DATA-DRIVEN ADAPTIVE LEARNING ENVIRONMENT FOR PROJECT-BASED CONSTRUCTION ENGINEERING

Mahadevan Subramaniam,* Parvathi Chundi,* and James D. Goedert**

Abstract

Serious computer games have led to a wide resurgence of game-based education and training systems. Intelligent serious games (*ISG*) that foster measurable learning without compromising the engaging qualities of serious computer games are described. An ISG employs a novel, multiphase approach to adapt, on-the-fly, both the game content and the data analyses software based on observed student behaviours to improve the learning experience perceived by the students. An ISG called Virtual Bridge Constructor where a student makes decisions on behalf of a construction supervisor to build a single span bridge was developed and tested with the help of several high school and undergraduate engineering students. Analyses of the test data validate our central hypothesis that the performance data guided co-evolution of gaming software and data analyses in an ISG improves student performance and their perceived knowledge gain.

Key Words

Intelligent tutoring systems, adaptation, data mining, game-based simulations

1. Introduction

An important requirement of serious computer games [1]–[5] is to provide adaptive and assessment support similar to a traditional human-to-human teaching environment. In an ideal teaching environment, a teacher continually assesses where a student is at and where she/he should be at the end of each tutoring session and adapts the lesson plans and evaluation methods accordingly. Designing serious computer games that similarly adapt the training scenarios to improve learning without compromising engaging game qualities can be highly beneficial for learning aided by current computational advances.

While the importance of adaptation in serious games has been well recognized [2], [3], [6]–[10], current serious games provide limited support for adaptations in terms of the underlying models as well as their presentations. Usually, adaptations can be performed only at the beginning of a game session or at predetermined points that are chosen arbitrarily with manual help. Furthermore, the type of adaptation that goes into effect at a particular point in the session is hardwired to the types of observed user responses. Such a deterministic mapping between the responses and adaptations totally ignores the multiple causes (and the associated adaptations) that may be responsible for the observed responses. Moreover, the adaptation presentations are restricted to make changes within a single game scenario and do not allow arbitrary navigation among different game scenarios. These limitations severely hamper the benefits that can be obtained by incorporating adaptations into serious games.

We propose a novel framework called intelligent serious games (*ISG*), using a phased, control-theoretic adaptive framework where training and educational data analyses [6], [11] are both continually adapted based on observed student behaviour. In an ISG, the data analyses results are used to adapt the game content and the presentation of the scenarios on-the-fly, as well as to validate the predictions. Thus, the ISGs are significantly different from earlier game adaptation efforts since they allow various adaptations to be automatically tested within the game itself to determine their efficacy. The ISGs are also different from intelligent tutoring and other adaptive learning systems in terms of the adaptation presentations since they allow dynamic adaptation across scenarios by arbitrarily changing simulation time and state [12].

In this paper, we describe an ISG called the Virtual Bridge Constructor (*VBC*), where decisions are made by students on behalf of a construction supervisor to build a single span bridge. The VBC was used by two groups of 40 students whose performance in a play session was measured in terms of the cost and the schedule for building the bridge. The VBC analysed the collected data to identify student learning patterns and hypothesize the likely causations for suboptimal performance. Based on subsequent ranking of the hypotheses, the game content and the data analyses were adapted on-the-fly so that student performance is measurably improved. The experimental results validated our central hypothesis that VBC can be

* Computer Science, University of Nebraska, Omaha, NE; e-mail: msubramaniam, pchundi@unomaha.edu

** Construction Engineering, University of Nebraska, Lincoln, NE; e-mail: jgoedert@unl.edu

Recommended by Dr. Rita Kuo

(DOI: 10.2316/Journal.209.2015.3.209-0036)

adapted on-the-fly based on student behaviour to improve their performance and their perception of knowledge gain for building a single span bridge.

2. Related Work

Adaptation and assessment play a key role in intelligent hypermedia education systems where usually courses are customized based on user preferences, learning styles, skills, goals, *etc.*, to maximize learning outcomes. Adaptation presentations in these systems typically involve the manipulation of web-based education components such as selective inclusion and deletion of links, ordering of course elements, modification of media formats, presentation modes, navigation paths, *etc.* Adaptations are performed usually with guidance obtained from either explicit user inputs or based on computational analyses of user responses and browsing behaviours. Please see [13] for an excellent survey. While assessments have typically been local to the intelligent hypermedia education systems, a framework to integrate them with learning management systems to leverage the assessment capabilities of the latter is discussed in [13].

Adaptation and assessment methods have also been incorporated into games [2], [3], [6]–[10][1] to enhance their education potential. Games such as [2], [3], [6], [7], [9], [10] support very limited adaptation models that allow players to configure each play session in terms of their roles, skills, and objectives. The available scoring mechanisms are used for assessment in these games. A more elaborate adaptation model described by Moreno-Ger *et al.* [8] based on a finite-state machine-based gaming model allows adaptations to be performed by checking and modifying certain states and transitions identified by an instructor. Whilee assessment mechanisms are typically hardwired into games, integrating games as learning objects into learning management systems to improve the assessment mechanisms in games has also been considered in [8].

The VBC is somewhat similar in spirt to that of [8] in that it attempts to incorporate adaptivity into a civil engineering game to enhance its pedagogical value. However, VBC game engines are based on the User Interface/Control object/Model (UCM) [14], [15] architecture, which supports an infinite number of states in comparison to the finite state machines used in [8], which is crucial to handle event-based and parameterized scenarios common in serious games [1], [7], [9], [10], [12]. Furthermore, the adaptation model in VBC is much more general than that supported in all of the above serious games since it automatically discovers adaptations and also the points in the game session where these should be applied.

The adaptation model of VBC, based on a three-phase control, is much more flexible and dynamic than that used in earlier hypermedia education systems and games. Every VBC game session is an infinite series of adaptations one after another in the steady state. The use of such a control-based approach enables the timing of the adaptation to be fully dynamic in a game session. Each adaptation is viewed as a prescription that leads to the beginning of the next adaptation cycle, when the prescription wears off. While dynamic adaptivity is supported in hypermedia education systems by allowing user attributes to be periodically updated, however, these adaptations are not truly dynamic since the schedule of these updates is determined statically in an arbitrary manner [13]. Furthermore, in the VBC adaptivity model, the mapping from user attributes to adaptations is not hardwired. In fact, several candidate mappings are formulated as hypotheses and tried. Thus, the mapping from update to adaptation presentation itself is adaptable in VBC.

