Storytelling Framework with Adaptive Interaction System for Interactive Content in Virtual Museum

Chaowanan Khundam[1] and **Frédéric Noël**[2]

ABSTRACT: Virtual Reality (VR) generates realistic visualization and sensation applied to various practices. A Virtual Museum (VM) is a use case where VR may be applied to convince museum visitors to participate with a story told through digital content. Recently, immersive VR technologies have been intensively developed providing a lot of devices which support interactive VM applications. In development of interactive VM, interaction always depends on the selected device. Then content is tuned to fit specific device capacity. Major development must be addressed again whenever the virtual environment is adapted to a new device. This paper proposes a storytelling framework to assist designing interactive content which is device independent. Our framework provides a high-level abstraction of story and interaction with it which is then translated to any low-level device. A storytelling model and an interaction model are introduced to create a common language for story making. It works with the Viewer, the Asset manager, the Event editor, and the Timeline to organize virtual environments and assign interactions. An example of interactive content design on our framework is presented to demonstrate the development process. We defined three interaction systems: a 2D system, a 3D system, and a CAVE system with interaction techniques based on the interaction model. The storytelling platform has the ability to change interaction systems as required on the same story, allowing us to study the use of different interaction system on VM applications.

Keywords: Virtual Reality, Virtual Museum, Interaction System, Storytelling

DOI: 10.37936/ecti-cit.2021151.239948

Received March 9, 2020; revised April 23, 2020; accepted July 18, 2020; available online December 29, 2020

1. INTRODUCTION

Museums, the

eld of cultural heritage, and the entertainment industry are widely using Virtual Reality (VR) technology to create Virtual Museums (VM). Many low-cost or high-performance devices are used for VM applications to provide visitors with an immersive experience. 3D digitization of a story is becoming a mainstream technique to transform museum information into digital data [1, 2]. This includes the ability to interlink content for interactive and immersive applications.

1.1 Storytelling

Digital content plays an important role in developing a VM application. To create a successful VM, we need interactive content which allows user engagement. The combination of interactive content and an immersive VM must capture user interest and enable knowledge gain. The results of visitors using VMs [3] showed that virtual environments and interactive content are both important elements for learning. Moreover, the effectiveness study [4] of immersive environments indicates that cognitive involvement is increasing when interactive content is used within VM applications. Using interactive content is a better way to educate, entertain, and engage visitors for learning within a VM. Immersive VR and interactive content are the main concepts used to develop VM applications, but the keystone remains good storytelling. When storytelling is related to cultural heritage, it presents tangible and intangible assets that

[1]School of Informatics, Walailak University, Thailand. E-mail: chaowanan.kh@gmail.com
[2]University at Grenoble Institute of Technology, France. E-mail: frederic.noel@grenoble-inp.fr

characterize past activities and events that may make visitors emotion [5]. We need a style of storytelling for interactive content to drive the immersive environment which responds to user interaction. The interaction is a critical part of VM development to achieve effective storytelling.

1.2 Virtual Museum Development

Depending on the involved digital chain, five categories of VM can be found: digital media VM, online VM, interactive Web3D VM, mobile application VM, and interactive VM (on-site installation). Each type has different patterns of development and interaction with different users depending on the technology and various devices used to access the data in the VM. Most of them are device dependent platform. Especially in an interactive VM, immersive VR technology is used and plays an important role for VM development [6]. Interactions are expected, and various devices are obviously connected to the application. Therefore, the interaction system of a VM application is based on the selected device: the application structure itself is device dependent. Maintenance and service of an interaction system is restricted to the contents of all development process. It remains difficult to support technologies which change over time without content modification [7]. In order to deal with this issue, we provide a device-independent VM development platform. The application must support using any device without changing its structure.

1.3 Virtual Experience and Devices

VR research is focusing on the development of immersive devices, on the creation of realistic virtual environments, and on perception [8]. The virtual experience through these devices is also different due to performance and technology.

The 2D system to support a virtual experience needs a large screen for more perception through high resolution and a wide angle of view. The Powerwall is a special multi-monitor [9] setup that consists of multiple monitors tiled together contiguously in order to form one large screen. We use the Powerwall instead of a single large screen to get tile layouts, to get a greater screen area per length, and a greater pixel density per area. This wide display is coming in many exhibition centres and is a good technology candidate for VM.

The 3D system provides depth perception with stereoscopic display by using binocular vision. An active or passive shutter glass is used to display stereoscopic images. It works by presenting only the image intended for the left eye while blocking the right eye, then switching sides. A 120 Hz repetition of this process does not interfere with the fusion of the two images into a single 3D image [10].

The CAVE (Cave Automatic Virtual Environment) system is a full immersion environment that usually occupies a room or place where up to six projectors create a complete immersive cube box. The MIHRIAD miniCAVE is a CAVE platform at the G-SCOP laboratory [11] with five stereoscopic projections assembled as an open box. The user wears stereoscopic shutter glasses inside the miniCAVE to see 3D graphics generated and synchronized with the stereoscopic projectors. The user's movements are tracked by a motion capture system recording the real time position of the users gaze. The computer generates a pair of images, one for each of the user's eyes, based on the motion capture data. The user perceives objects apparently oating in the air, getting a proper view of what it looks like in reality.

A Head Mounted Display (HMD) is a display device worn on the head or as part of a helmet. The HMD generally has one or two small displays with transparent lenses and glass embedded in the device. A VR system with HMD provides real-time head-tracked perspective with a large angle of view, interactive control, and stereoscopic display. A user moves in the 3D space and uses motion-tracked handheld controllers to interact with the environment. The wireless controllers are additional tool for more immersive experience to users. The controller has multiple input methods which include a track pad, grip buttons, and a dual-stage trigger which has infrared sensors that detect the base stations to determine the location of the controller.

We can see that VM development is based on devices. New devices will come in the future to support future challenges. The story is expected to encode with computer science in term of semantic storytelling [12] and VM is expected into new paradigms to fasten virtual environments preparation. Interactive VM as an on-site installation is the platform where most new technologies affect application design and user interaction.

1.4 Research Goal

This paper proposed a novel method for developing a VM application providing a digital storytelling template to create interactive content and an adaptive interaction system where an application is exportable into any device. We provide exible editing tools for developers to manage the content structure. The interaction usage will be interpreted into a high-level abstraction and run on a low-level hardware device where interactions have been adapted. Storytelling specifies interaction behavior which can drive interaction in a virtual scene even when a device may be switched. We claim that the development of an adaptive interaction system will help us find a good interaction system for any given content and that is device capacities can be evaluated. This implementation is useful to deploy not only for the development of digital heritage applications, but also for any interaction system where interactive content and collaborative working are required.

2. RELATED WORK

Most VMs are usually created on a platform which supports VM structure management within a VR framework. VR and authoring frameworks are related to this concept, but the development process and tools are different. We have observed four kinds of VR framework: VR toolkits, CVE toolkits, toolkits for research, and game engines. All toolkits provide interaction and rendering systems. Interaction is based on the connection of device. Some software offers more or less standard connection services. The person in charge of developing a VR environment faces a huge set of tools, and a long process to integrate selected software bricks correctly.

Most VR frameworks propose virtual environments development and support developers to handle configuration of devices. On the other hand, the authoring frameworks provide high-level abstractions to define object behaviors in general for interaction. An authoring framework is used to preset an interactive multimedia development. It can be defined as software that allows developers to create multimedia applications and to manipulate multimedia objects.

2.1 VR Toolkits

VR toolkits provide reusable components to create VR applications. Using a toolkit allows the developers to avoid starting from scratch, and reduces the amount of low-level programming and scripting required. VR toolkits include functions for handling, displaying, distributing, or managing input devices. The primary requirements for VR development systems are performance, exibility, and ease of use. Moreover, capabilities of the environment, including cross-platform development, support for VR hardware, rapid prototyping, runtime exibility, and development interface are also considered. The following VR toolkits are still active and representative solutions: VRJuggler [13], AVANGO [14], Vrui [15], FreeVR [16], CalVR [17], MIREDGE [18]. These toolkits reduce the necessary amount of programming effect. However, the features provided by toolkits are just the basis of the VR application development. VR toolkits do not explicitly support VM narration. It lack scene graph design that emphasizes interaction between objects.

