A Large-Scale Category-Based Evaluation of A Visual Language for Adaptive Hypermedia

Javed Khan
Computer Science
Warwick University
Javed.Khan@warwick.ac.uk

Alexandra I. Cristea
Computer Science
Warwick University
A.I.Cristea@warwick.ac.uk

Ahmed Alamri
Computer Science
Warwick University
A.Alamri.1@warwick.ac.uk

ABSTRACT
Adaptive Hypermedia (AH) provides a personalised and customised approach, enhancing the usability of hypermedia, by building a model of various qualities of a user and applying this information to adapt the content and the navigation to their requirements. However, authoring adaptive materials is not a simple task, as an author may be pressed for time, or simply lack the skills needed to create new adaptive materials from scratch. The most challenging part is the authoring of the adaptation specification (adaptive behaviour rules). This paper tackles this challenge by proposing and evaluating (on a large scale) a visual language for authoring of adaptive hypermedia.

CCS Concepts

Keywords
Authoring; adaptive hypermedia; visual language; block language.

1. INTRODUCTION
Current learning management systems offer a static approach to the delivery of learning materials. This means that every learner is given the same set of learning material. Adaptive Educational Hypermedia (AEH) provides a more personalised and customised approach to the field of eLearning than the outdated static methods. AEH enhances the usability of hypermedia, by building a model of various qualities of a learner and applying this information to adapt the content and the navigation to the requirements of the learner. Adaptive strategies are sets of adaptive rules to indicate the conditions under which adaptation behaviours can be applied in AEHs [1]. The AEH approach has been shown to be useful, as it displays more relevant content, according to the information stored in various models (user, goal and presentation model).

However, a known issue in adaptive hypermedia is the authoring process. A review of the literature [2, 3, 4] revealed that the authoring process is a bottleneck in adaptive course creation and it needs to be improved, in terms of interoperability, usability and reuse of the adaptive behaviour (strategies). Adaptive Hypermedia authoring is considered to be challenging and laborious [5]. Thus, a hypermedia system should make it easy and natural for the already burdened authors to create adaptive courses easily. Within authoring of hypermedia, the creation of an adaptation specification is one of the more difficult processes in creating an adaptive hypermedia course.

Thus, this paper aims to alleviate the authoring of adaptation specifications (adaptive behaviour), by proposing and evaluating a new, block-based visual authoring language for adaptive hypermedia.

ICIEI'18, June 30 – July 2, 2018, London, UK.

DOI: http://dx.doi.org/

2. RELATED WORK
Several authoring systems for adaptive hypermedia have been proposed in the past. InterBook [6] uses a concept-based approach and is authored via Microsoft Word. The familiar environment makes authoring easier; however, adaptation specification is very limited. AHA! [7], a well-known and comprehensive adaptive hypermedia system, offers several tools for authors, e.g., the Graph Author for creating domain models; an Application Management Tool for file management; a Concept Editor and a separate Adaptation Rule Form, allowing practically complete (in terms of Brusilovsky's taxonomy [8]) adaptive hypermedia authoring. The tool set complexity can however be overwhelming for the beginning author. ACCT [9] offers the course developer tools to design, test and deploy adaptive personalised eLearning, based on pedagogical support and instructional design principles, built on 'narrative structures'. APeLS [10], a personalised eLearning service provides authors a multi-model, metadata-driven approach for producing rich adaptive eLearning solutions that remain content and domain independent, allowing many pedagogical approaches and highly flexible solutions. However, both APeLS and ACCT adaptation descriptions need to be created in XML, which can be unwieldy. ACTSim [11], which teaches interpersonal relationship skills offers authoring that separates content and adaptivity of the simulation, allowing the adaptivity to be altered and the content to be reused. However, as adaptivity is placed across the content, reusability is limited. The Grapple Authoring Tool (GAT) has three main components: a Domain Model tool, for creating a conceptual representation, a Pedagogical Relationship Type tool (PRT), for defining types of pedagogical relationships between concepts and their associated adaptation, and a Conceptual Adaptation Model (CAM) tool for defining the pedagogical structure of a course. GAT allows very general types of relationships and adaptation rules and separation of concerns, by separating domain and adaptation. Adaptation is further separated into pedagogical, reusable relationships, and the. GAT uses drag&drop interfaces and even has a different interface for beginner authors and one for advanced authors. However, similar to other adaptation strategy creating tools, GAT strategies are not reusable, as they refer to some specific content (only the

