An Approach for Designing Applications in 3D Virtual Worlds

William Sawyerr
Anglia Ruskin University
Cambridge, United Kingdom
william.sawyerr@anglia.ac.uk

ABSTRACT

This paper discusses an approach for designing VW applications using the generative design grammar framework and the generative design agent model as tools. The need for a suitable method and tools for designing VW applications is evidenced by many of the problems that VWs continue to suffer due to poor design. Preliminary analyses suggest that, the combination of a method and tools created specifically for designing VW applications provides a robust and suitable approach for designing these types of applications.

CCS Concepts

• **Software and its engineering~Software design engineering**

Keywords

Methods; Design; Tools; Toolkits; Frameworks; Models

1. INTRODUCTION

In software engineering (SE), the application development process requires the design of specifications of applications that are intended to accomplish certain goals, using a set of primitive components and subject to constraints [1]. Using this rule, the application development process in virtual worlds (VWs) also requires the design of specifications that comprise goals, components, and constraints. SE already provides methods and tools for designing applications. However, VW applications are characteristically different from the classic types of applications in SE. For example, VW applications are three-dimensional and run in three-dimensional environments, while classic applications are two-dimensional and run in two-dimensional environments. Such differences have implications for design. Those methods and tools in SE that were developed for designing classic types of applications, are unable to effectively capture and describe the characteristics that are important for designing VW applications. Therefore, the VW application design activity requires methods and tools that are specifically for designing VW applications.

In spite of the above ideals, the application development process in VWs is currently at an early stage of development, largely utilising existing methods and tools for application design. These methods and tools were mostly developed for managing the application development process for the classic types of applications in SE. Therefore, their use for designing VW applications is unsuitable. The use of unsuitable methods and

tools for the VW design activity results in poorly designed specifications of VW applications. Poor design is one of the causes of usability problems in VW applications. Furthermore, poor design can also affect the adoption and use of VWs. By continuing to use unsuitable methods and tools, VW applications will continue to suffer from problems related to poor design, as well as other problems exacerbated by poor design. The use of a method created specifically for VWs could serve as a suitable guide for VW application design. Furthermore, the use of tools created specifically for designing the specifications of VW applications could enable capturing and describing the characteristics of VW applications in an effective manner.

This paper proposes a new approach for designing VW applications, inspired by generative design in architecture and built environments. The approach establishes a method that combines a generative design grammar (GDG) framework and a generative design agent (GDA) model [2] for designing VW applications. The GDG framework is used as a tool for designing the specifications of VW applications. The GDA model supplements the GDG framework, and enables the embedment of dynamic properties into the design specifications of VW applications. VWs are democratic platforms that were created so that any user could develop any type of application for any purpose. Therefore, the development of an approach that enables embedding dynamic properties into the design specifications of VW applications will create the opportunity for any VW user to be able to develop VW applications of their own, with little effort and technical expertise needed. It is hoped that such an opportunity will contribute to a future in which VW application development is democratised.

A brief overview of current methods and tools for designing VW applications is next, followed by details of the proposed approach, and finally a summary of the initial findings.

2. METHODS AND TOOLS

Methods and tools for designing VW applications usually fall into one of four general categories: (1) classic, (2) adapted, (3) ad-hoc, and (4) pertinent.

Classic are those originally developed for managing the development process for the classic types of applications in SE, and are used without any adaption for designing VW applications. Examples in this category include the use of questionnaires, online surveys, or interviews for designing VW applications.

Adapted are those developed by modifying classic methods and tools for use in designing VW applications. An example in this category is the use of VW heuristics (an adaption of Nielsen's Heuristics) for designing VW applications.

Ad-hoc are those developed on a makeshift basis and in response to the lack of pertinent design methods and tools, for use in designing VW applications.

Pertinent are those developed as suitable or more relevant solutions for designing VW applications. This paper proposes an approach for designing VW applications that uses a pertinent method and tools.

3. TOWARDS DYNAMIC APPLICATIONS

Generative design is an established approach in architecture and built environments, in which a set of rules or an algorithm generates the output of architectural models using a computer program. The proposed approach for designing VW applications combines the use of a GDG framework and a GDA model. Figure 1 illustrates the proposed approach, reflecting the integration of a GDG and a GDA.
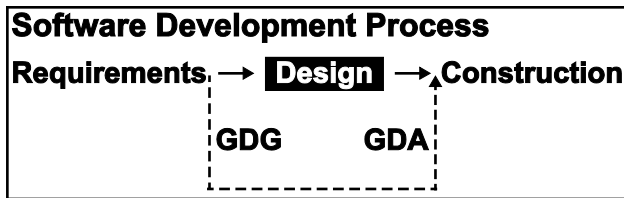

Figure 1. Approach for designing VW applications.