2.2 Toolkits for Research

Toolkits for VR research have features similar to VR toolkits, but are dedicated to VR research. However, they are designed for a set of specific tasks. These toolkits are interesting because they are designed to be used with VR devices as well as for further development, so that they can be used to create other VR applications. Examples of such toolkit are DIVE [19], Alice [20], inVRs [21], VARU [22], and RUIS [23]. These toolkits are used for explicit applications with a specific list of devices. Application development for research does not differ from using a VR toolkit. IDEs were added for easier management. They focus on virtual environment management rather than on device connection and display processing. In addition, resource management is accessible at a low-level for system interaction. This is a design usable for an on-site installation VM that focuses on user interaction.

2.3 CVE Toolkits

Some research toolkits have been extended to support more functions and more types of interaction. They usually enable the interconnection of heterogeneous devices and become toolkits for collaboration. A Collaborative Virtual Environment (CVE) is relevant for multiple user applications. A CVE is used to distribute device organization. Most existing CVEs support developers with easy to use ways A to handle device configuration. In order to study the interaction process, a CVE manages device connections in the system where collaboration and sharing features of virtual environment are considered. Typically, a CVE is used for collaboration and interaction of multiple users working together within a VR application [24]. However, a CVE does not just provide for multiple users in the system, but it also enables using multiple devices together in the same application. COVISE [25] is designed for collaboration through a network, and it supports presession access control using a multi-server design to distribute workload. The objective is to integrate together simulation, post-processing, and visualization functionalities. COLLAVIZ [26] focuses on webbased technology developed on the top of shared highperformance visualization platforms for scientific and industrial work to do remote analysis. Large data sets can be handled on a single workstation. COLLAVIZ allows engineers and researchers to collaborate easily access functionalities and customizations with just a simple Internet connection.

While COVISE and COLLAVIZ are focusing on interconnection, CVE (CVE application developed at G-SCOP laboratory) focuses on a meta-model framework to support structured information, including visualization and device interconnection. In CVE, every system is a machine governed by states which can be structured in a more or less complex semantic graph. CVE shares the graph between the set of interconnected devices or applications [27]. The semantic graph is controlled by model driven engineering engines [28].

2.4 Game Engine Toolkits

A game engine toolkit is a software development environment designed to provide the necessary set of features to build games. Such a toolkit brings together several core areas such as graphics, audio, networking, physics, GUI, and scripting. The objective of using a game engine is obviously to create games, but with the mentioned capacities game engines are also used to create VR applications. The following game engine toolkits are widely used to build VM applications: Unity3D, Unreal Engine 4 (UE4), and Blender [29]. VM applications can be easily designed and developed using a game engine. Game engines create closed worlds. The content may be defined once at the development stage. Behaviors are encoded with scripting languages. They provide libraries of predefined effects. With the recent deployment of head mounted devices, game engines have become VR preparation toolkits and may integrate basic functions for collaboration. They also provide connectors to VR devices. However, there are still limitations to access specific VR devices and to switch devices: interaction with the virtual environment is based on the selected device. Moreover, game engines are dedicated to a wide public and few devices (as Head mounted device) are supported. Some middle VR technologies remain necessary for extra connections. In the last few years Unity3D and UE4 have completed research development, where experiences and demonstrations fit closed virtual worlds.

2.5 Authoring Framework

Most VR frameworks are proposed as tools for virtual environment development and support device configuration. However, storytelling support and structure organization are still lacking. An authoring framework provides a high-level abstraction to define object behaviors in general and supports integration of interactive media in a story. An authoring framework is used to pre-set interactive multimedia development. It allows developers to create multimedia applications and to manipulate multimedia objects. Adventure Author [30], Storytelling Alice [31], and StoryTec [32] are examples of this framework developed to support the creation of a learning applications. In educational application development such as <e-Adventure> [33] and WEEV [34] allow non-programmers to easily create application with programming features.

The three main components of the authoring framework are content organization, content control, and content assessment [35]. The content organization allows users to structure and sequence content or other multi-media. Content control refers to the ability to determine presentation of content and storytelling to engage learners with the content. Content assessment refers to the ability to test the learning outcomes within the system. Other tools such as

Table 1: *Comparison of characteristics and features of both VR frameworks and authoring frameworks.*

	Toolkit	Main approach	Storytelling	Anim. sup.	Physic eng.	Devices	Interaction
VR toolkit	VRJuggler	Scripting	No	No	No	VR immersive	Device dependent
	AVANGO	Scripting	No	No	No	VR immersive	Device dependent
	Vrui	GUI	No	No	No	High-interactive	Device dependent
	FreeVR	GUI	No	No	No	Desktop, CAVE	Device dependent
	CalVR	XML config	No	No	No	Limited	Device dependent
	MIREDGE	Visual language	No	Yes	Yes	MVR device	Hybrid
Toolkit for research	DIVE	Scripting	No	No	No	HMD, wand,glove	Device dependent
	Alice	IDE	Yes	Yes	Yes	Desktop, HMD	Device dependent
	inVRs	XML config	No	Yes	Yes	CAVE, joy stck.	Pre-defined
	VARU	XML config	No	Yes	Yes	VR immersive	Pre-defined
	RUIS	Building blocks	No	Yes	Yes	VR immersive	Device dependent
CVE toolkit	COVISE	Visual programming	No	Yes	No	VR immersive	Device dependent
	COLLAVIZ	IDE	No	No	No	Visualization	Device dependent
	G-SCOP CVE	Scripting	No	No	No	VR immersive	Device dependent
	Second Life	LSL/GUI editor	Support	Yes	Yes	Desktop	Point and click
Game engine	Unity3D	IDE	Support	Yes	Yes	Comm. devices	Device dependent
	UE4	IDE/ Blueprints	Support	Yes	Yes	Comm. devices	Device dependent
	BGE	Logic Brick	Support	Yes	Yes	Comm. devices	Device dependent
Authoring framework	Adv. Author	GUI editor	Support	Yes	No	Desktop	3DUIs
	St. Alice	Visual Language	Yes	Yes	Yes	Desktop	3DUIs
	StoryTec	IDE/ Visual Language	Yes	Yes	No	Desktop	3DUIs
	<e-Adventure>	GUI editor	Yes	2D	No	Desktop	Point and click
	WEEV	Visual Language	Yes	2D	No	Desktop	Point and click
	Thinking Worlds	Hybrid	Support	Yes	Yes	Desktop	3DUIs
	Scratch	Visual Language	Support	2D	No	Desktop	Point and click

Thinking Worlds and Scratch support the concept of an authoring framework. They provide features and functions to create educational applications with storytelling components.

2.6 Frameworks Comparison

Existing VR and authoring frameworks do not provide complete sets of functions to address development of interactive VM applications. We compare the available technologies in Table 1 with main approach, storytelling, animation support, physics engine, supported devices, and characteristics of interaction.

The main approaches for VR frameworks use scripting with some xml configuration support. Each system is set up to adjust the parameters according to the desired values [36]. CVE toolkits and game engines usually have their own IDE (Integrated Development Environment) or specific visual language which allows scripting. However, some VR frameworks have been developed with GUI (Graphic User Interface) editors, whereas all authoring frameworks have GUIs which may be considered a feature of visual languages [18].

Storytelling requires the modules to support virtual environment creation including animation and multimedia [37]. VR toolkits and CVE toolkits do not support these features by default, while game engines can support storytelling with the features of cinematic processes and video/audio editing [38]. Even though an authoring framework is concerned with a narrative and storytelling, some of them lack this feature (Adventure Author, Thinking Worlds, and Scratch).

Most VR frameworks do not have animation features. However, some frameworks for research such as Alice, inVRs, VARU, and RUIS are designed to support making animation. CVE toolkits have few animation tools or only simple animation (COVISE). All game engines have animation features. They provides specific editors for making complex animation. Authoring frameworks support animation as a component of story ow, but some of them support only simple 2D animation. The physics engine is used to simulate collisions or gravity with animation. The game engines are useful for this feature [39] which is applied to enhance the realism of a virtual environment. For VM, physics realism is not the main trigger, but visual effects are expected to work as they do in game engines.