pedagogical relationships are reusable). This burden for GAT authors is reflected in evaluations of authoring the behavioural part [12]. My Online Teacher (MOT) [13], a generic authoring system, uses the LAG [5] programming language, due to its flexibility and comprehensiveness in expressing adaptation. Additionally, by using a layered approach, each component of the AHS can be defined independently of the others, permitting easy reuse and extensibility. MOT allows for the authoring of domain content (via a concept hierarchy), as well as various manipulations of this content, and separate labelling of content pieces with pedagogical information. MOT uses the PEAL (Programming Environment for Adaptation Language) system [14] as an adaptive strategy editor, in which authors can use an adaptation language (LAG [11]) to program adaptation behaviours (strategies). However, PEAL is limited as it is usable only by programming-savvy persons [13].

2.1 Visual Programming System

Visual programming languages have a unique ability to make programming a more intuitive experience. The need to have knowledge of the programming language syntax is not necessary. Thus, the programming focus can be directed towards solving the actual problem, rather than trying to recall the programming language syntax. Many graphical programming systems exist, attempting to break down the barriers to learn computer programming. OpenBlocks [15] is an open-source Java library for creating user-interface (UI) elements for blocks-based programs. It only requires users to connect puzzle-piece-like objects, called blocks, of varying shapes and colours, to build their program. The Block shapes not only look like puzzle pieces, but also are connected like puzzle pieces, by matching the blocks at their connector shapes. Blockly [16] has the ability to prevent incorrect combinations from being constructed, addressing correctness issue within programming. Each of Blockly's connection types (value inputs/outputs, next/previous statements) can be labelled with type information, so that invalid connections will refuse to connect. This provides authors with instantaneous feedback and avoids many simple mistakes, hence improving correctness of adaptive specification during the authoring phase.

3. VISUAL AUTHORING SYSTEM – VASE

Responding to issues in prior research, and opportunities in visual programming, we have created a new authoring system, called VASE, based on a visual specification, and thus visual programming, as briefly described below.

3.1 VASE, The System

VASE (Visual Adaptive Strategy Environment) was targeted towards creating adaptive specifications for non-programming savvy users. To achieve this goal, several technologies and techniques were used. The system was implemented, and a pilot experiment was conducted to validate the initial features offered by the authoring system. The interaction of VASE with external tools is shown in Figure 1, including the LAG language [5], which allows to export to multiple delivery tools, such as ADE [17] and AHA! [7].

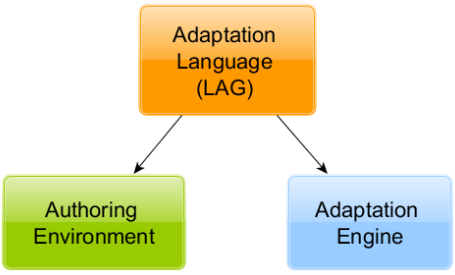

Figure 1: VASE (Authoring Environment) and its interactions

The authoring process itself is based on LAGBlocks, which further described in the next section. VASE was created to support authoring with LAGBlocks, and went through several iterations. The latest version, VASE 2.0, can run in any internet browser, including internet browsers on mobile devices. Figure 2 below shows interaction between different VASE internal components, including the adaptive authoring process's components and how the conversion of the visual representation of the adaptive strategy to the LAG programming language occurs. This process ensures interoperability with other tools (ADE, MOT, AHA!, etc.) which support LAG code.

