes collaborative problem solving as comprised of two dimensions: social (collaborative) and cognitive (problem solving). Social regulation focuses on how students direct the collaborative space and includes negotiating and determining changes, assessing their self and their partner and taking accountability for the solving of the problem. On the other hand, the cognitive dimension refers to the ways in which students manage the problem space and apply their mental or perspective skills.

The games emphasis on collaborative problem solving; some of games were primarily developed by Zoanetti [4] for single student use in computer based problem solving. The development of online games for education has progressed rapidly in current years. The essential for digital games that can be transformed to permit students to play and express themselves during their learning is presented by Marchetti and Valente [5]. The methodology and mixture of the application SimApps within GameScapes is elaborated by Amresh et al. [6] and describes a novel technique of integrating rich narrative based storytelling within short repetitive game play procedure to inspire student learning and progress preservation. To integrate multiplayer use Masuch and Rueger's [7] paper has defined on teaching collaborative game design within a precise platform referred to as OpenCroquet. A generic framework of the game design process, the structure of digital games and the requirements for the use of OpenCroquet as a platform for collaborative game design is described. Generally, there is consensus, with commonly agreed 21st century skills including problem solving, critical thinking, collaboration, information technology literacy and creativity is described by Griffin et al. [8-12]. Construct2 is multiplayer game platform; the web source to develop multiplayer game is described in [13]. We have developed some games on this platform named as animal farm, rocket ship, and flower garden. How to measure the ability of the students using different assessment framework for collaborative problem solving tasks or games are described in [14-19].

The online educational games presented in this study were designed to measure collaborative problem solving, where pairs needed to work together to

successfully progress through the games. Development of games is a challenging effort. This paper discusses good practices and design implications for some of the fundamental elements of such intricate undertakings to yield efficiency in order to enable future scalability and progress.

2.Design of log files

Log files can provide innovative ways to capture, store and analyse what students do and say so that inferences can be drawn to develop further understanding of student learning. Log files were generated from student engagement with the games was used directly to identify actions, interactions and communications between students while solving problems, time allocated to the steps, and lack of action.

A unique feature of the log files designed in this study was that they contain consistent events across games and students. Therefore, all game events are more common, rather than unique to every game played by the students. This allows for easier design, interpretation and analysis of the log files. The log files, capture data consistently so that common behaviours can be readily identified within and across games. Log file analysis can take the form of quantitative or qualitative analysis and can be conducted manually or automatically, this study adopting multiple approaches. As a result, regular patterns of activity or behaviours were identified in the log files. An example of generic log stream data is presented in *Table 2*. Each data event is captured in a new row. The event columns display the common data events and different information about the event across three variable columns. We have also presented a task specific log stream (non-generic) data in an appendix in *Table 8*. Both logs represent different data events for the same tasks for the same team player.

2.1Database structure

Data is collected from the games through the capture of distinct keystrokes and mouse events such as typing, clicking, dragging, cursor movements, hovering time, and action sequences and so on. This data is recorded in the database structure. Each row of the database represents a meaningful piece of information. Each participant's response is recorded as separate attributes as single instances with corresponding user identification, present state, timestamp, record index and other data as deemed necessary by researchers for the task analysis. The database consists of several columns labelled as (id,

student_id, task_id, role, page, event, arg (arg1, arg2, arg3), timestamp.

Id represents the number of the lines sequentially in the log file. Team_id presents the id that links both paired students together, this is the same number. Student_id allows for identification of a particular student. Both students in a pair will have the same number sequence, with a different letter after to differentiate between students. Bundle_id represents the assessment being taken. Task_id allows for identification of which task the events took place. Role helps differentiate between students and is represented by a letter which is the same letter at the end of their student ID (A or B). Page refers to the page in which the event took place within. Event refers to the common data event occurring. These usually refer to the resources and information in the task that are being used. These may vary across tasks for different tasks will use different resources and information. Arg (arg 1, arg 2, arg 3) refer to the variables or arguments that are captured within that. These columns represent more detail information regarding a particular event. Timestamp represents the time and date of the event to milliseconds.

2.2 Event design

The data events capture all actions between students and game, and student, as well as all text entered into the chat box by each student. During game play every action (selection, movement and clicks) and communication completed by each student is captured in the log file in a time linear structure. It is important that every event is captured in the database as each action and communication, even if ineffective for collaborative problem solving, is used to interpret the students' performance and experience in the game. Within log files, multiple actions may occur

simultaneously, so this needs to be clearly distinguished from other activities for both the order of their occurrence as well as a type of the activities.