Immersive VR devices are used in various projects. VR frameworks support device connectivity, depending on settings and related tasks. Game engines focus on supporting commercial devices [38] to enable collaboration with their users. Usually, HMD devices and AR features are used on mobile. All authoring frameworks mainly use desktop devices and do not focus on using VR technology. All VR frameworks have the same interaction based on basic 3DGUIs, but they have different features depending on usage.

A VR framework supports device organization and a CVE supports user collaboration, but most of them do not contribute anything helpful for storytelling, animation and simulation. Game engines support game development. They provide animation and physics engines for simulation [40], but a complete tool for storytelling is still needed. Authoring frameworks have features for developing structured learning, but applications remain locked on desktop devices and need more interaction. Therefore, a complete platform integrating features from existing systems is lacking, but could out our goal.

We need a tool that allows us to develop VM applications on different devices and to use different interaction techniques. This led us to develop a new storytelling framework that creates a VM application and runs it on a variety of devices without any modification of the applications. It enables usage comparison of applications that interact with users on different devices. A storytelling platform is proposed to support a conceptual framework for developing an interactive on-site installation of VMs with the following objectives:

- To develop a complete platform for designing interactive virtual environments.
- To develop an interaction system for managing device connections.

We define a high abstraction model that supports VM storytelling definition and the details are given in the next section.

3. DESIGN AND DEVELOPMENT OF A STORY MODEL

From related work, we know there is a lack of interaction between a user and a story in many VMs. Design and development of the story model should integrate interaction within the story [41, 42]. We introduce a new storyline by adding an interaction model adapted to historical and storytelling models.

Fig. 1 is a UML diagram showing the main concepts of our story model. It was separated into three submodels. The historical model describes historical facts which are the basic knowledge and artefact references of a museum used to tell a story about a selection of these facts. This leads to the second part, the storytelling model, where it is possible to create relationships between components in terms of the virtual environment and user interaction. Then this story is accessed through an interactive system by the interaction model.

3.1 Historical Model

The historical model is developed as an abstraction of a historical story based on the CIDOC-CRM [43], the ontology for cultural heritage relationships. It provides a generic structure for the definition of history facts and is extensible to the requirements. The main classes in the model are the "historical fact" and the "component". The historical fact is the core process to present an event. It takes a component as input and returns a new modified component as output. The historical model allows events to be linked to external events by triggers. The trigger is called when the event of historical fact is entered, referring to the storytelling model. To test our framework, we implemented a VM about Wat Phra Mahathat Woramahawihan, the famous temple in the south of Thailand.

The following scene is an example of historical model implementation. It is called "Abandon ship scene". There are various legends about Phrathat. One legend mentioned that the Buddha relics (the teeth of the Lord Buddha) were kept inside. The relics were moved from Tontha Buri by Prince Tha-

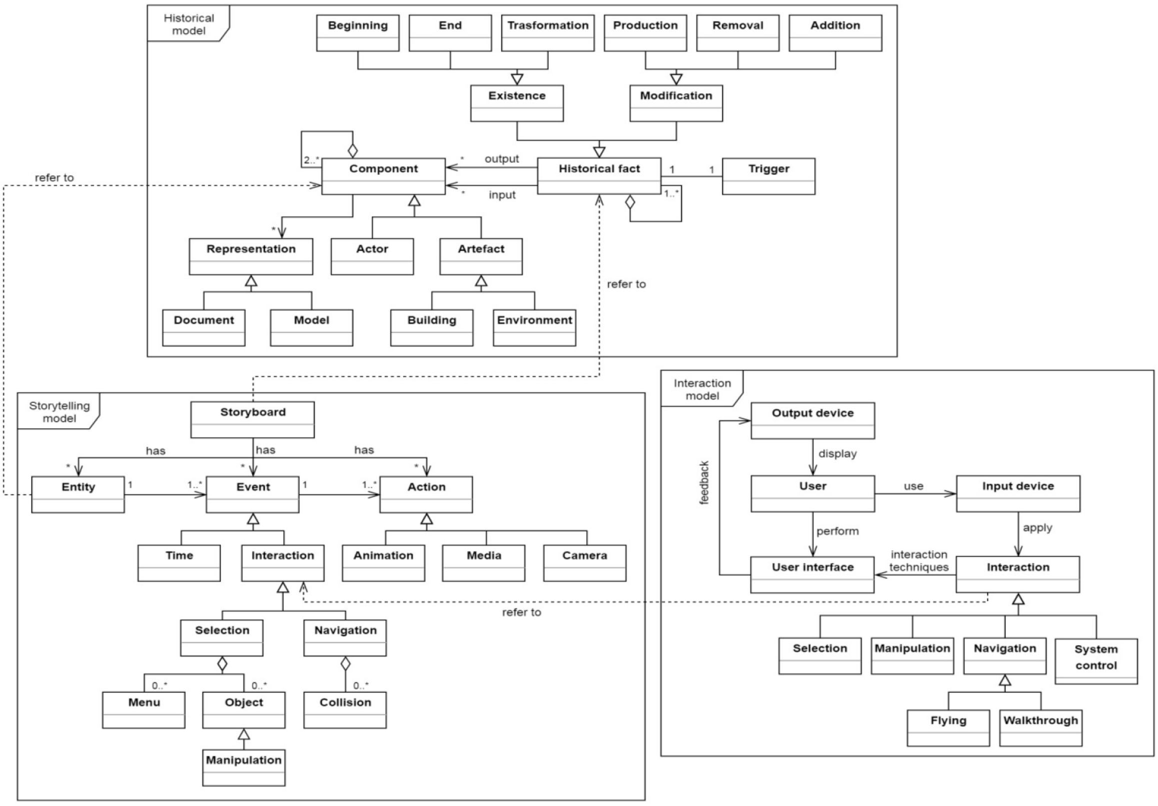


Fig. 1: *UML class diagram of the generic story model.*

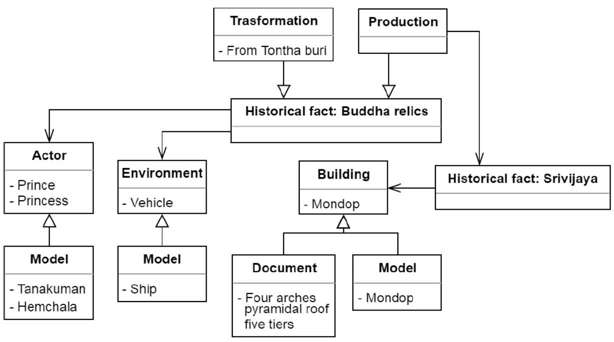


Fig. 2: *Historical model for abandon ship scene.*

nakuman and Princess Hemchala with a ship. The ship sunk and they moved the relics to a beach. It has been assumed that the original form of Phrathat was Mondop, a structure with four arches and a pyramidal roof topped with five tiers in the Srivijaya style. Here, this scene will use:

- Historical facts: Buddha relics, Srivijaya style
- Components: Prince Thanakuman, Princess Hemchala, Ship, Mondop, etc.

When the history is designed within the historical model, the implementation of the story to create a virtual environment is possible. Both historical facts and components of each scene from the historical model are used to design a storytelling model. Example, the historical model of "abandon ship scene", is shown in Fig. 2. Prince Thanakuman, Princess Hemchala, Ship and Mondop are there as the entity related to the storytelling model (see Fig. 1). These components support historical fact in the story as shown in Fig. 3.

3.2 Storytelling Model

The storytelling model uses historical facts and writes a story from the museum perception of this fact. It is a story following a storyboard model, which is usual for animation issues. The storytelling model is driven by the storyboard, with subsequent elements to outline the story and to design scenarios with their situations. The storyboard is described with three concepts: entity, event, and action. One entity is associated with many events, and one event is also associated with many actions. This structure allows many choices of actions to describe situations of the entity. The action is the way to present an entity through event conditions.