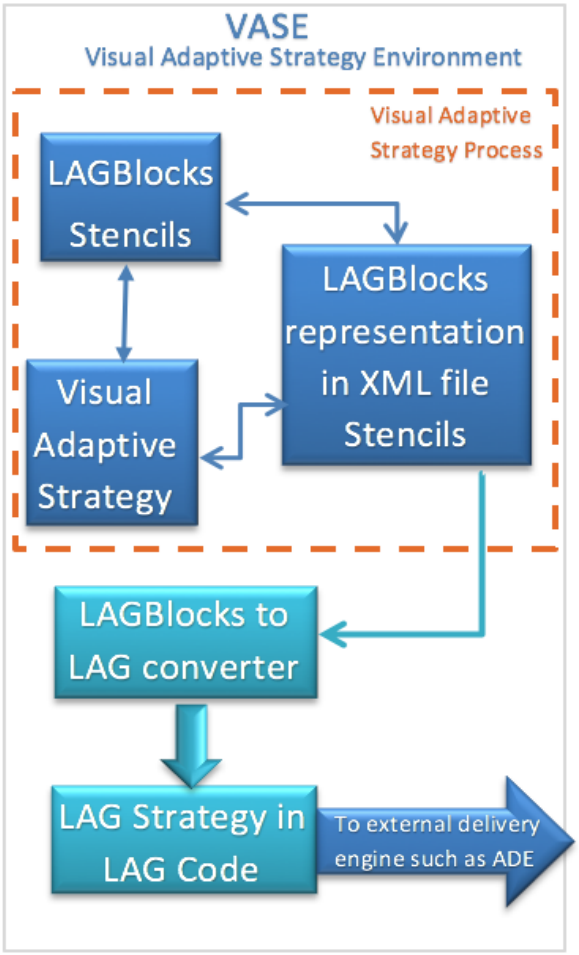

Figure 2: VASE Internal Processes

3.2 LAGBlocks: the visual language for adaptive hypermedia

Whilst the LAG language has, as mentioned, currently the widest power of expression for adaptive hypermedia authoring, visual programming languages are easier to use. Thus, we have created LAGBlocks, a visual specification, which could be used, especially, by novice authors. LAGBlocks was created on top of the LAG language, to represent any LAG statement in a visual form. The purpose of LAGBlocks was to allow Adaptive Hypermedia authors to create an adaptive strategy, without having to type any programming syntax, while still being able to export to several delivery tools, such as ADE [17] and AHA! [7]. Additionally, new adaptive specifications should be able to be created by mixing and merging existing ones. New strategies should be able to be created with ease, by reusing existing strategies. Thus, to create a visual representation for LAG, which should be user-friendly, and familiar to users, the well-known OpenBlocks [15] framework was selected as the basis of the development. This framework was further extended, to create a visual representation of the entire LAG programming language grammar, including the data types and logic constructs. LAGBlocks allows thus to manipulate LAG statements visually. Figure 3 shows the visual nature of the language; it also illustrates the drag&drop nature of the programming style.

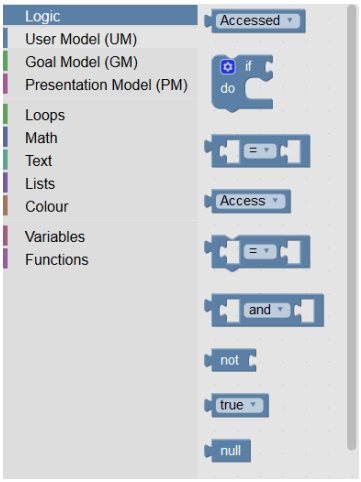

Figure 3: VASE Logic Blocks

Figure 4 shows that only grammatically correct constructs are allowed to be inserted into the slots. Additionally, suggestions for correct insertion, as well as the semantics of those insertions, are given. The figure also shows that complex usage is also allowed, although beginner users are encouraged to use only the most simple constructs.

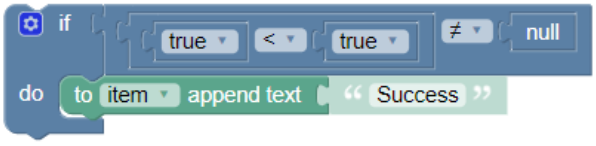

Figure 4: VASE Logic Blocks complex usage

In addition, in VASE 2.0, dynamic LAGBlock were introduced, which have the ability to change according to the user interaction (see Figure 5).