The most common events designed for the games are labelled as start, move-resource, chat, pass-resource, rule-confirm, select-choice, solution-found, enter-value, team-success, review, summary, share-view, retry. The definition of each event is presented in *Table 1*.

Table 1 Description of each event

| Event | Description |
|----------------|--|
| Start | Both students have entered the task. |
| Chat | The student sends a chat in the chat box. |
| Pass-resource | The student sends resource to partner. |
| Move-resource | Moves a resource from one place to another. |
| Enter-value | The student enters text or value into the text box. |
| Select-choice | The student selects an option. |
| Team-success | Both players have successfully solved the problem. |
| Review | Reviews of previous selection/choice/answer. |
| Rule-confirm | Confirms the rule they have entered/selected. |
| Retry | Sets all the objects or resources and movement to its initial position. |
| Finish | Both students have selected to finish the task. |
| Share-view | Captures the view of each other contents, when Host or player A will click share view button, Peer would be able to see the contents of player A, but would not be able to move or drag. |
| Solution-found | When the solution is obtained on the task. |

Table 2 Database structure of the generic log file

| ID | Team_ID | Student_ID | Bundle_ID | Task_ID | Role | Page | Event | Arg1 | Arg2 | Arg3 | Timestamp |
|---------|------------|-------------|-----------|------------|------|------|---------------|--------|------|------|-----------------|
| 1647433 | auvarc0033 | auvarc0033b | 12 | ICP0903025 | B | 1 | Request-page | 2 | | | 22/08/201711:55 |
| 1647432 | auvarc0033 | auvarc0033a | 12 | ICP0903025 | A | 1 | Request-page | 2 | | | 22/08/201711:55 |
| 1647430 | auvarc0033 | auvarc0033b | 12 | ICP0903025 | B | 1 | Select-choice | opB | 1 | | 22/08/201711:54 |
| 1647431 | auvarc0033 | auvarc0033b | 12 | ICP0903025 | B | 1 | Select-choice | result | 14 | | 22/08/201711:54 |
| 1647429 | auvarc0033 | auvarc0033a | 12 | ICP0903025 | A | 1 | Select-choice | opA | 1 | | 22/08/201711:54 |
| 1647428 | auvarc0033 | auvarc0033a | 12 | ICP0903025 | A | 1 | Select-choice | opA | 4 | | 22/08/201711:54 |
| 1647427 | auvarc0033 | auvarc0033b | 12 | ICP0903025 | B | 0 | Request-page | 1 | | | 22/08/201711:54 |

| | | | | | | | | | | | | |
|-------|--------|-----------|----|---------|---|---|----------|--------|-----------|-------|--|------------|
| 16474 | auvarc | auvarc003 | 12 | ICP0903 | A | 0 | Request- | 1 | | | | 22/08/2017 |
| 26 | 0033 | 3a | | 025 | | | page | | | | | 11:54 |
| 16474 | auvarc | auvarc003 | 12 | 103 | B | 1 | Finish | | | | | 22/08/2017 |
| 25 | 0033 | 3b | | | | | | | | | | 11:53 |
| 16474 | auvarc | auvarc003 | 12 | 103 | A | 1 | Finish | | | | | 22/08/2017 |
| 24 | 0033 | 3a | | | | | | | | | | 11:53 |
| 16474 | auvarc | auvarc003 | 12 | 103 | A | 1 | Chat | used 2 | | | | 22/08/2017 |
| 23 | 0033 | 3a | | | | | | | | | | 11:50 |
| 16474 | auvarc | auvarc003 | 12 | 103 | B | 1 | Move- | ball5 | R | B-R | | 22/08/2017 |
| 22 | 0033 | 3b | | | | | resource | | | | | 11:49 |
| 16474 | auvarc | auvarc003 | 12 | 103 | A | 1 | Move- | ball0 | R | A-R | | 22/08/2017 |
| 21 | 0033 | 3a | | | | | resource | | | | | 11:49 |
| 16474 | auvarc | auvarc003 | 12 | 103 | B | 1 | Move- | ball11 | L | B-L | | 22/08/2017 |
| 20 | 0033 | 3b | | | | | resource | | | | | 11:49 |
| 16474 | auvarc | auvarc003 | 12 | 100 | B | 1 | Finish | | | | | 22/08/2017 |
| 16 | 0033 | 3b | | | | | | | | | | 11:47 |
| 16474 | auvarc | auvarc003 | 12 | 100 | A | 1 | Finish | | | | | 22/08/2017 |
| 15 | 0033 | 3a | | | | | | | | | | 11:47 |
| 16474 | auvarc | auvarc003 | 12 | 100 | B | 1 | Pass- | oil | completed | [0,4] | | 22/08/2017 |
| 11 | 0033 | 3b | | | | | resource | | -transfer | | | 11:44 |
| 16474 | auvarc | auvarc003 | 12 | 100 | B | 1 | Move- | 0 | base | [0,4] | | 22/08/2017 |
| 12 | 0033 | 3b | | | | | resource | | | | | 11:44 |