In order to drive the entity, the event is set up to define a situation for the entity. There are two types of event, time and interaction, depending how it is triggered. Event triggered by time is a condition that occurs at a specific time or delay. An interaction event is the condition to run the event triggered from a user interaction. Three kinds of interaction modes are derived from interaction events as classified by [44]: selection, manipulation, and navigation.

The story is a static or dynamic story depending

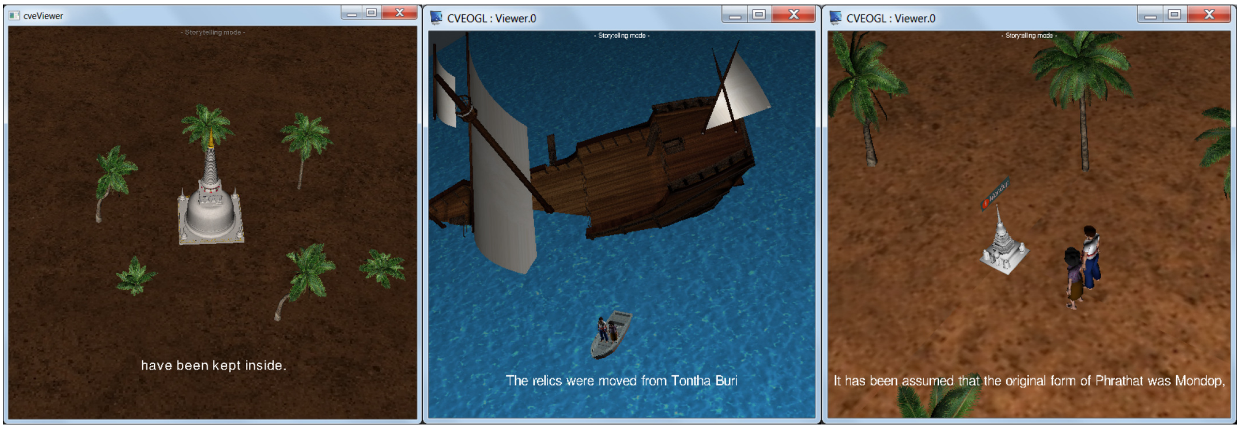


Fig.3: *The story result of abandon ship scene on the Viewer.*

on how the events are set. Animation, media, and option are the components of action used to control the story. The original feature of this storyboard is to include some optional branches in the story. A video typical provides a single sequence of views.

3.3 Interaction Model

We also define a high-level abstraction for a flexible interaction model for supporting different devices. The same interaction tasks should be applied yielding the same result, even when using different devices. Model-Driven Engineering (MDE) is the conceptual model chosen for abstract representation [45]. We developed the framework and applied MDE to define metamodels, transformations, and mappings to address the problem of interaction plasticity and to support multiple devices [46]. The interaction tasks are considered to be selection, manipulation navigation, and system control [44, 47]. All devices are based on predefined interaction techniques which are set up as a device configuration. It needs to define the relationship between the device and the interaction techniques to allow changing devices. Interaction tasks in the interaction model are linked to interaction events in the storytelling model to implement events from high-level abstractions to low-level devices.

4. THE STORYTELLING FRAMEWORK

A story model was implemented to create an operational platform. An historical model is used to transform a history into components of story in terms of relationships. A storytelling model creates a theme in terms of a virtual environment where user interaction can be added. The Interaction model is used to determine the use of devices, including interactive techniques. The details of the platform development are described below to show how these models are applied to create a VM application.

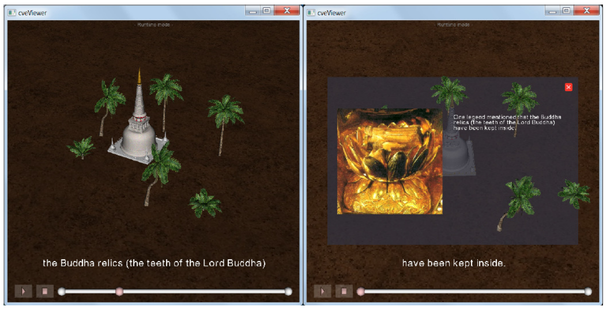


Fig.4: *The Viewer on runtime mode with additional timeline.*

4.1 Components of the storytelling platform

The storytelling platform provides story organization and high-level abstraction to define object behaviors in general for interaction. The platform was developed to support interactive content creation. The high-level abstract model is defined corresponding to storytelling with events and actions. The details of each component are described in the following subsections.

4.1.1 The Viewer

The Viewer is a display window with 3D visualization providing camera modes, navigation modes, and many API features connected to the application. This viewer was developed on top of an home made environment (G-SCOP CVE). It can be configured seamlessly to display content on a 2D desktop, on a 3D stereoscopic display, or in a CAVE. Storytelling mode and runtime mode are applied on the same Viewer as shown in Fig. 4.

4.1.2 The Asset manager

The Asset manager is a tool panel to manage all objects in a scene which can be 3D models or terrain. The two components of the Asset manager are the list of objects within the scene and the transformation

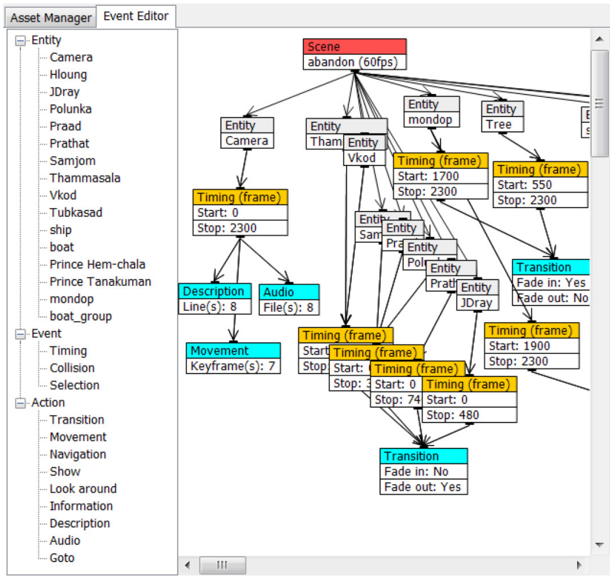


Fig.5: Event editor appearance showing the structure of the linked nodes on canvas.

attributes management. The objects are representations of historical facts in a model list. Both model and terrain can be selected to modify their transformation values through the transformation attribute management and the result of the transformation is directly displayed on the Viewer. When an icon is added, the icon name will be present under the model name on the models list. The asset manager is an original development from this study because it is directly connected to the historical model rather than connected to the geometric assets. It adds more semantics in the VR environment even though historical components are associatedto geometric assets.

4.1.3 The Event editor

The Event editor is a tool for defining high-level abstraction events. It provides a common template for every story. There is a component list on the left and a canvas on the right as shown in Fig. 5. An entity, an event, or an action is created by dragging the name from the components list and dropping it on the canvas. Story structure is made by linking these components together so that their relationships will be interpreted to make a story. The event editor is a direct GUI for the storytelling model. Other metaphors could be used, but in a first step, we propose a direct representation of the connection between entities,events,and actions.

4.1.4 The Timeline

The Timeline is a tool where a slider represents the running time of a storyline. The executed storyline is controlled by the storytelling structure within the event editor when the structure from the Event editor is interpreted. In storytelling mode, we can use the Timeline window for animation checking. In runtime

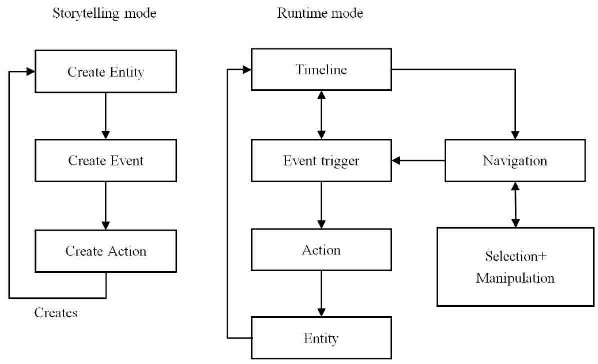


Fig.6: Working process for storytelling and runtime modes.

mode, we create a new virtual timeline in the Viewer, allowing users to jump to any frame. The Timeline provides functions to compile for interpreting, to play for checking the whole story, to run for enabling interaction, and to move the slider for animation checking.