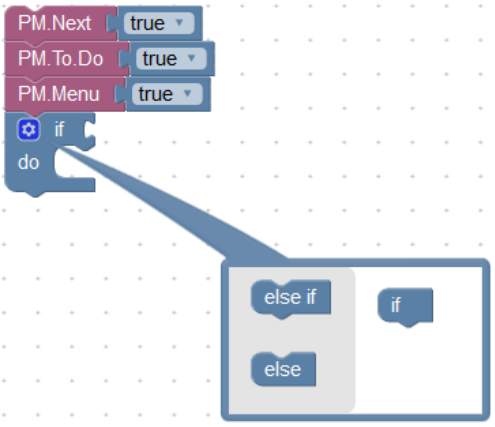

Figure 5: VASE Dynamic LAGBlocks

In general, LAGBlocks is a complete language (in the sense of Brusilovsky's taxonomy [8]), which can represent the full LAG adaptation language grammar [5]. The complete description of the language and the VASE system is beyond the scope of this paper. Instead, in the following, we report on a large-scale evaluation (with a large number of users) performed with the VASE visual authoring system, and focus on the results obtained in four specific categories of evaluation.

4. EVALUATION

VASE was tested with large numbers of users (380, very close to the ideal of 383 **as** per sample size calculation[1]) via Amazon **Mechanical** Turk (AMT[2]), which has been proven as an effective tool for conducting survey research [18].
Here, the AMT users (called 'workers') were filtered according to their reliability profile first, and then were asked to complete a set of authoring tasks on our external web server, as follows:

- to use three LAG adaptive strategies listed below, available on http://adaptive.dcs.warwick.ac.uk;
- to create two adaptive courses, based on content also available on the URL, and the strategies above.

The adaptive course contained all elements of an adaptive course. For example, the workers were able to view the contents to be used in a course and the adaptive strategy used to apply adaptive behaviour in a course. Workers could choose which authoring system to use. A mix of complex and simple adaptive strategies was used for this experiment.

Finally, workers were asked to answer a questionnaire, to assess their experience with the authoring systems. The questionnaire was focused towards uncovering the issues relating to the adaptive authoring process for comparing the two systems, VASE and PEAL (which was the most comprehensive system for authoring available in the literature, as well as was also based on the LAG language, thus allowing to compare like-for-like). The questionnaire had a total of 25 questions, which used answers on a Likert scale, from 1 to 5 (1 = strongly disagree, 2 = disagree, 3 = Neither agree nor disagree, 4 = agree and 5 = strongly agree). The questions were divided into the following groups.

[1] https://www.surveysystem.com/sscalc.htm
[2] https://www.mturk.com/

- Usability Questions: The questions focused towards guidelines for designing usability related concepts; for instance, usage in context, user interface and user-interaction.
- Usefulness Questions: These questions were aimed at measuring authoring system features in terms of correctness, completeness and other functional features. These were further grouped into three categories, user roles, benefits and quality.

To establish which significance test to use, a Pearson-Chi squared test was performed for each question for each of the two systems, VASE and PEAL. The results, with $p>0.05$, showed no normal distribution. Thus, the T-test could not be used, and the Wilcoxon signed-rank test was used instead.

Table 1 indicates that the mean for VASE was much higher than the mean for PEAL for all four categories, suggesting that users perceived VASE to be better than PEAL in terms of user usability, user roles, benefits, and quality. Please note that all these differences are statistically significant even with the Bonferroni corrections, used due to the multiple tests performed ($p'=p/4=0.05/4= 0.0125$).

Table 1. Summary of PEAL versus VASE Results for question related to each category

Category for questions	PEAL Mean	VASE Mean	Wilcoxon p<0.0125?
Usability	2.85	3.418	0.000
User roles	2.73	3.682	0.000
Benefits	2.80	3.705	0.000
Quality	2.85	3.856	0.000

5. DISCUSSION

One of the key findings in the literature review which is related to this research was reported by the LAG-XLS [19] creators, who have stated that "the major requirements for the ideal language are: reuse, flexibility, high level semantics, and ease of use". The research presented is this paper has attempted to address every aspect of these requirements. For instance, high level semantics has been created (LAGBlocks) to hide the programming syntax from the author, to lower the barrier of entry, in order to create adaptive specifications. Ease of use of the proposed visual framework has been demonstrated by allowing to create adaptive specifications visually. The reuse is addressed by using the visual representation, which can be further converted into the LAG programming language behind the scenes, able thus to be used by other adaptive tools which use LAG as the adaptation language. In addition, the tools have been found to be easy to use by the users during the evaluations conducted by this research.