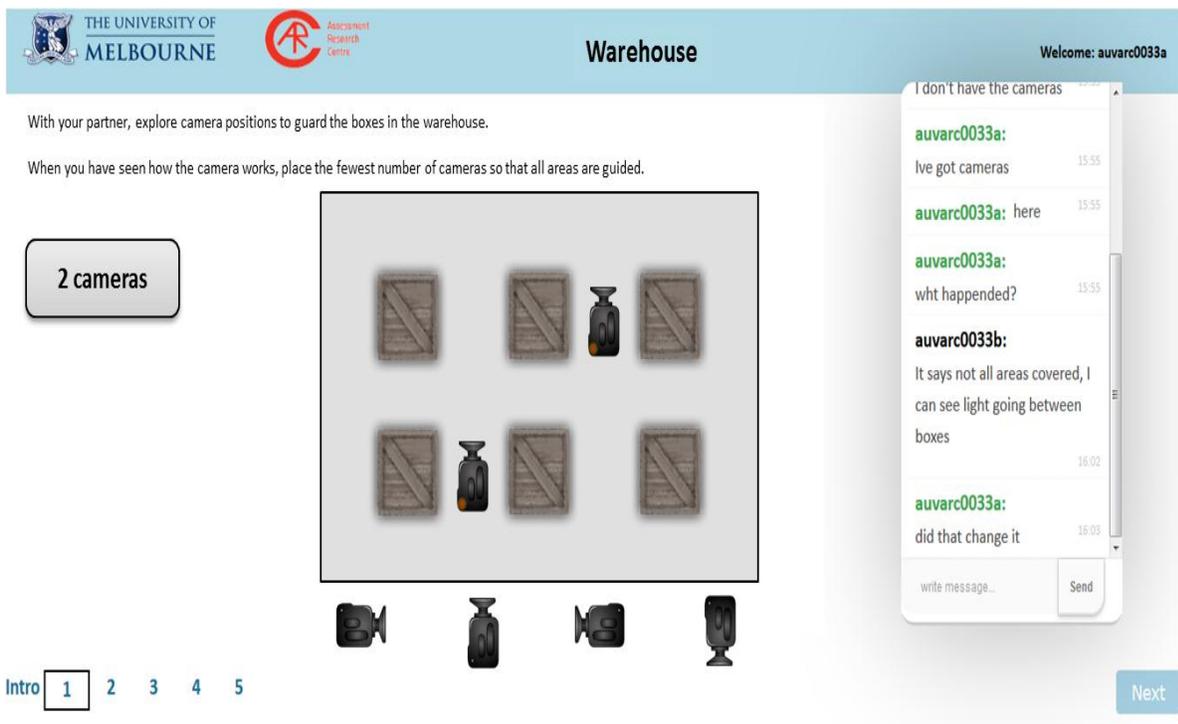


Figure 1 Student A’s screen view for warehouse game

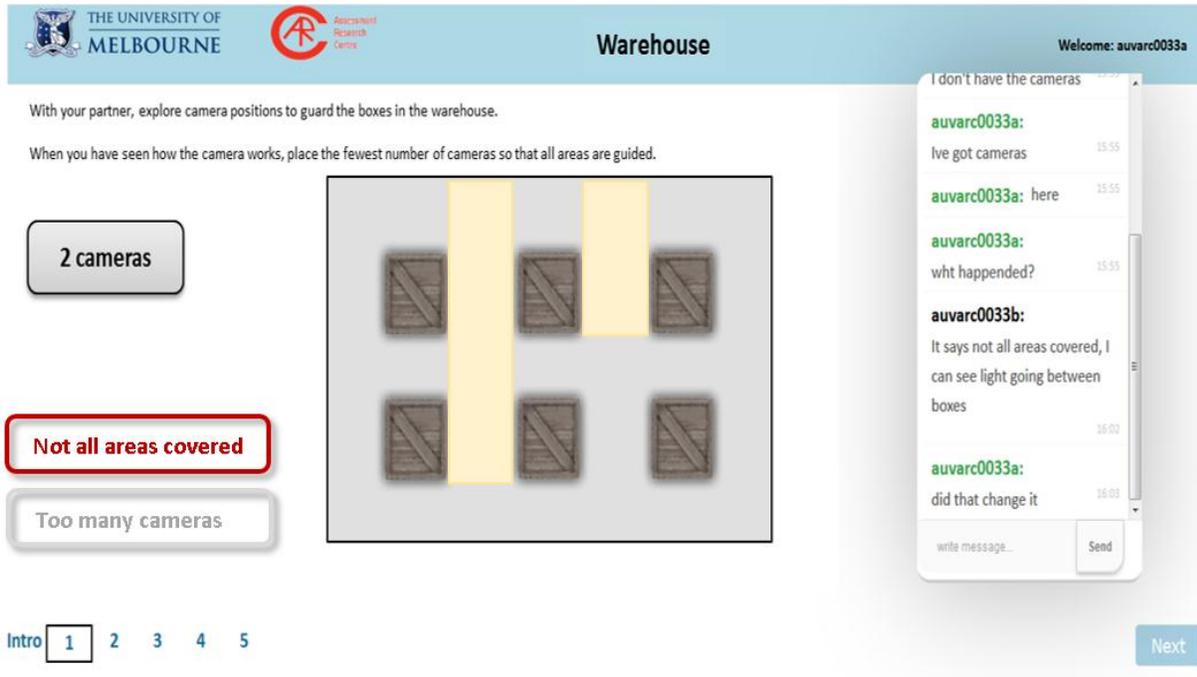


Figure 2 Student B’s screen view of warehouse game

3.Design of algorithm

Sequences of data events, (e.g. Chat-action-chat, action-chat-action) were identified and interpreted in relation to student learning of collaborative problem solving skills. Scoring algorithms were coded to search the log files for the data event sequences.

The scoring algorithms work as an input/output flow. The patterns of events, or indicators identified in the log stream data are considered the input and a record file based upon the algorithms considered the output. The general searching technique of the algorithm is presented in the *Figure 3*. One of the examples of the scored file (generated by the algorithm) is presented in the *Table 4*.

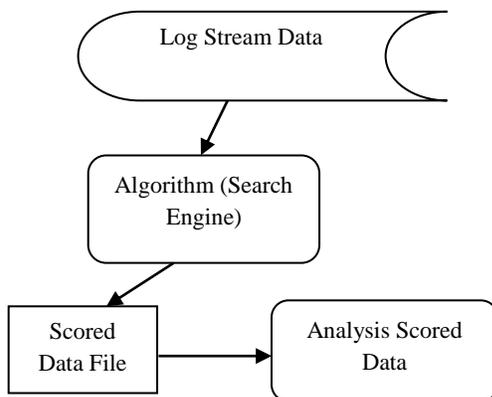


Figure 3 General search technique

4.Method

Each algorithm searches for the data event sequence for every student in each game they play. The algorithms are then treated like items on a test, where a student receives a score of ‘1’ if the algorithm finds that pattern and a ‘0’ if it does not find that pattern. The pattern of responses from the scoring algorithms provides identification of the patterns of events that were found and allow for interpretation of student learning.

An advantage of this approach is that, since generic data events were developed during the data capturing design process, generic algorithms could then be developed. The term generic algorithm refers to the applicability of the patterns across all the games. Two generic algorithms are described in details in section 4.1 and section 4.2. The algorithm presented in section 4.1, represents a chat-chat pattern which suggests that students are responding to one another.