4.2 Storytelling Platform Implementation

The platform is used for both story design and execution. It corresponds to two modes of object control in a scene, the storytelling mode and the runtime mode, as shown in Fig. 6. It explains the two modes through state chart diagrams. Each rectangle is an action, and arrows define the actions that can follow.

4.2.1 In Storytelling mode

We prepare animation and define the interactions for each object. The Viewer and Event editor will be used together to define the story using a graph association of entity, event, and action. First, all entities must be imported to the scene. Then the Event editor is used to assign events and actions to the entity. Transformation settings of each entity are organized with the Viewer links to the Event editor. The Viewer in the storytelling mode is responsible for displaying the orientation and position of objects within the scene. It cooperates with the Asset manager to adjust object transformations. It visualizes the animation result when it is used together with the Timeline slider.

4.2.2 In Runtime mode

The Timeline registers the storytelling result frame by frame to present the story as a video player does. The user can play, pause, or even approach to interact with any object in the scene at any time. When entering runtime mode, the Viewer becomes an input window and allows user interaction. The object selection triggers an interaction as defined in the Event editor. Runtime mode has an event listener for each frame. When an event is triggered, the object (entity) will run an action or jump to the next frame. Selection, manipulation, and navigation are connected to the event trigger. Thus, the process of runtime mode

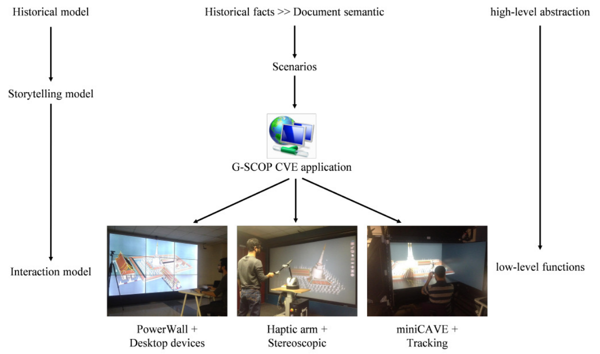


Fig. 7: *Transformation of storytelling model to low-level device connection.*

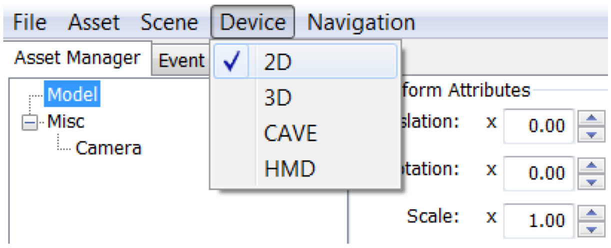


Fig. 8: *The toolbar menus for the interaction system.*

is a synchronization loop over the event listener. At any time, we can switch from one mode to the other.

4.3 Story Adaptation to Various Devices

Our storytelling model is device independent and can be projected into a device dependent virtual environment. The model transformation process begins with device installation and description of the interaction behaviors of each device. A device interaction capacity depends on a few state values, which are either Boolean, integer, real, string, or positions. Buttons and events are linked to Boolean state values. The storyboard can be defined as a set of buttons, but can be also linked with strings. Mouse and tracking systems capture position, speed, and acceleration. The definition of device behavior must associate the evolution of its state value to the edition of the interaction model. It can be achieved at a low abstraction level which is story independent. Several behaviors may be predefined and are stored at the high-level abstraction in the CVE.

As a top-down design, high-level abstraction is decomposed into lower level functions. The storytelling model is related to high-level abstractions through the definition of entities, events and actions. The interaction model will be used through event node in the Event editor to define interaction behavior along the scene.

Thus, the storytelling platform enables device and interaction switching. The Event editor is the combination of methodology between the storytelling and interaction models and is the core component of the

Fig. 9: *The interactor design for the CAVE interaction system.*

Table 2: *Selection and manipulation tasks for each interaction system.*

Interaction	2D	3D	CAVE
Selection	Lclick	Lclick+ 3D ball collision	Lclick+ 3D wand collision
Positioning	Lclick+ dragging	Selection+ translation	Selection+ translation
Rotating	Lclick+ dragging	Selection+ rotation	Selection+ rotation
Scaling	Rclick+ dragging	Rclick+ 3D ball collision+ dragging	Rclick+ 3D wand collision+ dragging

platform to create interactive content that can be adapted to any interaction system.

In the runtime mode, specification of interaction techniques will be transferred into low-level functions related to selected devices. When the interaction event is triggered, the action is launched as defined. Fig. 7 illustrates how high-level abstraction is transformed into a virtual environment that can be deployed on various immersive devices. Fig. 8 shows how it becomes as simple as a menu selection to switch to another device. No extra latency is added with the implementation.

5. USE CASE DEMONSTRATION

The use case describes the development process of an on-site installation VM which may use several devices for user interaction. A story in terms of a virtual environment is developed on the storytelling platform connected to various interaction systems. Different interaction systems are set up with immersive devices locally available for our study.

5.1 Design of the Interaction Systems

We set up three types of interaction systems with distinct user interfaces depending on the display device. The implementation interactors for each system are designed to supports of interaction techniques. Then the interaction technique will be defined as a method to interact with each system as shown in Tables 2 and 3.

5.1.1 2D interaction system

A 2D interactionsystemis set up with a Powerwall and on a desktop device. The user interface uses

Table 3: *Navigation by travel tasks for each interaction system.*

		Navigation	2D	3D	CAVE
Camera free	Translation	Forward Backward	Mouse FBscrolling	Lclick+ FBtranslation	Lclick+ FBtranslation
		Left Right	Mclick+ LRdragging	Lclick+ LRtranslation	Lclick+ LRtranslation
		Up Down	Mclick+ FBdragging	Lclick+ UDtranslation	Lclick+ UDtranslation
	Rotation	Pitch	Lclick+ FBdragging	Lclick+ UDrotation	Look up Look down
		Yaw	Lclick+ LRdragging	Lclick+ LRrotation	Turn left Turn right
Walkthrough	Translation	Forward Backward	Keyboard FB buttons	Push and pull gripping tool	Lclick Rclick
		Left Right	Keyboard LR buttons	LRtranslation of gripping tool	FBscrolling
	Rotation	Pitch	Mouse FBdragging	UDrotation of gripping tool	Look up Look down
		Yaw	Mouse LRdragging	LRrotation of gripping tool	Turn left Turn right

mouse point and click and menus on the top left. The standard interactoris a pointer.

5.1.2 3D interaction system

A 3D interaction system is set up on a stereoscopic projection display with a Virtuose 6D haptic arm or an Xbox controller. The user interface uses a 3D environment and menus on the top left. The standard interactor is a 3D pin avatar.

5.1.3 CAVE interaction system

A CAVE interaction system is set up on a full immersive CAVE (the MIHRIAD miniCAVE) with a 3D wand. The user interface uses the 3D environment and menus on the left space for better accessibility. The standard interactor is a 3D stick aligned with the wand pointer (Fig. 9).

5.2 Story Making

A story was described in the Event editor with a walkthrough navigation mode. A first-person perspective camera attached to the avatar enables users to live inside the virtual environment. "Collision" events are added to support walkthrough navigation commands. When a user navigates into a given area, we customize the actions as a result of the collision. Selection and manipulation events still play an important role for interacting with virtual environment allowing the story to be fully driven by user interaction. It is also referred to as a non-linear story, which depends on user exploration and the areas of interaction.