The concept of place is a common design metaphor in VWs. This metaphor provides a rich basis for designing VW applications, and allows the use of generative design in the VW application development process. As such, a VW application is a virtual place built to support various VW activities. Typical VW activities include gaming, socialising, or attending events. A VW application has a layout, includes objects and navigation aids for supporting various VW activities, and interaction rules embedded in the various objects. Interaction rules are typically in the form of scripts.

A GDG is a set of rules that describe a design style. A GDG G is comprised of design rules R, an initial design D_i, and a final state of the design D_f: $G = \{R, D_i, D_f\}$. The basic components of the GDG are the design rules R. The general structure of the GDG for designing VW applications consists of four sets of design rules, which are layout rules R_a, object design rules R_b, navigation rules R_c, and interaction rules R_d: $R = \{R_a, R_b, R_c, R_d\}$. The four sets of design rules correspond to the four phases of designing VW applications, which are as follows:

Layout design: developing the layout of the application in the VW, where each area has a purpose that accommodates certain intended VW activities.

Object design: configuring the application with objects that provide visual boundaries of the VW application, as well as visual cues for supporting the intended VW activities.

Navigation design: specifying navigation methods that use wayfinding aids such as hyperlinks and teleportation devices for assisting the movements of users (using avatars) between different areas of the VW application.

Interaction design: designing algorithms, writing code, and ascribing scripts to objects, to enable users to interact with the VW application.

The GDG framework provides guidelines and strategies for developing GDGs. It allows us to specify the general structure of a GDG and its basic components, which are the design rules. By using the GDG framework, GDGs can be developed for designing VW applications, reflecting a certain design style. In order to design a VW application, the GDG is applied first to generate a design specification of the VW application. Next, the GDA interprets the design specification for the design to be instantiated.

The GDA is a computational agent model with computational processes for reasoning, and therefore can be used for designing VW applications. The GDA's reasoning mechanism uses sensors and effectors as an interface between the GDA and the VW, and for constructing the VW application based on the design specification. Figure 2 is an illustration of the GDA model.
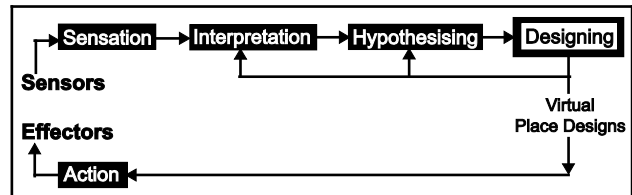

Figure 2. The GDA model and its computational processes.

The GDA wraps around the GDG and provides mechanisms for sensing and changing attributes of VW applications. The GDA's reasoning mechanism has five computational process as follows:

Sensation - using sensors to retrieve raw data from the VW to prepare for the process of interpretation.

Interpretation - interpreting the current design needs and the current state of the VW.

Hypothesising - setting up design goals that aim to eliminate mismatches between the current design needs in the VW and the current state of the VW.

Designing - providing the design of a virtual place (i.e., the VW application) in order to satisfy current design goals.

Action - planning actions for implementing the design specification in the VW, as well as activating the planned actions in the VW.

4. CONCLUSION AND OUTLOOK

Preliminary analyses show that the proposed approach effectively captures and describes several facets of VW applications, thereby suggesting that it is a robust and suitable solution for designing VW applications. Comparative evaluations are currently being conducted between the proposed approach and existing approaches for designing VW applications. There are also plans to validate and verify the new approach by using it to design and develop some VW application prototypes.

5. REFERENCES

[1] Ralph, P. and Wand, Y. 2009. A proposal for a formal definition of the design concept. In Design Requirements Engineering: A Ten-Year Perspective, K. Lyytinen, P. Loucopoulos, J. Mylopoulos, and B. Robinson, Eds. LNBIP Series. Springer-Verlag, Berlin, 103-136. DOI= http://dx.doi.org/10.1007/978-3-540-92966-6_6.

[2] Gu, N. and Maher, M. L. 2005. Dynamic designs of 3D virtual worlds using generative design agents. In Computer Aided Architectural Design Futures 2005, B. Martens and A. Brown, Eds. Springer-Verlag, Berlin, 239-248. DOI= http://dx.doi.org/ 10.1007/1-4020-3698-1_22.