This pattern could measure student communication, interaction and broader social ability.

The algorithm presented in section 4.2, presents an action-action-action. When specific actions are considered in the variable event, this pattern indicates whether the students try each resource in turn or sequentially, and could be used to measure their cognitive skill to work systematically or strategically

to solve a problem. To implement the algorithm the important definitions are presented in *Table 3*.

Table 3 Definitions to implement algorithm

| Event | Description |
|-----------------|--|
| Chat | The information is recorded in the database, when the players communicate with each other using chat box. |
| Self | Every participant is known as self and his team member is known as a partner. |
| Id | This represents the number of the lines sequentially in the log file. |
| Minimum Id | This represents the starting Id of the game. |
| Maximum Id | This represents the end Id of the game. |
| Distinct Action | During game play every action (selection, movement and clicks) and communication completed by each student is captured in a log file in a time linear structure. Unique action is referred as distinct action. |
| Block | Represents repeated occurrence of either chat or Action. |

To implement the example algorithm 1 in section 4.1, three steps are described to address the rule of this algorithm ‘to find a pattern of partner chat immediately followed by self-chat’. At step 1, the algorithm search the partner chat Id from the log. Then the algorithm immediately tries to find self-chat after that partner chat Id generated from step1. The algorithm scores ‘1’ in step 3, if the conditions are satisfied in step1 and in step 2, otherwise scores ‘0’.