Fig. 9 shows the visual result of a story made in the Event editor. A collection of interactions are placed

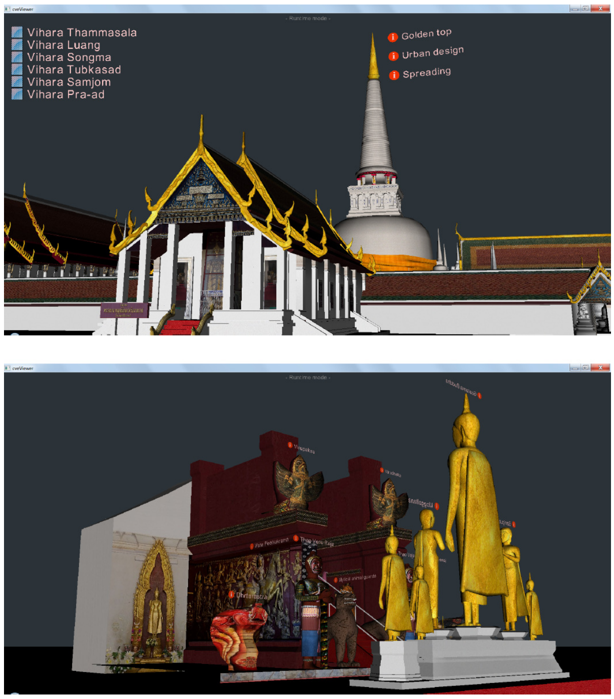


Fig.10: *The result of story description seen in runtime mode.*

on this scene where the story is run by user interaction with walkthrough mode navigation. The interactions lead to various results which are animation, information, or jumping to another scene. All interactions including selection, manipulation, and navigation are used with this story and are defined independently of the execution device.

5.3 Connection of Devices

The platform provides a function to select the device for interaction, such as the 2D system, the 3D system, and the CAVE system (Fig. 8). All devices are set up and determine interaction behaviors. The interactive content created in the Event editor provides the same story handled by interaction system without structure or program modification. The platform allows changing the interaction system with predefined interaction techniques applying the interaction model as defined in a high-level abstraction configuration when devices are installed. When the interaction event is triggered, the action is launched with the same result as defined in the storytelling model and for any selected system. Storytelling becomes device independent. The results of interaction transformation are shown in Fig. 10 - 11, where the same story is deployed on three different devices.

5.4 Interactions Report

From the development of the story experience, we designed interaction with objects in the scene fol-

Fig.11: *Interaction transformation on the same story is applied to 2D system, 3D system and CAVE system, respectively.*

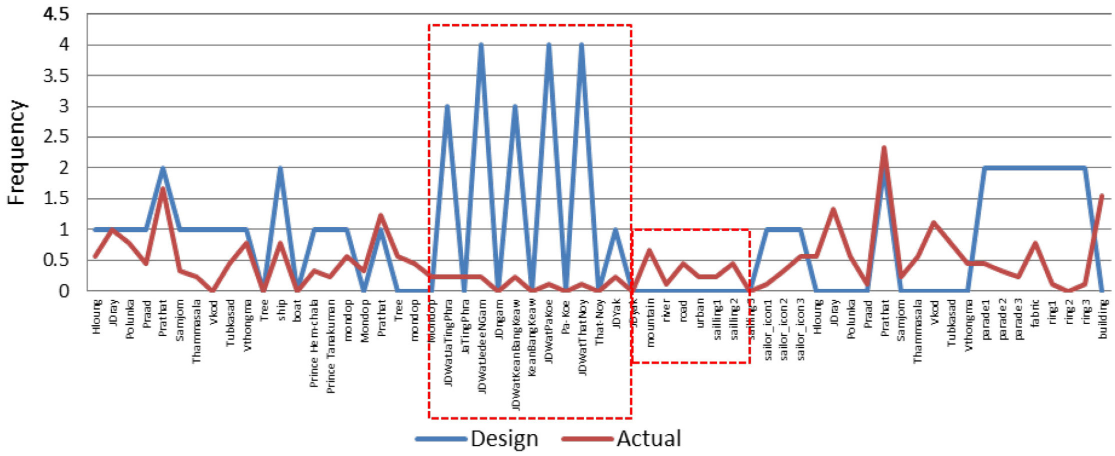


Fig.12: *Interaction with objects from actual usage compared with the interaction design of the story experience.*

lowing the story and providing information to users. However, in actual usage, the user may have a different interaction from the design. We should capture the user behavior to learn how often users interact with each object within the designed scene. If the design does not meet the needs of the users, we will recognize which object or locations should be updated to meet the needs of the users.

The storytelling framework is designed to support this idea. We kept all interaction data which can be used to generated a report for the developer after a user has finished using the system. The report will show the amount of interaction with the scene and where the interesting points were for the user. Virtual museum developers can use this report to improve their design to make it more attractive and interactive meet to the needs of users.

Fig. 12 shows the interaction design within the scene compared to the actual usage. It was found that some object interactions are quite different. Within the large dashed rectangle, we predicted a large number of interactions, but the users interacted rather less than we expected. Maybe the arrangement is not interesting, or the story in this part does not appeal to the user. Within the small dashed rectangle, we do not propose any interaction, but many users expected more information by trying to interact with objects. This report addresses the needs of users and can be used to improve the design of interactions for user benefit. Using our reports, VR becomes a user centered design environment for VM.

6. RESULTS AND CONTRIBUTIONS TO KNOWLEDGE

The result of this study will help us to design the content for interactive on-site installation VM applications. This research achieved a number of contributions to the development of applications to better immerse museum visitors in VMs. We highlight relevance topics in following subsections.

6.1 A new framework for interactive content development

A new storytelling framework to design interactive content with various devices was implemented. This framework allows us to develop applications for on-site VM installations. It also enables us to develop applications with interactive content that run on a variety of devices. This feature allows researchers to study user behavior with applications developed through devices using different interaction techniques. Using HMD devices in the learning process [4850] is better than using video and also more satisfying. However, with the same lesson, how we can prove that when we change the device the result remains the same? Our framework provides the necessary tools required to prove this idea. This is very useful for research about VM development and the development of interactive media to assess learning and user engagement. In our research, we set up devices for our experience with only three systems: the 2D system, the 3D system, and the CAVE. There may be new devices coming in the future. With the addition of the device to the platform, it will be possible to study user behavior using those new devices and to compare them with older devices or systems.

6.2 Design of interaction techniques based on user's behavior

We added functionality for reporting user interaction to the platform, allowing us to capture user data during interaction with the system. When users are finished, the platform will report elapsed times of each user as navigation time, manipulation time, and selection time. It supports developers, allowing them to investigate mistakes in the actual usage in order to design better interaction techniques in new development.

6.3 Design of interactive content based on user's behavior

Designing a VE to match user expectations is an important issue that makes VM development more effective. This platform is designed to monitor not only the time of interaction, but also the amount of interaction with each object in the scene. The report shows how many times each user interacts with each object in the scene, which allows developers to know user behavior when using their design. This information can be used to improve the interactive content of a VM to meet the needs of users. We have achieved a kind of user centered VM design.

7. LIMITATIONS AND PERSPECTIVES

7.1 Limitations

There are some limitations of our research work which can be summarized as follows:

• The storytelling platform has an interactive model that allows changing devices. However, we need to define interaction techniques for each device first. The interaction techniques should be consistent with other devices to allow the application to run without malfunction no matter which system is enabled. Some functions are not so easy to reproduce on every device.

• In the CAVE, we need to use the wand as an interactor for selection and manipulation of objects within the scene. Navigation requires a joystick because the wand does not have enough control buttons and overlaps other functions on selection and manipulation. Because there are two interactors on the CAVE system, it is difficult for users to switch between them. New solutions could improve the CAVE perception.

• The whole stories comprise many scenes with a lot of models to create animation and interaction with the user. This makes the system performance become very slow. Especially on the CAVE system, we have to separate the scenes into sub-files to prevent animation delay or interaction not responding. We cut the details of the model and cut some of the textures, so the realism of the story was reduced. The perception of the system is obviously impacted by some misbehavior of the overall rendering capacity.

• Initial experiences were very long and not compatible. We tried to reduce the experience time. So, we removed some parts of the experience, leaving only the main part, which led to some details being lost in the experience.

• Based on the design of the experience, we first designed four interaction systems: 2D system, 3D system, CAVE system, and HMD system. Because the time-consuming trial and testing of the HMD system was incomplete, we removed the HMD system, so unfortunately we cannot study the user's interaction with this system.

Most of these limitations are due to technological issues with our proof of concept system which is not perfect. However, we intend to validate the overall process and the concept of this framework is useful.