A large-scale evaluation with 380 users was conducted using Amazon Mechanical Turk (AMT). The large evaluation generated results that were matched to the recommended sample size. This experiment had statistical significance for most of the questions, showing that results were conclusive within the users questioned.

This paper briefly introduces a block language for adaptive hypermedia, as well the main outcomes of its large-scale evaluation. The results are very promising, supporting the building-block approach based on visualisation. This work represents one step further towards easy access to creating personalised adaptive hypermedia, especially, personalised web information.

A visual programming framework (LAGBlocks) to represent the LAG adaptation language and an adaptive authoring system (VASE 1.0) were created, by using a block programming framework called OpenBlocks [15]. To address the limitation of VASE 1.0, VASE 2.0 was re-designed, using a different block programming framework called Blockly [16]. Two iterations of the VASE authoring system were designed and implemented using various open technologies to address the issues and the desired features highlighted by the evaluations. These requirements were implemented by using various web technologies based on W3C standards. To follow W3C standards served two purposes; firstly, this was required by the authoring imperatives [13] in the literature and secondly, it meant that any browser which implements the W3C standard will automatically be able to run the VASE 2.0 authoring system. It also meant that to run VASE 2.0 will not require any third-party plug-in to run the authoring system to create adaptive specification rules.

On one hand, the work focused towards using user roles to display relevant authoring options specific to author expectations with the authoring system, to avoid confusion and cognitive overload.

Users indicated that the VASE authoring system displayed relevant options, with a mean value of (3.9 ± 0.609) with a mode of 4, the mode indicating the most frequent answer as positive. Users preferred to choose the options they would like to see in the authoring system, with a mean value of (3.9 ± 0.611) with mode of 4. The highest mean amongst the questions was the one regarding the user roles. It means that the users preferred to be able to control the options displayed to them.

Largely, the users were satisfied with the features which were displayed to them, not by a huge majority, but they did answer it in positive, with a mean value of (3.73 ± 0.702). The results indicate that considering features related to the user roles, the users expressed positive feedback during the experiment. The value of 3.616 is believed to be positive.

On the other hand, the work aimed at the benefits of using visual programming techniques to create adaptive specifications in the terms of shallow learning curve, higher user satisfaction and less error-prone processing. The results to the questions related to the benefits of the authoring paradigm used in VASE showed that the majority of the users thought that VASE was easy to learn to use and were not confused, this would be one of the benefits of the authoring system from the user's viewpoint. Popular view amongst the users was that, VASE authoring system can be used to create adaptive strategies without much knowledge of the LAG adaptation language, with a mean value of (3.821 ± 0.712) and mode of 4. The largest mean value is generated for the questions asking if the authoring system provided a fun way to create adaptive strategies and if the authoring system provided a simple way to create adaptive strategies. Overall, the results indicate the benefits of using visual programming techniques in adaptive strategy creation. The overall benefit mean value is (3.705 ± 0.786).

6. CONCLUSIONS

In conclusion, this paper has proposed a visual language and visual learning environment for adaptive hypermedia, addressing one of the most challenging problems in the field, that of creating tools fit for use for authors. The environment and proposal has

been evaluated via a large scale evaluation, with 380 users, and this paper reports on results on four important categories of evaluation: usability, user roles, benefits and quality. For all these categories, users showed an overwhelming preference for the visual system proposed, VASE, against the arguably best competitor in the field, PEAL. This research opens up adaptive, personalized web creation for every-day users.

7. REFERENCES

[1] Stash, Natalia, Cristea, Alexandra I., and Bra, Paul De. Explicit intelligence in adaptive hypermedia: Generic adaptation languages for learning preferences and styles. In HT'05 CIAH Workshop (Salzburg 2005).