Most importantly, VBC takes adaptation one step even further, by providing adaptivity in the choice of the functions used to analyse user responses in a particular adaptation phase of the game session. The VBC adaptation of model is similar to [16] in terms of supporting a community-based adaptation by automatic detecting user subgroups by using clustering methods. Adaptation presentation in the VBC not only requires creation, deletion, and hiding of new multimedia elements including audios, videos, and avatars similar to those in the hypermedia courses, but it also involves careful rollback of parameterized simulation states to formulate incremental solution plans to make progress in the game.

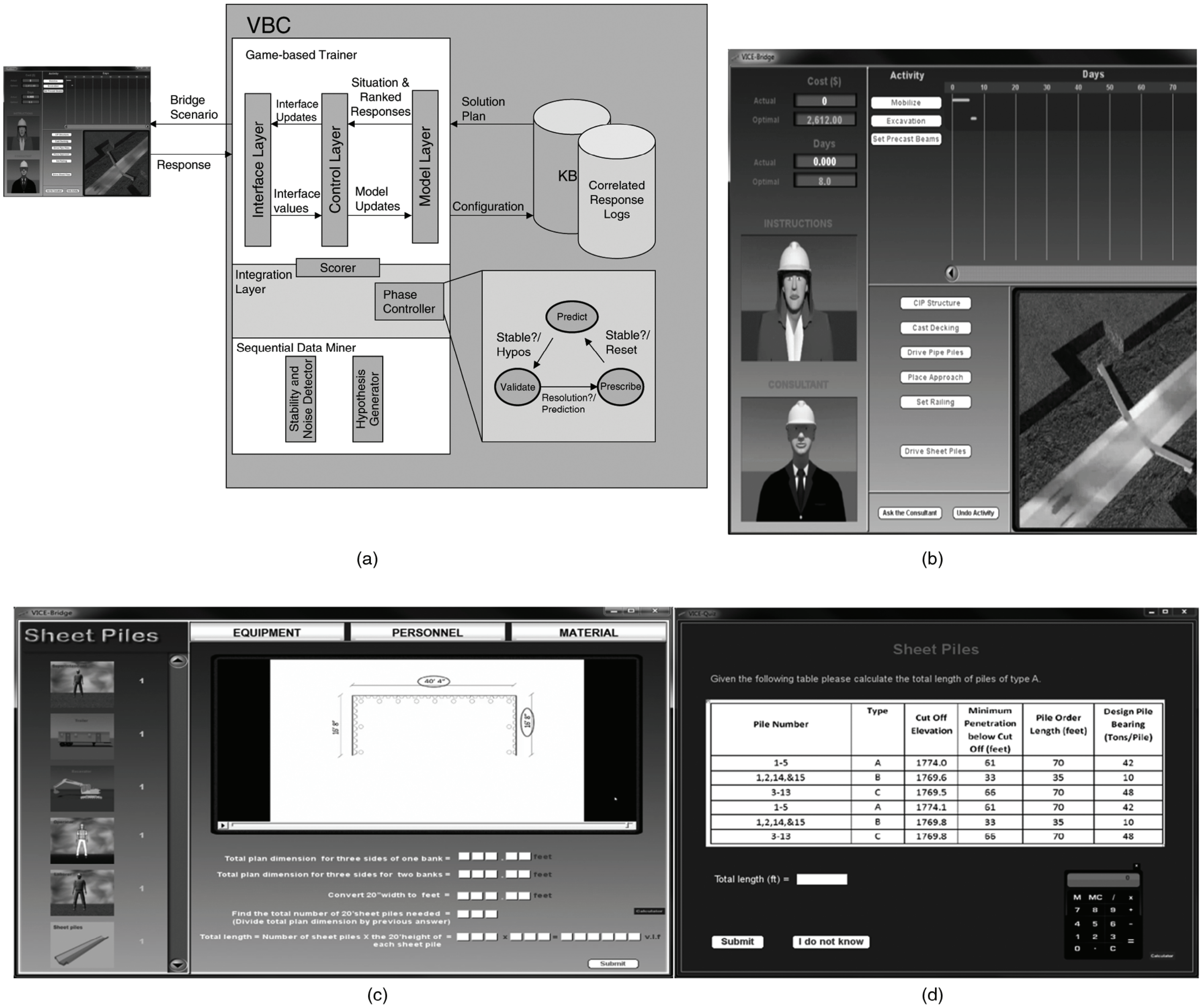
3. Virtual Bridge Constructor Architecture and Bridge Game

The VBC system architecture Fig. 1 has a modular structure with four basic components namely: (a) game-based trainer, (b) sequence data miner, (c) integration layer, and (d) knowledge bases (KBs).

3.1 Trainer

The UCM [14], [15] , commonly used to design game-based trainers, consists of three layers – *i*) an *interface layer* to control elements such as buttons, dials, gauges, equipment, and movie clips, *etc.*; *ii*) a *control layer* that includes mechanisms for coordinating interface elements and mediates the user interface with the simulation model; and *iii*) a *model layer* that includes computations and processes required to realize the functionality in the simulation model. User responses are captured by the interface layer including the values of the different interface elements (and their states) and communicated to the control layer, which interprets the responses based on the state of the interface elements and generates the appropriate actions for the simulation model. The model layer incorporates these actions into the current model configuration in the available solution plan and moves the model to the next configuration. The actions are used by the *scorer module* to generate a score for the user response. The score and the scenario corresponding to the new model configuration are sent to the control layer, which generates interface updates to display the next scenario to the user. The process is repeated on receiving the next user response.

[1] More details are in http://www.educationarcade.org/



Total plan dimension for three sides of one bank = □□□.□□ feet

Total plan dimension for three sides for two banks = □□□.□□ feet

Convert 20"width to feet = □□□.□□ feet

Find the total number of 20'sheet piles needed = □□□
(Divide total plan dimension by previous answer)

Total length = Number of sheet piles X the 20'height of each sheet pile = □□□ x □□□ = □□□□□□ v.l.f

Given the following table please calculate the total length of piles of type A.