7.2 Perspectives

This research focuses on presenting a methodology for studying factors affecting user interaction. There are also topics and details in this research that are interesting to complete with additional context, which are divided into the following issues:

7.2.1 Interactor design

In this case, the interactor is an input device that is used to interact with the system. The more com-plex the system is, the more capable the interactor must be. A good interactor design will allow users to interact with the system more easily. For instance, in the CAVE system, the wand is used as the main interactor for selection and manipulation, but when navigation is added, it is not capable enough to con-trol all interaction. Thus, we have to design a new interactor or add another input device such as a joy-stick integrated with a wand. Users may have differ-ent interactions when using different interactors. It is interesting to study potentially more appropriate in-teractor designs for the application that will be used in the VM.

7.2.2 Study of interaction techniques

One interaction system can have many interaction techniques. On the 2D system, we use a mouse and keyboard as an input device and the user manipulates VE by clicking and dragging on an object. We should change the interaction technique to use buttons on keyboard instead, and study which interaction techniques are better for different contents. We can use the storytelling platform to study the interaction techniques of each interaction system. With the features of the storytelling platform, applications can use different interaction techniques on the same interaction system and content.

7.2.3 Full study of interaction systems

From the VE design (including the design of interactor and interaction techniques) we can propose a new conceptual framework to develop applications for VMs installed on-site. First, we must design the content and story to be presented in the application using various interactions. Next, we must install the interaction system and test the design of interactor and interaction techniques. This step will find the best solution of interactor and interaction technique of each system before it is compared. Then we can compare each interaction system by monitoring and capturing user interactions. Finally, the collected data can be analyzed to find the optimal interaction system and use interaction behavior of the users to improve the content designed in the first step. This approach will enable the creation of a VM application that responds to user needs with better learning. Our goal is to offer a full suite of tools to follow this process rather than proving the concept platform.

7.2.4 Future work and VMs in Thailand

For the development of the storytelling platform, we created a wide range of application interfaces that enabled us to choose the right device for the content. We highlighted that when creating applications using different devices,this can result in different levels of performance. We can enhance the user's interest in the content through the selection of appropriate devices and interaction techniques.

However, most VM applications have features adapted to multi-user applications. In the future, storytelling platforms should be developed to support multi-user and inter-user interaction. This will allow us to study the behavior between users to enhance the learning experience of the VM in the future.

What we built can be used to develop VM applications in Thailand. It still requires a design which is consistent with the immersive VR technologies that are increasingly more able to enhance the learning experience of the user. The storytelling framework will support choosing the appropriate devices and interactions for the VM's content about historical sites and culture in Thailand. This enables user interactions to enhance learning abilities for users of VM applications.

8. CONCLUSIONS

Our state of the art software demonstrates that the main expected functions, which are device organization and authoring features were developed. However, both frameworks still separate and have no integration.The creation of an interactive on-site installation VM needs storytelling abilities to support educational features for user learning. A new platform and integration layer were created to combine all features together. It not only advances interaction of the systems, but also enables device switching to study various devices' performance when using it. This paper proposed the storytelling platform to support digital interactive content for VM applications with various devices. Storytelling and interaction models are introduced to provide a high-level abstraction of story making for virtual environments. The platform provides tools to organize the story and interaction. These are the Viewer, the Asset manager, the Event editor, and the Timeline. The main process to generate the story abstraction is in the Event editor which combines storytelling and the interaction model together and also supports device changing abilities. Device connection and interaction techniques are handled on a CVE application during the story execution by the Event editor. Model transformation allows a user to switch devices and interaction techniques while the story structure remains the same.

An example of the development process is presented to explain how interactive content is implemented using low-level functions with various interaction systems. Three interaction systems: a 2D system, a 3D system, and a CAVE system, are defined. The interaction techniques are based on the interaction model. The storytelling platform has the ability to change interaction system as required. This allows us to study the use of the different interaction systems with VM applications for the same story. The storytelling platform enables independent interaction systems by creating interactive content that uses interaction without depending on a specific device. Nevertheless, it depends on the interaction techniques that are pre-defined, including selection, manipulation, and navigation. With this feature, even new devices coming in the future can be supported. It will be possible to study user behavior of new devices and to compare them with older devices or systems, while the story remains the same.

The case study was designed to be used with the interaction system to investigate how different interaction systems and stories affect the learning outcomes. The case study demonstrated the ability of a storytelling platform to create a wide range of interactive content including an interaction system which can be freely altered. In the storytelling framework, interaction report is provided to analyze user behavior on our designed story to address user needs. Then we can improve the story and the design of VE to match user expectations. This can be useful for interactive content design and to investigate the appropriate interaction system for VM applications.

References

[1] A. Solter and D. Gajski, "Project "towards the virtual museum"–exploring tools and methods for 3d digitalization and visualization," *Opvscvla archaeologica*, vol. 39, no. 1, pp. 117–124, 2018.

[2] P. Clini, L. Ruggeri, R. Angeloni, and M. Sasso, "Interactive immersive virtual museum: Digital documentation for virtual interaction.," *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, vol. 42, no. 2, 2018.

[3] J. E. Katz and D. Halpern, "Can virtual museums motivate students? toward a constructivist learning approach," Journal of Science Education and Technology, vol. 24, no. 6, pp. 776–788, 2015.

[4] S.-Y. Tsau and K.-C. Wu, "Impact of digital content curation on audience communication in interactive exhibitions," *International Journal of Arts and Commerce*, vol. 5, no. 8, pp. 59–77, 2016.

[5] E. Panopoulou, K. Kouros, A. Pasopoulou, G. Arsenikos, S. Vosinakis, P. Koutsabasis, M. Stavrakis, and D. Gavalas, "Threads: A digital storytelling multi-stage installation on industrial heritage," in *Euro-Mediterranean Conference*, pp. 457–469, Springer, 2018.

[6] M. Hirose, "Virtual reality technology and museum exhibit," *International Journal of Virtual Reality (IJVR)*, vol. 5, no. 2, pp. 31–36, 2015.

[7] C. Khundam and F. Noël, "Storytelling platform for virtual museum development: Lifecycle management of an exhibition," in *IFIP International Conference on Product Lifecycle Management*, pp. 416–426, Springer, 2017.

[8] M. K. Bekele, R. Pierdicca, E. Frontoni, E. S. Malinverni, and J. Gain, "A survey of augmented, virtual, and mixed reality for cultural heritage," *Journal on Computing and Cultural Heritage (JOCCH)*, vol. 11, no. 2, p. 7, 2018.

[9] M. Burch and H. Schmauder, "Challenges and perspectives of interacting with hierarchy visualizations on large-scale displays," in *Proceedings of the 11th International Symposium on Visual Information Communication and Interaction*, pp. 33–40, 2018.

[10] C. Han, G. Xu, Y. Jiang, H. Wang, X. Chen, K. Zhang, J. Xie, and F. Liu, "Stereoscopic motion perception research based on steady-state visual motion evoked potential," in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 3067–3070, IEEE, 2019.

[11] J. Basset and F. Noël, "Added value of a 3d cave within design activities," in *International Conference on Virtual Reality and Augmented Reality*, pp. 230–239, Springer, 2018.

[12] J. Flotyński, "Semantic modelling of interactive 3d content with domain-specific ontologies," *Procedia Computer Science*, vol. 35, pp. 531–540, 2014.

[13] A. Bierbaum, C. Just, P. Hartling, K. Meinert, A. Baker, and C. Cruz-Neira, "Vr juggler: A virtual platform for virtual reality application development," in *Virtual Reality, 2001. Proceedings. IEEE*, pp. 89–96, IEEE, 2001.

[14] R. Kuck, J. Wind, K. Riege, M. Bogen, and S. Birlinghoven, "Improving the avango vr/ar framework: Lessons learned," in *Workshop Virtuelle und Erweiterte Realität*, pp. 209–220, 2008.

[15] O. Kreylos, "Environment-independent vr development," in *International Symposium on Visual Computing*, pp. 901–912, Springer, 2008.

[16] W. R. Sherman, D. Coming, and S. Su, "Freevr: honoring the past, looking to the future," in *The Engineering Reality of Virtual Reality 2013*, vol. 8649, p. 864906, International Society for Optics and Photonics, 2013.