To implement the example algorithm 2 in section 4.2, three steps are presented to capture the rule of this algorithm ‘to find a block of pattern of three distinct actions by self’. At step 1, the algorithm search for a block of distinct action of self. Then the algorithm at step 2, try to check whether the block generated at step1 is unique/distinct. The algorithm scores the ‘1’ in step 3, if the conditions are satisfied in step1 and in step 2, otherwise scores ‘0’, This process repeat until to finish the whole log.

Table 4 Scored data from the log file after search algorithm

| Student ID | Soco001 HEX | Soco001 CLO | Soco001 OIL | Cosy001 HEX | Cosy001 CLO | Cosy001 OIL | Comr001 HEX | Comr001 CLO | Comr001 OIL |
|-------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| auvarc0001a | -1 | 1 | 0 | 0 | -1 | 1 | 0 | 0 | 0 |
| auvarc0001b | -1 | 0 | 1 | 1 | -1 | 0 | 0 | 1 | 0 |
| auvarc0002a | -1 | -1 | 1 | 1 | -1 | 0 | 0 | 0 | 0 |
| auvarc0002b | -1 | -1 | 1 | 1 | -1 | 1 | 0 | 0 | 1 |
| auvarc0003a | -1 | -1 | 1 | 1 | -1 | 0 | 0 | 1 | 0 |
| auvarc0003b | -1 | -1 | 1 | 1 | -1 | 0 | 0 | 0 | 1 |
| auvarc0007a | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | -1 |
| auvarc0007b | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| auidem0014a | -1 | 0 | 1 | 1 | -1 | 1 | 0 | 1 | 0 |
| auidem0014b | -1 | 1 | 1 | 0 | -1 | 0 | 0 | 0 | 1 |
| auvkon0029a | 1 | 1 | 1 | 1 | 1 | 0 | 1 | -1 | 0 |
| auvkon0029b | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |

4.1 Example algorithm 1

Rule: Find a pattern of partner chat immediately followed by SELF chat. If the pattern is found, then score 1; else score 0. If there is no chat from partner, then score -1; means missing value.

Begin

Step 1: Find the partner chat Id from the log file.

Step 2: Find any chat from self immediately followed by partner chat Id.

Step 3: If step 1 and step 2 is found, then score 1, otherwise score 0. If step 1 is not found, then score as missing value -1.

End

4.2 Example algorithm 2

Rule: Find a block of pattern of three distinct actions by self.

Begin

Step 1: Find a block of 3 actions by self, starting from Minimum Id.

Step 2: Check whether the block of actions is distinct or unique at step 1?

Step 3: If step 2 is true, then score the value 1, otherwise try to find the next block until to reach Maximum Id.

Step 4: If step 2 is false, then the Minimum Id of the next block will be the last action Id generated at step 1.

Step 5: If no block found, then score 0. If no action is found then missing value, score-1.

End

5.Experimental results

In this study, there has been a considerable number of the algorithms (more than 600) written in PHP language. PHP is a widely-used open source general-purpose server scripting language designed primarily for web development. The multiplayer architecture is hosted on a remote server and includes several components, including, Linux, Apache HTTP Server, MySQL and PHP [2]. The database is presented as a relational structure and the application packages are configured to support the various target languages and output, the resultant game view at the client’s end. The Server is set up just as before, a MySQL database, with PHP support.

In total 42 generic algorithms were developed. To better understand student learning in the game environment the frequency with which each pattern occurred was analyzed. The frequency analysis of Example algorithm 1 indicator is presented in Table 5, for two different games olive oil and laughing clowns. The algorithm indicates that 68% of students in olive oil and 57% in laughing clowns completed the chat-chat pattern. This suggests that greater than half the students demonstrated social skills by responding to one another during game play. 26.2% students were scored as missing means either partner or self-did not participate in chat communication. The percentage is presented for this indicator in the olive oil task in Figure 4. Similarly, the percentage of this indicator for the laughing clown task is presented in Figure 5. Student’s interaction and communication in task laughing clown were more than that of the task olive oil (Figure 4 and Figure 7).