Pile Number	Type	Cut Off Elevation	Minimum Penetration below Cut Off (feet)	Pile Order Length (feet)	Design Pile Bearing (Tons/Pile)
1-5	A	1774.0	61	70	42
1,2,14,&15	B	1769.6	33	35	10
3-13	C	1769.5	66	70	48
1-5	A	1774.1	61	70	42
1,2,14,&15	B	1769.8	33	35	10
3-13	C	1769.8	66	70	48

Total length (ft) =

Figure 1. (a) Architecture; (b) WBS failure scenario; (c) education module for sheet piles; and (d) sheet pile pre-survey.

3.2 Miner

It consists of a collection of processes and computations to analyse user responses and correlate them to the knowledge components. It comprises of two modules, namely, *i*) *data, noise, and stability analyser* and *ii*) a hypotheses generator. The first module has a library of customizable routines to measure score of individual response and discernibility of a sequence of responses, cluster responses, *etc*. The hypothesis generator module contains routines that can formulate a set of hypotheses based on user responses. It also has routines that can validate a set of hypotheses against a sequence of user responses and check the conformance of user responses against a set of validated hypotheses.

3.3 Integration Layer

It consists of user response KB and the *phase controller*. KB records user responses to each scenario along with the associated knowledge components used to generate the scenario and the score and other related function values computed by the miner. Accesses to the user response KB are performed by the miner and are managed by the integration layer.

The phase controller divides the operation of the VBC into three phases, namely, *prediction*, *validation*, and *prescription* phases. VBC starts in prediction phase where scenarios are generated by the trainer based and scored without any interaction with the miner. The miner analyses each user response until it detects a stable and discernible sequence of responses. Upon this, the miner formulates hypotheses predicting future scenarios that are most likely to improve learning, which are communicated to the trainer, transitioning VBC into validation phase. The hypotheses affect adaptations to both the simulation model and the interface layer. First, in the model layer, the knowledge components referred to in the hypotheses are identified. User responses in the solution plans that are not connected with these knowledge components are pruned so that scenarios based on these knowledge components are only generated. The pruned actions are communicated to the control layer, which then suitably adapts the interface elements. This may involve adding/removing certain simulation elements. The behaviour of the miner in the validation phase is different from that in the prediction phase. Each user response to a scenario with the concerned hypothesis is analysed to determine the trend, if any, in the cumulative benefit due to this scenario. Hypotheses showing significant promise for improving learning objectives are considered validated and are ranked. Upon determining a validated set of hypotheses, the phase controller transitions the VBC to the prescription phase. In the prescription phase, the trainer is adapted to generate scenarios based on the top ranked hypotheses. VBC transitions to the prediction phase upon reaching a stable sequence of user responses. A stable sequence of user responses in the prescription phase indicates that the prescription has produced the desired effect or that it has worn off. In either case, VBC enters the next adaptation cycle beginning with the prediction phase.

3.4 Single Span Bridge Game

In VBC, a player has to make resource decisions to build a single span bridge with optimal cost and time. The game consists of eight modules developed using animations, voices, texts, and educational modules. Each play session requires a student to plan and complete all these modules one after another. The first activity at the start of the game is to organize the eight modules of the project correctly into a work breakdown sequence (*WBS*). Three avatars provide guidance in completing this activity and others. A *prompter* avatar directs the student to the next component in the sequence. On a successful choice, an animated video shows the progress made in constructing the bridge. The consequences of a failed choice are also visually shown. For instance, choosing beams component (*Beams*) instead of the sheet piles component (*Spiles*) results in beams crashing into the river as depicted in Fig. 1(b). A *consultant* avatar provides solutions at a cost throughout this activity and others. A *superintendent* avatar arrives at the location of failure, mandates the player to ask the consultant to understand a failed choice such as why the beam placement failed, and choose the correct alternative. The number of *consultant hits* measures a player's performance in a session. Higher performing sessions have smaller number of hits. Currently, each session is initialized so that the consultant avatar is invoked as soon as a student commits an error.

The subsequent activities in a session involve completing each of the eight modules of the WBS in order. The first module, *Mobilize*, serves as a tutorial for the others. The student is directed through a sequence of decisions that are typical of all modules. The tutorial ends with the animated delivery of a jobsite trailer and the construction of temporary road across the river.

To complete the remaining modules, a student must make their own resource decisions involving equipment, personnel, and material selections. These modules also include one or more *education modules*, where students receive instruction about relevant concepts through videos and are then tested to assess their ability to apply acquired knowledge instead of rote memory recalls. An education module testing the amount sheet piles required depicted in Fig. 1(c). One of the WBS modules, *cast in place superstructure* (CIP), consists of 5 sub-modules. These sub-modules must also be sequenced correctly and then completed individually. Suboptimal decisions in any activity of a play session results in *cost and schedule penalties*, which are used in scoring a student's performance. The prompter, consultant, and the superintendent avatars are available throughout a game to provide guidance.

4. Virtual Bridge Constructor Data Analyses

The VBC and SQLite database version 3.0 were installed on IBM Windows 7 desktops having 4GB RAM. VBC runs in the client–server mode with the trainer located in the clients. Forty high school juniors from across the state, attending an academy for students inclined towards engineering and/or information sciences, were selected randomly on a first come first serve basis to play VBC. The group consisted of 60% male and 40% female students.