[2] Cristea, A. I. Authoring of Adaptive Educational Hypermedia. In International Conference on Advanced Learning Technologies (ICALT 2007) (Niigata, Japan 2007), IEEE, 943-944.

[3] Cristea, A. I. Automatic Authoring in the LAOS AHS Authoring Model. In ACM Conference on Hypertext and Hypermedia (HT03) (AH Workshop) (Nottingham, UK 2003).

[4] Hendrix, Maurice and Cristea, A. I. A spiral model for adding automatic adaptive authoring to adaptive hypermedia. Journal of Universal Computer Science (JUCS), 14, Special Issue on Authoring of Adaptive and Adaptable Hypermedia (September 28, 2008), 17.

[5] Cristea, A I, Smits, D, Bevan, J, and Hendrix, M. LAG 2.0: Refining a REusable Adaptation Language and Improving on Its Authoring. In Learning in the Synergy of Multiple Desciplines. Springer, 2009.

[6] Eklund, J. and Brusilovsky, P.. Interbook: an adaptive tutoring system. (1999), UniServe Science News 12, no. 3, pp.8-13.

[7] De Bra, P., Smits, D., and Stash, N. The Design of AHA!. In In Proceeding of the ACM Hypertext Conference (Odense, Denmark 2006), 133-134.

[8] Brusilovsky, P. Adaptive Hypermedia. User Modeling and User-Adapted Interaction, Ten Year Anniversary, 11 (2001), 87-110.

[9] Dagger, D., & Wade, V. Evaluation of adaptive course construction toolkit (ACCT). In Proceedings of the 3rd International Workshop on Authoring of Adaptive and Adaptable Educational Hypermedia (A3EH'05) (2005).

[10] Conlan, O and Wade, V. Evaluation of APeLS - An Adaptive eLearning Service based on the Multi-model, Metadata-driven Approach. In Third International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (Eindhoven, The Netherlands 2004).

[11] Gaffney, C, Dagger, D, and Wade, V. Evaluation of ACTSim: a Composition Tool for Authoring Adaptive Soft Skill Simulations. In Fifth InternationaFol Conference on Adaptive Hypermedia and Adaptive Web-based Systems (Hannover, Germany 2008, pp 113-122).

[12] Hendrix, M, De Bra, P, Pechenizkiy, M, Smits, D, and Cristea, A. Defining Adaptation in a Generic Multi Layer Model: CAM: The GRAPPLE Conceptual Adaptation Model. In Times of Convergence. Technologies Across Learning Contexts. Springer, 2008.

[13] Foss, Jonathan and Cristea, Alexandra. The Next Generation of Authoring Adaptive Hypermedia Platform: Using and Evaluating the MOT3.0 and PEAL tools. In Proceedings of the 21st ACM conference on Hypertext and hypermedia (New York, NY, USA 2010), ACM Hypertext 2010, 83-92.

[14] Bothma, J D and Cristea, A I. Augmenting Authoring of Adaptation Languages via Visual Environments. UG Research in Computer Science - Theory and Applications, Student Conference, King's College London (2010).

[15] Roque, R. V. OpenBlocks: an extendable framework for graphical block programming systems. Boston, 2007.

[16] MARRON , A, WEISS, G, and WIENER, G. A decentralized approach for programming interactive applications with javascript and blockly. In Proceedings of the 2nd edition on Programming systems, languages and applications based on actors, agents, and decentralized control abstractions ACM (2012), 59-70.

[17] Scotton, Joshua. Supporting Delivery of Adaptive Hypermedia. University Of Warwick, 2013.

[18] Gosling, S. D., Vazire, S., Srivastava, S., and John, O. P. Should we trust web-based studies? A comparative analysis of six preconceptions about Internet questionnaires. In Am Psychol (2004), US National Library of Medicine National Institutes of Health, 93–104.

[19] Stash, N, Cristea, A, and De Bra, P. Learning Sytles Adaptation Language for Adaptive Hypermeida. In Proceedings of the Adaptive Hypermedia (AH) (Dublin, Ireland, pp.323-327 21-23 June 2006).