Table 5 Percentage distribution of the indicator SOCO001 for olive oil task

| Scored value | SOCO001 OIL | | SOCO001 CLO | |
|--------------|-------------|-----|-------------|------|
| | Frequency | % | Frequency | % |
| -1 | 471 | 26 | 277 | 5.3 |
| 0 | 109 | 6 | 1958 | 37.6 |
| 1 | 1220 | 68 | 2972 | 57.1 |
| Total | 1800 | 100 | 5207 | 100 |

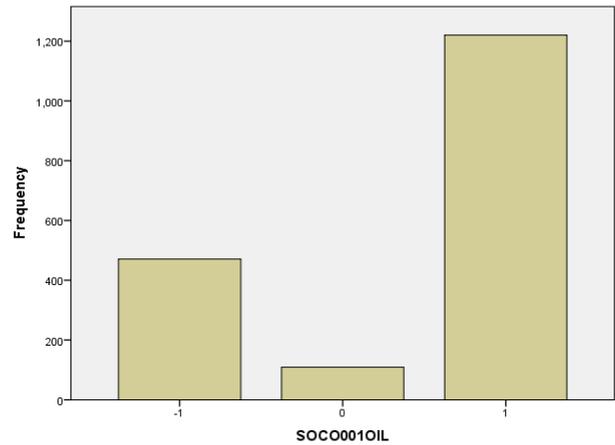


Figure 4 Frequency distribution of the indicator SOCO001 for olive oil task

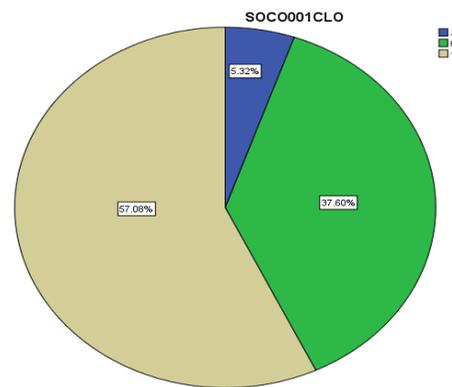


Figure 5 Frequency analysis of the indicator SOCO001 for laughing clown task

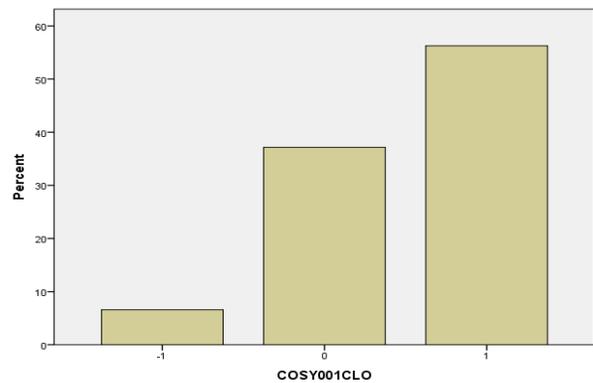


Figure 6 Frequency (in percentage) analysis of the indicator COSY001 for laughing clown task

Example algorithm 2 presents an action-action-action pattern with consideration for the specific activities to identify potential systematicity indicates the behaviour pattern of distinct actions by the student

that leads to the systematic approach whether the students try each resource in turn or sequentially. The graphical representation of this indicator for the olive oil and the laughing clown task is presented in *Figure 6*, and *Figure 7*. These 2 graphs show that the students were more systematic about doing action on the task olive oil than that of laughing clown. The necessity for development of games of this complexity is efficiency. *Table 6* presents a comparison of development time and coding length of two if the ATC21S™ games: olive oil and balance beam. We have presented the coding efficiency improvement for CONSTRUCT2 platform in [20].

Searching time of the algorithms is presented in *Table 7*. From the searching time, it is evident that the time for generic algorithms are consistent across the game with respect to the number of participants in the game. Although non-generic algorithm taking less time to extract information from log files, but these non-generic algorithms are task specific not applicable across all the tasks. Non-generic algorithms are also simple to extract information from the log file. Most of the generic algorithm finds the behaviors pattern from the whole log which takes search time bit longer.