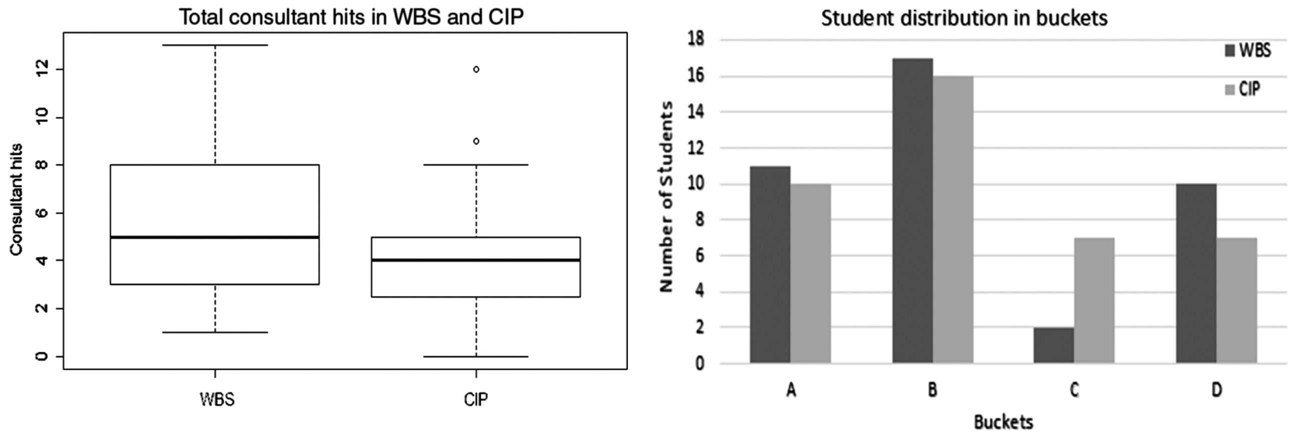
After completing a demographic and skill questionnaire, each participant went through three main phases: a pre-quiz, the play session, and a post-survey, and data were collected in all three phases. The pre-quiz assessed the construction engineering knowledge and familiarity through 25 interactive simulation questions. For example, a pre-quiz question for finding (using calculator) the amount of sheet piles of certain type needed to build a bridge can be seen in Fig. 1(d). The data collected in the pre-quiz consisted of 25 tables, with following common schema.

("Uname", "Answer", "Attempts", "Start_time", "End_time", "Duration", "Click_donotknow")

Uname is the unique identifier given to each participant. *Answer* stores the answer given by the participant and the *Attempts* records the number of attempts used by the participant before getting the correct answer where each question was allowed 3 attempts. *Start_ time* and *End_ time* record the start and end times for each question, respectively. The *duration* records the time taken in seconds. Some of the questions consisted of a "Do not know" button that can be used by the participant if he/she does not know the answer. This information was recorded under the field *Click_ donotknow.* Using the information from the login and sign-up stage, the username and the demographic data (personal details, education details, and work experience data) were also recorded in KB.

The pre-quiz scores were used to adapt the play session by setting the expected time duration for each activity in each module, the level of detail in education modules, as well as to weigh the penalty for wrong choices. Novice players were penalized lesser than those professing higher skills. Then, each participant started the game trying to organize the tasks in the WBS activity.

KB recorded the data in the play session using 21 tables. During the WBS and CIP activities, start time



(a) (b)

Figure 2. (a) Consultant hits for WBS and CIP and (b) WBS consultant usage groups.

when the activity was begun, end time when the student completed the activity, and the consultant hits were recorded. During the education modules, start time, end time, and number of wrong attempts for each task were recorded. Finally, the total cost and the total number of days were recorded for each player at the end of a play session. We also recorded every mouse click in a play session. Here is an example of a portion of the data recorded in WBS activity; the data shows that the player first selected beams (which was a wrong choice), was forced to select consultant support, and then selected *excavation* and *pipepiles*, respectively. The system time and the system millisecond count for each click are also recorded as shown.

```
“Beams∼8:43 AM∼311425004; Consultant∼8:43 AM
∼311433733; Excavation∼8:43 AM∼311445624;
Pipepiles”
```

An example of the part of the data recorded for the next play session stage where each module in the Data after WBS completion is given below; the data shows that student first selected the equipment crane, an excavator, an operator, two laborers, pipe-piles, and then finally removed the excavator.

```
“clicked∼Equipment_Crane∼8:47 AM∼311681036;
clicked∼Equipment_excavator∼8:49 AM∼311793150;
clicked∼Personnel_Operator_WF∼8:50 AM∼311841517;
clicked∼Personnel_Laborer_HM∼8:50 AM∼311842674;
clicked∼Personnel_Laborer_HM∼8:50 AM∼311843622;
clicked∼Material_Pipepile∼8:51 AM∼311911164;
Removed∼Equipment_excavator∼9:02 AM∼312571351;”
```

In the education modules, answers of each attempt are recorded along with the corresponding times. A small sample of data collected for the module in Fig. 1(c) is given below; the data show that the student entered (or calculated) 73.65, 147.30, . . . , 1760.00 as values for the input boxes in that sub-module. These values are recorded when the student clicks the *submit* button.

```
Spiles∼73.65|147.30|1.67|88.00|20.00|88.00|1760.00
∼9:43AM∼314993042
```

4.1 Results

To illustrate the effectiveness of our three-phased adaptation approach, we focussed on the correct sequencing of tasks in the WBS and CIP activities, which are crucial to obtain a good overall understanding of single span bridge construction. However, we should emphasize that VBC automatically discovered adaptations in other modules and these were equally effective as well.

4.1.1 Consultant Hit Metric

The performance of the players in these two activities is calculated as a function of duration, hints, and education module accesses that the players take to place each task in its correct position. If the function outputs a value that is more than a threshold that is configured based on the player skill level, then the player has used too many resources despite correctly completing the task. This suboptimal performance manifests on the interface as a *consultant hit.* A wrong placement, of course, results in a consultant hit. The aggregate number of consultant hits in each activity and the number of consultant hits per task per activity measured player performance in the WBS and CIP activities.

4.2 Aggregate Hits without Adaptation

The aggregate number of consultant hits in the WBS and CIP activities was measured without any adaptation to analyse the performance of students in these activities, in general.

The number of consultant hits for the players for the WBS activity is depicted in Fig. 2(a). The box plots show that the number of consultant hits in the WBS was in the range 1–13 with median number of hits around 5. The number of consultant hits for the CIP was in the range 1–8 with the median number of hits around 4. Two outliers in the CIP had about 9 and 12 consultant hits. The plots also show that there was more consultant usage in the WBS in comparison to that in the CIP, which could be attributed to the larger number of component tasks to be sequenced in the WBS. The data logged in the database from the play sessions had several back-to-back consultant hits without any component task selection for sequencing. We believe that this may be due to interface characteristics, particularly relating to the audio quality. These hits were counted as a single hit while preprocessing the logged data for analysis. Based on the WBS performance, the players were classified into four buckets – A (0–3 hits), B (3–5 hits), C (5–8 hits), and D (higher than 8 hits). The ranges for the CIP buckets were A (0–2 hits), B (3–4), C (5–8), and D (higher than 8 hits). The number of players in each bucket for the WBS and CIP activities is depicted in Fig. 2(b). The number of players in a bucket for the CIP closely tracked those in the corresponding one for the WBS in all cases except bucket C. The number of players in bucket C for the CIP was higher than that in the WBS despite the latter having more tasks to be sequenced.

We also studied the relative performance across these two activities both of which assessed a player's ability to integrate several component tasks in the proper order by analysing the percentage of players who transitioned from a WBS bucket to the various CIP buckets. The performance of players in bucket A with close to 0 consultant hits was given the *good* rating, that of players in buckets B and C (combined) with around median number of consultant hits was given the *meeting expectations* rating, and those in bucket D was given the *needing improvement* rating. Our results about the player transitions, depicted in Fig. 4(a), showed that around 80% of players with *needing improvement* rating in WBS received the *good* or *meeting expectations* rating in CIP. Majority of players with *good* or *meeting expectations* rating in the WBS were similarly rated in the CIP. We attribute the improvement in the sequencing of WBS to that of CIP to players getting familiar with the interface since CIP sequencing follows that of WBS during a play session. Surprisingly, no player retained their good rating from WBS to CIP. This may be attributed to the fact that the videos of WBS show bridge construction at a higher level and perhaps aided sequencing more than the detailed videos in the CIP.

4.3 Three-phased Dynamic Adaptation

We describe the working of the three-phased dynamic adaptation in the VBC in detail for the WBS sequencing activity. The adaptation was similarly carried out to improve player performance for CIP activity as well.

4.3.1 Prediction Phase

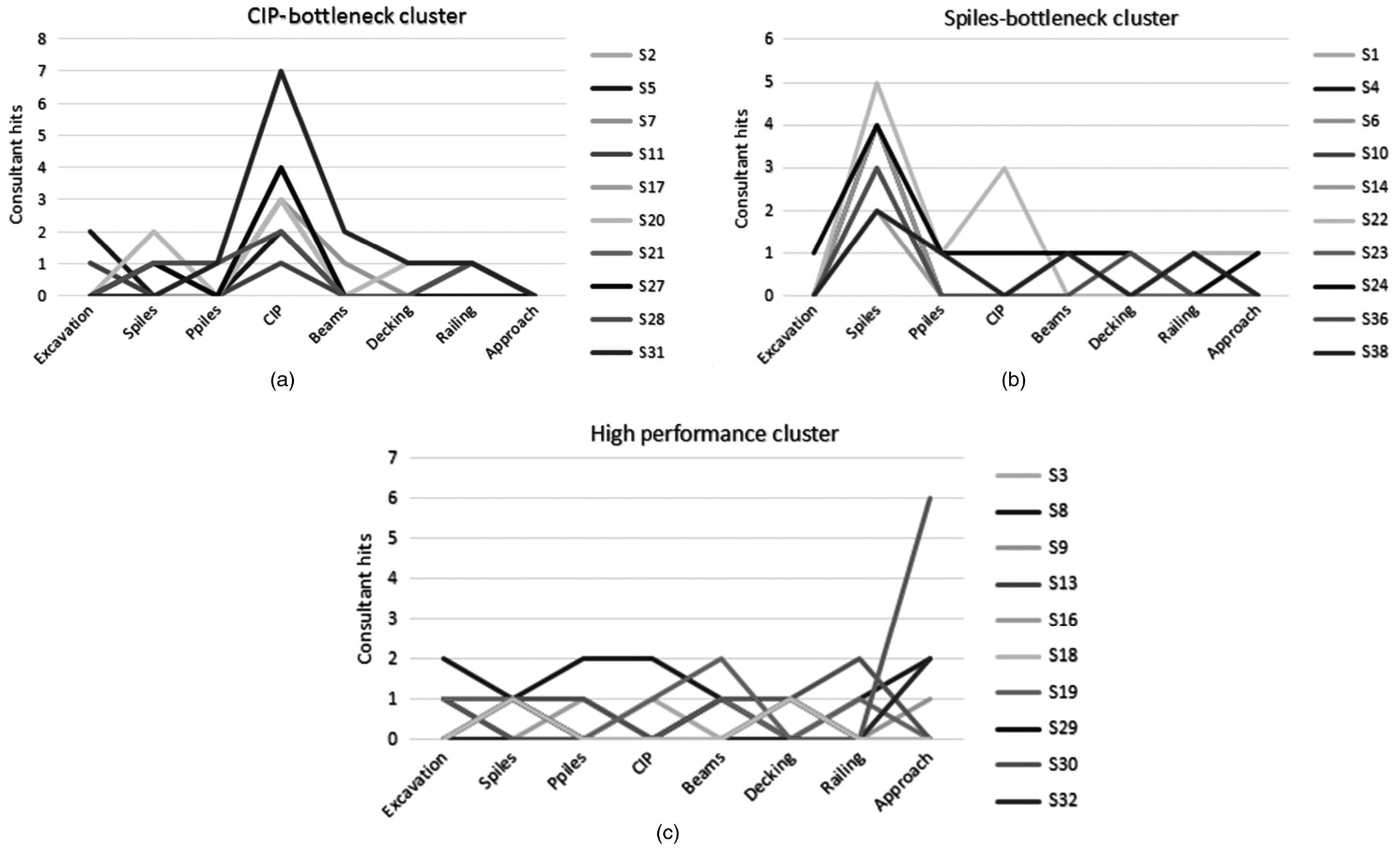
The players encountered the WBS sequencing activity, for the first time, in the prediction phase of the VBC. In this phase, the control layer in the UCM recorded the current WBS task that was being sequenced by the player and the number of times the consultant avatar was invoked due to sub-optimal performance detected by the simulation model based on the state variables and stored in database. All VBC clients were made to wait via a barrier until every player has completed the WBS. The consultant hit data were clustered by the miner to produce the consultant hit clusters to identify potential patterns of consultant usage in the WBS sequencing. The K-means clustering algorithm [17] was applied to obtain seven hit clusters. There were three clusters with 10 or more elements that are displayed in Fig. 3(a)–(c). The remaining four clusters each had less than five elements and were deemed insignificant for further analysis. The cluster in Fig. 2(a) consists of 10 players with consultants hits in the range 0–7. Sequencing the CIP module was the main bottleneck since the peak number of consultant hits occurred in this task for all the students in this cluster. The cluster in Fig. 2(b) is similar except that the bottleneck is the *Spiles* module. The cluster in Fig. 2(c) with 13 students was the largest among all the clusters. This was a high performance cluster having low number of hits in the range 0–2 (except for one outlier).