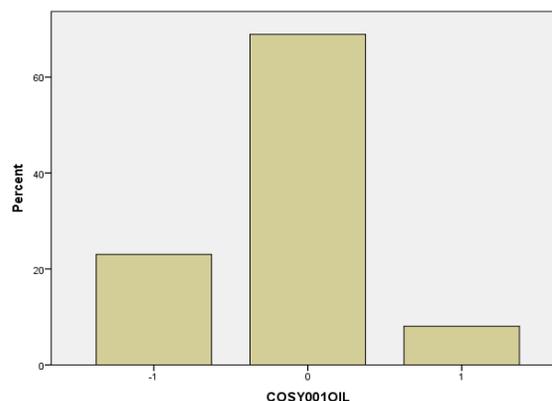


Figure 7 Frequency (in percentage) analysis of the indicator COSY001 of olive oil task

Table 6 Comparison of game completion time and coding complexity [20]

| Platform | Game name | Time taken | Coding length |
|------------|--------------|------------|---------------|
| HTML5 | Olive Oil | One month | 3738 lines |
| Construct2 | Olive Oil | Three Days | 1723 lines |
| HTML5 | Balance Beam | Two months | 29369 lines |
| Construct2 | Balance Beam | Two weeks | 17913 lines |

Table 7 Searching time of algorithms

| No. of records in database | Game name | Type of log file | Type of algorithm | Total no. of algorithms | Time taken | No. of students participated in game |
|----------------------------|---------------|------------------|-------------------|-------------------------|--------------|--------------------------------------|
| 1.6 million | Animal farm | Generic | Generic | 42 | 13.4 minutes | 624 |
| | Warehouse | Generic | Generic | 42 | 12.3 minutes | 301 |
| | Rocket ship | Generic | Generic | 42 | 15 minutes | 848 |
| 1.99 million | Olive oil | Not generic | Game specific | 20 | 28.8 seconds | 2189 |
| | Balance | Not generic | Game specific | 11 | 1.02 minutes | 3545 |
| | Game of 20 | Not generic | Game specific | 31 | 49.5 seconds | 2483 |
| | Small Pyramid | Not generic | Game specific | 36 | 27.5 seconds | 1055 |

6. Discussion

It is apparent from the log file extraction process that common behavior patterns can be found across students and tasks. The generalized algorithm has played a role in identifying, coding, searching, and extracting of behaviours from generic log files. The behavioural patterns are common across students, tasks, and assessments are an essential emphasis in

this study. From the analysis of algorithms across assessments it was clear that fewer students communicated with, their partner before to reaching the final goal or solution in the first game in each assessment than they did in subsequent games. Students communicated more repeatedly as each assessment progressed and communication became easier.

The algorithms identify whether students respond to their partner, answer their questions, and incorporate their partner's contributions into their own activities. These give a measure of a student's ability to value the contributions that others make. This is also a critical skill in the definition of collaboration outlined by the Partnerships 21 framework [21]. Some algorithms identify the student's ability to communicate knowledge and share resources which are aligned with a perspective of measuring interaction in collaborative tasks [22]. Given that action is considered to be the best indication of student willingness and readiness to be successful in collaborative problem solving tasks [23], it is covered well in the assessment through some algorithms.

The majority of algorithms measure the social behaviour, while the remainder measures the cognitive behaviour. Social behaviour refers to student engagement with the task, the extent to which they persevere to solve the problem and how well they interact with others. Social behaviour also focuses on the quality of the interaction between students during collaborative problem solving such as how students respond and adapt to one another. Generally, the cognitive dimension refers to the ways in which students manage the problem space and apply their reasoning skills.

7. Conclusion

The games were developed based on the principles of common features and event structure. Common game features are a significant strategy to permit easy identification of common behaviours across games. These common features allow the same search algorithm to be applied to each game in order to identify, code and extract information from log files. In this study, we describe the generic algorithms that capture the observable behaviours from log files generated from multiplayer educational games. The generic algorithm is used to make inferences about the performance of the students demonstrated in the game. In essence, the generic algorithms capture information from the log files that reflects the social and cognitive skills of the students. Time complexity analysis, efficiency of the generic algorithms to capture generic logs from CONSTRUCT2 platform, over search algorithms to capture logs from HTML5 platform is the future research work.

Acknowledgment

Our thanks to the Assessment Research Centre project members carrying out research work in the area of 21st century skills.

Conflicts of interest

The authors have no conflicts of interest to declare.