4.3.2 Validation Phase

The validation phase resulted in different adaptations at each (group) of the clients based on the cluster to which they belong. Therefore, each player(s) can potentially be playing a customized version of the game at this phase. Players belonging to the third low-hit cluster were deemed to have successfully completed WBS and moved to the remaining game modules. For the players in the first two clusters, a set of hypotheses explaining their sub-optimal performance were formulated by miner. The adaptation presentation forced these players to repeat WBS to validate and rank the hypotheses.

As an example, hypotheses generated to explain bad performance in the CIP-bottleneck cluster were as follows: *Hypothesis 1*: poor VBC user interface, *Hypothesis 2*: not knowing where to place CIP task in the sequence in comparison to others, or *Hypothesis 3*: inadequate construction knowledge about CIP task in comparison to the other WBS tasks. We derived these hypotheses using predefined first-order logic error models [1] in the KB for sequencing activities and adapted them for our scenario.

To validate *Hypothesis 1*, the adaptation did not make any interface changes, but it simply updated the miner to compare the number of hits incurred in placing CIP task to that in placing the other WBS tasks. The subsequent data analyses determined that VBC interface was not a likely cause of poor performance since several players who had a high number of hits in placing the CIP task had incurred a low number of hits in placing the other tasks. Consequently, *Hypothesis 1* was ruled out. To validate *Hypothesis 2*, the miner was adapted so that its routines analysed the "sharpness of the peaks" of the hits for the CIP. There was no change to the interface with the game being in the wait mode. The miner measured the difference in the number hits for the CIP task with that for tasks immediately preceding and following the CIP. It was



(a)

(b)

(c)

Figure 3. Clusters with (a) CIP-bottleneck; (b) Spiles-bottleneck; and (c) Low hit usage.

inferred that the students knew where to place the tasks immediately preceding and following CIP but were unsure of the place of CIP in the sequence. Hence *Hypothesis 2* was a deemed a likely cause.

To validate *Hypothesis 3*, the game sequence was altered to force players to complete the CIP module to evaluate their knowledge of the CIP. Only 6 out of the 10 students in the CIP-bottleneck cluster completed the CIP module with a high number of consultant hits. Consequently, *Hypothesis 3*, was a likely cause for the poor performance in WBS but it was accorded a lesser rank than *Hypothesis 2* since the validation threshold for hypothesis 3 was set to 0.75, *i.e.*, at least 8 out of 10 students must encounter a high number of consultant hits in completing the CIP.

Note that in any normal bridge construction, mobilization of resources and excavation must precede CIP. Such a significant change was feasible only because of the loose coupling of the VBC game modules to support adaptation. The adaptation also changed the miner to collect data about the performance of these players in the CIP module, in terms of their performance in sequencing CIP tasks and perform cross-validation with a threshold value of 0.75.

Note that if multiple hypotheses require adaptation involving additional player interactions, VBC currently sequentially validates hypotheses one at a time. It is also possible to use different players to validate different hypotheses. This could make it more efficient to validate hypotheses but would need sufficient number of players for meaningful validation.

4.3.3 Prescription Phase

As before, the players were made to wait until all of them completed the CIP sequencing and the hypotheses were validated and ranked. In the prescription phase, the game was rolled back to the WBS sequencing activity, whose interface had been adapted with additional scaffolding to aid the players in placing the CIP task. The consultant avatar was made to appear proactively in the WBS sequencing when it was time to place the CIP. The associated audio files were dynamically changed to produce a detailed explanation about the place of CIP in comparison to the other tasks. Note that these files were always available at VBC whenever the consultant hits occurred at CIP task placement. Their order of appearance was only changed for the adaptation presentation. The adaptation also changed the scoring to avoid penalties associated with CIP task placements.

Note that providing additional scaffolding to improve a learning activity is common in intelligent tutoring systems [18]. However, VBC was able to perform selective scaffolding due to hypothesis ranking. For instance, it ignored other scaffolding choices such as relocating an education module, instructional videos related to the CIP modules as they were related to the overall knowledge of CIP due to the low ranking of *Hypothesis 3*. The consultant help could be provided either from the very beginning for CIP or after some number of unsuccessful attempts. The VBC prescription provided consultant after one attempt to retain the flexibility to learn from mistakes.

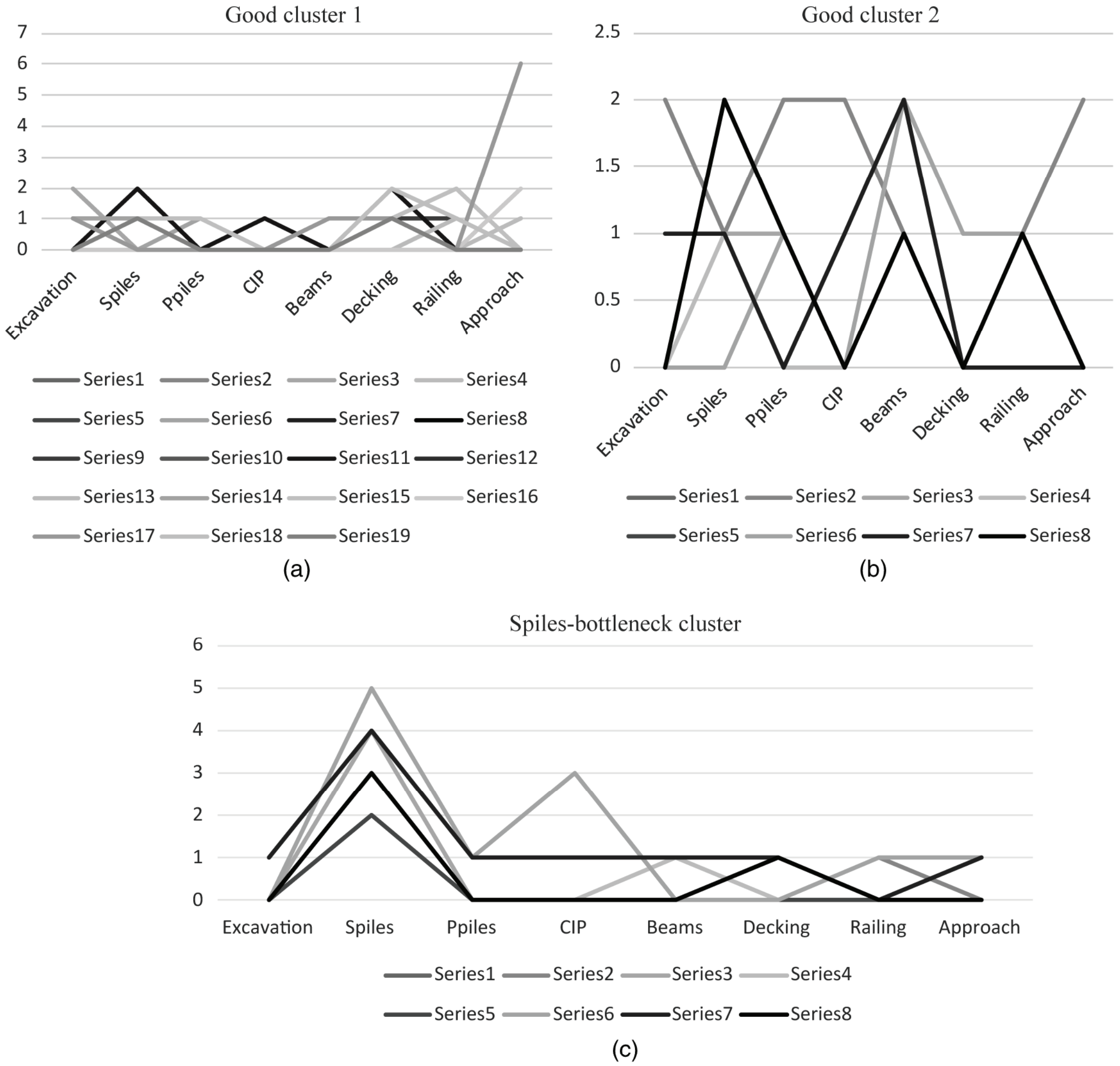


Figure 4. CIP-bottleneck prescription effects on performance.

Q3 Rate how each impacted your performance on simulation today
a) Instructions in the simulation
b) Ask a consultant feature
c) Instructional videos
d) Learning from my mistakes

No help:-2, A little help:-1,
Some help:0, Much help:1,
Excellent help:2

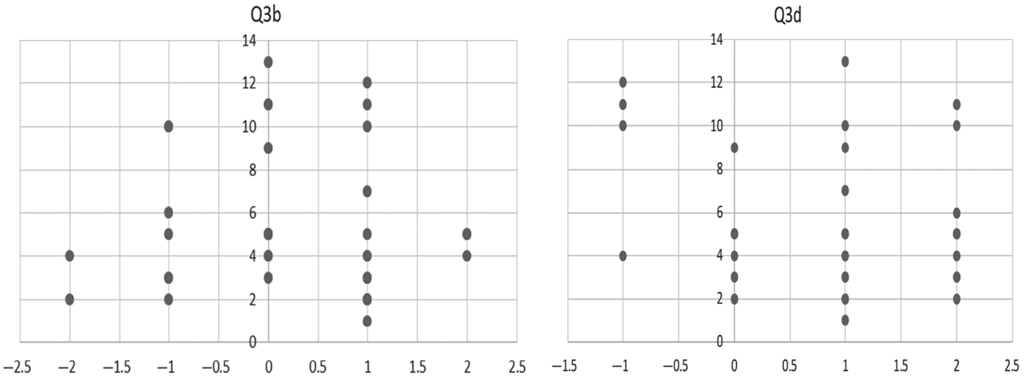


Figure 5. VBC features and perceived performance.

The effect of the prescription on improving learning for the CIP-bottleneck cluster was measured by repeating WBS sequencing for the members of the cluster for the second time. Their performance was recorded as before. This performance data replaced their older performance in the original data about all students and the updated data were clustered again. Three of the four resulting clusterings are depicted in Fig. 4. It can be seen from the figure that around 80% of the students in the CIP-bottleneck cluster have moved to one of the two good clusters. Cluster 3 (original *Spiles* cluster) is unaffected. A new cluster, not depicted in the figure, had bottlenecks in beams and decking module placements and VBC enters the next adaptation cycle to fix these bottlenecks.

4.4 Features Aiding Perceived Learning

In the post-survey, around 90% of students indicated that a play session with VBC helped them gain knowledge about construction engineering that they did not possess earlier. We then analysed the responses to question about the features that positively impacted their learning experience. The relevant question and results are depicted in Fig. 5. The results showed that around 83% of the students indicated that the *Ask a consultant feature* positively impacted their performance and around 90% of the students liked the *flexibility to correct their mistakes*. The instructional videos were also perceived to positively impact their performance though to a lesser extent.

5. Conclusion

A novel framework towards ISG that seamlessly integrate serious games with data analyses methods to customize training so that learning is measurably improved as perceived by students is described. In ISG, training and data analysis functions are both dynamically adapted based on the behaviour of students. The data analyses results are used to dynamically adapt the game content and the presentation of the scenarios while the trainer is used to validate mining predictions on-the-fly. The ISG framework was implemented in a serious computer game VBC where a student makes decisions on behalf of a construction supervisor to build a single span bridge. Two groups of 40 students played VBC and the performance of students during the game was continuously measured and automatically analysed to identify student learning patterns and hypothesize the likely causations. Based on subsequent validations and ranking of the hypotheses, performed during the play session itself, the game content and the data analyses were adapted so that student performance is measurably improved by using the features that were perceived by them to aid their learning. The experimental results validate our central hypothesis that serious games such as VBC can be adapted on-the-fly based on student behaviour to measurably improve their performance and learning.