References

- [1] Awwal N, Griffin P, Scalise S. Platforms for delivery of collaborative tasks. In *assessment and teaching of 21st century skills 2015* (pp. 105-13). Springer Netherlands.
- [2] Alom M, Awwal N, Scoular C. Technology integration in multiplayer game design. In *European conference on games based learning 2015* (pp. 10-4). Academic Conferences International Limited.
- [3] Hesse F, Care E, Buder J, Sassenberg K, Griffin P. A framework for teachable collaborative problem solving skills. In *assessment and teaching of 21st century skills 2015* (pp. 37-56). Springer Netherlands.
- [4] Zoanetti N. Interactive computer based assessment tasks: how problem-solving process data can inform instruction. *Australasian Journal of Educational Technology*. 2010; 26(5):585-606.
- [5] Marchetti E, Valente A. Design games to learn: a new approach to playful learning through digital games. In *proceedings of the European conference on games based learning 2014* (pp. 356-63).
- [6] Amresh A, Clarke D, Beckwith D. GameScapes and SimApps: new techniques for integrating rich narratives with game mechanics. In *proceedings of the European conference on games based learning 2014* (pp. 18-25).
- [7] Masuch M, Rueger M. Challenges in collaborative game design developing learning environments for creating games. In *international conference on creating, connecting and collaborating through computing 2005* (pp. 67-74). IEEE.
- [8] Griffin P, McGaw B, Care E. *Assessment and teaching 21st century skills*. Springer Netherlands; 2012.
- [9] Griffin P, Care E. *Assessment and teaching of 21st century skills: methods and approach*. Springer Netherlands; 2015.
- [10] Griffin P, Care E, Harding SM. Task characteristics and calibration. In *assessment and teaching of 21st century skills 2015* (pp. 133-78). Springer Netherlands.
- [11] Griffin P, Care E, editors. *Assessment and teaching of 21st century skills: methods and approach*. Springer Netherlands; 2014.
- [12] Nelson LM. Collaborative problem solving. *Instructional design theories and models: a new paradigm of instructional theory*. Lawrence Erlbaum Associates, Inc.; 1999, p.241-67.
- [13] <https://www.scirra.com/construct2>. Accessed 26 July 2017.
- [14] Crockett L. HTML5 canvas, user illusions and game flow. In *European conference on games based learning 2014* (pp. 68-76). Academic Conferences International Limited.
- [15] Adams R, Vista A, Scoular C, Awwal N, Griffin P, Care E. Automatic coding procedures for collaborative problem solving. In *assessment and teaching of*

- century skills 2015 (pp. 115-32). Springer Netherlands.
- [16] Care E, Griffin P, Scoular C, Awwal N, Zoanetti N. Collaborative problem solving tasks. In assessment and teaching of 21st century skills 2015 (pp. 85-104). Springer Netherlands.
- [17] Care E, Scoular C, Bui M. Australia in the context of the ATC21S Project. In assessment and teaching of 21st century skills 2015 (pp. 183-97). Springer Netherlands.
- [18] Woods K, Mountain R, Griffin P. Linking developmental progressions to teaching. In assessment and teaching of 21st century skills 2015 (pp. 267-92). Springer Netherlands.
- [19] Care E, Griffin P. An approach to assessment of collaborative problem solving. *Research & Practice in Technology Enhanced Learning*. 2014; 9(3):367-88.
- [20] Alom BM, Scoular C, Awwal N. Multiplayer game design: performance enhancement with employment of novel technology. *International Journal of Computer Applications*. 2016; 145(1):27-32.
- [21] Trilling B, Fadel C. *21st century skills: learning for life in our times*. John Wiley & Sons; 2009.
- [22] Zagal JP, Rick J, Hsi I. Collaborative games: lessons learned from board games. *Simulation & Gaming*. 2006; 37(1):24-40.
- [23] Stasser G, Vaughan SL. *Models of participation in face-to-face unstructured discussions*. Davis JH and E. Witte, Hillsdale: Erlbaum. 1996.



Dr. B.M. Monjurul Alom is a Programmer at the Assessment Research Centre, University of Melbourne, Australia. He completed a PhD (Computer Science) in 2011 at the University of Newcastle, Australia, where he worked as a research fellow from 2011–12. He was an Assistant

Professor at Dhaka University of Engineering & Technology from 2000–2007. His work includes the development of collaborative web applications as a basis for research in the area of educational assessment. He has developed a registration portal system, web database application (multiplayer game) and scoring engine to assess the ability of students, which is being used in schools in Australia and Internationally. His research also includes data mining and management for large scale databases. Monjurul also works on software development specifically for the assessment of 21st century skills, including collaborative problem solving. Monjurul has proficiency in web programming (game development, database application), server administration.

Email: monjurul.alom@unimelb.edu.au



Dr Claire Scoular's work focuses on the improvement of measurement practices in education. Her expertise is embedded in the application and statistical analysis of psychological measurement. Recent investigations have centered around measurement methodology and application of 21st

century skills, including traditional and automated test delivery systems.



Nafisa specialises in the design, development and implementation of web-based educational assessments, reporting and other data collection tools. She manages the system integration and architecture for traditional and innovative, individual and collaborative assessments for students in multiple languages. Her research includes data management, analysis, and item writing.