Acknowledgement

The authors thank Aruna Weerakoon for implementing the VICE platform.

References

[1] A. Derryberry, Serious games: Online games for learning, *Adobe Whitepaper*, 2007.

[2] J. Goedert, Y. Cho, M. Subramaniam, H. Guo, and L. Xiao, A framework for virtual interactive construction education (VICE), *Automation in Construction*, *20*(1), 2011, 76–87.

[3] J. Goedert, R. Pawloski, S. Rokooeisadabad, and M. Subramaniam, A project oriented pedagogical model for construction engineering education using cyber-infrastructure tools, *Journal of Professional Issues in Engineering Education and Practice*, *139*(4), 2013, 301–309.

[4] G.J. Hwang, H.Y. Sung, C.M. Hung, I. Huang, and C.C. Tsai, Development of a personalized educational computer game based on students' learning styles, *Educational Technology Research and Development*, *60*(4), 2012, 623–638.

[5] C. Steinkuehler, Massively Multiplayer online games as an educational technology: An outline for research, *Educational Technology*, *48*(1), 2008, 10–21.

[6] R.S. Baker and K. Yacef, The state of educational data mining in 2009: A review and future visions, *Journal of Educational Data Mining*, *1*(1), 2009.

[7] K. Koidl, F. Mehm, C. Hampson, O. Conlan, and S. Göbel, Dynamically adjusting digital educational games towards learning objectives, *4th European Conf. on Games Based Learning*, 2010, 175–180.

[8] P. Moreno-Ger, D. Burgos, I. Martínez-Ortiz, J.L. Sierra, and B. Fernández-Manjón, Educational game design for online education, *Computers in Human Behavior*, *24*(6), 2008, 2530–2540.

[9] M. Prensky, Digital game based learning (New York, NY: McGraw-Hill, 2001).

[10] R. Purushotma, Commentary: You're not studying, you're just..., *Language Learning and Technology*, *9*(1), 2005, 80–96.

[11] C. Romero and S. Ventura, Educational data mining: A review of the state of the art, *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, *40*(6), 2010, 601–618.

[12] M.K. Singley, P. Fairweather, T. Wolf, and D. Lam, PASTEL: Pattern-driven adaptive simulations, in *Human-Computer Interaction-INTERACT*, in C. Baranauskas, P. Palanque, J. Abascal, and S.D.J. Barbosa (eds.), Taylor & Francis, 2007, 349–352.

[13] S. Graf, *Adaptivity in learning management systems focussing on learning styles*, Doctoral Dissertation, Vienna Institute of Technology, 2007.

[14] I. Horrocks, *Constructing user interface with statecharts* (Boston, MA, USA: Addison-Wesley, 1999).

[15] J. Kayes and D. Castillo, Flash MX for interactive simulation, (NY: Thomas Publishing Company LLC, 2003).

[16] E. Gaudasio and J.G. Boticario, Towards web-based adaptive learning communities, *Eleventh Intl. Conf. on Artificial Intelligence in Education (AI-ED)*, Sydney, Australia, 2003.

[17] J. Han, M. Kamber, and J. Pei, Data Mining Concepts and Techniques, Morgan Kaufmann Publishers, Inc., 2011.

[18] M.J. Prince and R.M. Felder, Inductive teaching and learning methods: Definitions, comparisons, and research bases, *Journal of Engineering Education*, *95*(2), 2006, 123–138.

[19] R.S. Baker, A.T. Corbett, I. Roll, and K.R. Koedinger, Developing a generalizable detector of when students game the system, *User Modeling and User-Adapted Interaction*, *18*(3), 2008, 287–314.

[20] R.S. Baker, A. Mitrović, and M. Mathews, Detecting gaming the system in constraint-based tutors, in *User Modeling, Adaptation, and Personalization*, in P. De Bra, A. Kobsa, and D. Chin (eds.), Springer Berlin Heidelberg, 2010, 267–278.

[21] S. Cetintas, L. Si, Y.P. Xin, and C. Hord, Automatic detection of off-task behaviors in intelligent tutoring systems with machine learning techniques, *Learning Technologies, IEEE Transactions on*, *3*(3), 2010, 228–236.

[22] J.S. Kinnebrew, G. Biswas, and B. Sulcer, Modeling and measuring self-regulated learning in teachable agent environments, *Journal of e-Learning and Knowledge Society*, *7*(2), 2011, 19–35.

Biographies

Mahadevan Subramaniam is an associate professor of computer science at the University of Nebraska-Omaha, He has a B.E. in computer science from the Birla Institute of Technology in India and a Ph.D. in computer science from SUNY at Albany. His research interests are formal methods, modelling and simulation, and testing of hardware/software systems. He has significant industrial experience in architecting high-end hardware and system designs. He has published more than 70 conference and journal papers in leading IEEE, ACM conferences, and international journals. He is the director of the Modeling and Simulation Lab at the Peter Kiewit Institute in the University of Nebraska-Omaha.

Parvathi Chundi is an associate professor in the computer science department at the University of Nebraska-Omaha. She has a B.E. in computer science from Anna University, India, and a Ph.D. in computer science from SUNY at Albany. She has extensive research experience in the areas of information retrieval, data mining, and databases. She is the director of the Big Data Lab, holds five US patents, and has published more than 50 research papers in journals such as the Springer's Data Mining and Knowledge Discovery Journal, and conferences such as the SDM, CIKM, *etc.* She also has several years of industrial research experience in research labs such as the HP Labs.

James D. Goedert, Ph.D., P.E., is a professor in the Durham School of Architectural Engineering and Construction in the College of Engineering at the University of Nebraska-Lincoln. He has a B.S. in Construction Engineering from the University of Nebraska-Omaha, a Masters of Business Administration from Indiana University, and a Ph.D. in the Interdisciplinary Field of Business Administration from the University of Nebraska-Lincoln. His research interests include sustainable residential design, modelling and simulation, and sustainable design in developing countries. He has several years of construction industrial experience in addition to 25 years at the university. He has published more than 40 papers in leading construction conferences and journals. He was the Chairman of Construction Engineering and Management department for 8 years and then Director of Research and Industry Outreach at the University of Nebraska-Lincoln prior to returning to the faculty to pursue his international interest.