[17] J. P. Schulze, A. Prudhomme, P. Weber, and T. A. DeFanti, "Calvr: an advanced open source virtual reality software framework," in *The Engineering Reality of Virtual Reality 2013*, vol. 8649, p. 864902, International Society for Optics and Photonics, 2013.

[18] G. Loup, S. George, I. Marfisi, and A. Serna, "A visual programming tool to design mixed and virtual reality interactions," International Journal of Virtual Reality (IJVR), vol. 18, no. 02, pp. 19–29, 2018.

[19] C. Carlsson and O. Hagsand, "Dive a multiuser virtual reality system," in *Virtual Reality Annual International Symposium*, 1993., 1993 IEEE, pp. 394–400, IEEE, 1993.

[20] M. Conway, R. Pausch, R. Gossweiler, and T. Burnette, "Alice: a rapid prototyping system for building virtual environments," in *Conference Companion on Human Factors in Computing Systems*, pp. 295–296, ACM, 1994.

[21] C. Anthes and J. Volkert, "invrs–a framework for building interactive networked virtual reality systems," in *International Conference on High Performance Computing and Communications*, pp. 894–904, Springer, 2006.

[22] S. Irawati, S. Ahn, J. Kim, and H. Ko, "Varu framework: Enabling rapid prototyping of vr, ar and ubiquitous applications," in *IEEE Virtual Reality 2008*, pp. 201–208, IEEE, 2008.

[23] T. M. Takala, "Ruis: A toolkit for developing virtual reality applications with spatial interaction," in *Proceedings of the 2nd ACM symposium on Spatial user interaction*, pp. 94–103, ACM, 2014.

[24] T. Wright and G. Madey, "A survey of collaborative virtual environment technologies," *University of Notre Dame-USA*, Tech. Rep, pp. 1–16, 2008.

[25] D. Rantzau, U. Lang, R. Lang, H. Nebel, A. Wierse, and R. Ruehle, "Collaborative and interactive visualization in a distributed high performance software environment," in *High Performance Computing for Computer Graphics and Visualisation*, pp. 207–216, Springer, 1996.

[26] F. Dupont, T. Duval, C. Fleury, J. Forest, V. Gouranton, P. Lando, T. Laurent, G. Lavou´e, and A. Schmutz, "Collaborative scientific visualization: the collaviz framework," in *JVRC 2010 (2010 Joint Virtual Reality Conference of EuroVR-EGVE-VEC)*, 2010.

[27] C. Pontonnier, T. Duval, and G. Dumont, "Sharing and bridging information in a collaborative virtual environment: application to ergonomics," in *2013 IEEE 4th International Conference on Cognitive Infocommunications (CogInfoCom)*, pp. 121–126, IEEE, 2013.

[28] A. R. Da Silva, "Model-driven engineering: A survey supported by the unified conceptual model," *Computer Languages, Systems & Structures*, vol. 43, pp. 139–155, 2015.

[29] A. Zarrad, "Game engine solutions," in *Simulation and Gaming*, IntechOpen, 2018.

[30] J. Robertson and K. Nicholson, "Adventure author: a learning environment to support creative design," in *Proceedings of the 6th international conference on Interaction design and children*, pp. 37–44, ACM, 2007.

[31] C. Kelleher, R. Pausch, R. Pausch, and S. Kiesler, "Storytelling alice motivates middle school girls to learn computer programming," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 1455– 1464, ACM, 2007.

[32] S. Göbel, L. Salvatore, and R. Konrad, "Storytec: A digital storytelling platform for the authoring and experiencing of interactive and nonlinear stories," in *Automated solutions for Cross Media Content and Multi-channel Distribution, 2008. AXMEDIS'08. International Conference on*, pp. 103–110, IEEE, 2008.

[33] J. Torrente, A. Del Blanco, E. J. Marchiori, ´ P. Moreno-Ger, and B. Fernández-Manjón, "<eadventure>: Introducing educational games in the learning process," in *Education Engineering (EDUCON), 2010 IEEE*, pp. 1121–1126, IEEE, 2010.

[34] E. J. Marchiori, J. Torrente, Á. del Blanco, P. Moreno-Ger, P. Sancho, and B. Fernández-Manjón, "A narrative metaphor to facilitate educational game authoring," *Computers & Education*, vol. 58, no. 1, pp. 590–599, 2012.

[35] Á. Del Blanco, E. J. Marchiori, J. Torrente, I. Martínez-Ortiz, and B. Fernández-Manjón, "Using e-learning standards in educational video games," *Computer Standards & Interfaces*, vol. 36, no. 1, pp. 178–187, 2013.

[36] L. Hu, Z. Liu, and J. Tan, "A vr simulation framework integrated with multisource cae analysis data for mechanical equipment working process," *Computers in Industry*, vol. 97, pp. 85–96, 2018.

[37] W.-Y. Hwang, R. Shadiev, J.-L. Hsu, Y.-M. Huang, G.-L. Hsu, and Y.-C. Lin, "Effects of storytelling to facilitate efl speaking using web-based multimedia system," *Computer Assisted Language Learning*, vol. 29, no. 2, pp. 215–241, 2016.

[38] J. Gregory, *Game engine architecture*. crc Press, 2018.

[39] E. Lengyel,*Game Engine Gems 3*. AK Peters/CRC Press, 2016.

[40] E. Paravizo and D. Braatz, "Using a game engine for simulation in ergonomics analysis, design and education: An exploratory study," *Applied ergonomics*, vol. 77, pp. 22–28, 2019.

[41] M. Carrozzino, M. Colombo, F. Tecchia, C. Evangelista, and M. Bergamasco, "Comparing different storytelling approaches for virtual guides in digital immersive museums," in *International Conference on Augmented Reality, Virtual Reality and Computer Graphics*, pp. 292–302, Springer, 2018.

[42] S. Sylaiou, V. Kasapakis, D. Gavalas, and E. Dzardanova, "Avatars as storytellers: affective narratives in virtual museums," *Personal and Ubiquitous Computing*, pp. 1–13, 2020.

[43] M. Doerr, "The cidoc conceptual reference module: an ontological approach to semantic interoperability of metadata," *AI magazine*, vol. 24, no. 3, p. 75, 2003.

[44] D. A. Bowman, "Interaction techniques for common tasks in immersive virtual environments," *Georgia Institute of Technology*, 1999.

[45] G. Calvary, J. Coutaz, D. Thevenin, Q. Limbourg, L. Bouillon, and J. Vanderdonckt, "A unifying reference framework for multi-target user interfaces," *Interacting with computers*, vol. 15, no. 3, pp. 289-308, 2003.

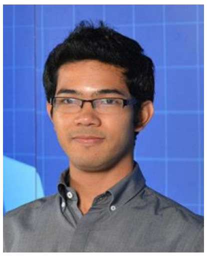
[46] J. Coutaz and G. Calvary, "Hci and software engineering for user interface plasticity," 2012.

[47] J. J. LaViola Jr, E. Kruijff, R. P. McMahan, D. Bowman, and I. P. Poupyrev, *3D user interfaces: theory and practice*. Addison-Wesley Professional, 2017.

[48] D. Checa and A. Bustillo, "Advantages and limits of virtual reality in learning processes: Briviesca in the fifteenth century," *Virtual Reality*, vol. 24, no. 1, pp. 151-161, 2020.

[49] D. Checa and A. Bustillo, "A review of immersive virtual reality serious games to enhance learning

and training," *Multimedia Tools and Applications*, vol. 79, no. 9, pp. 5501-5527, 2020.
[50] S. M. Sepasgozar, "Digital twin and web-based virtual gaming technologies for online education: A case of construction management and engineering," *Applied Sciences*, vol. 10, no. 13, p. 4678, 2020.



Chaowanan Khundam received his B.S. in Mathematics and M.S. in Computer Science from Prince of Songkla University, Thailand, in 2005 and 2011, respectively. He received his Ph.D. in Industrial Engineering from University at Grenoble Institute of Technology, France, in 2018. His main research interests are mostly in the area of Virtual Reality, Augmented Reality, Human Computer Interaction, 3D animation, simulation with game engine, special effects, and rendering.

Frédéric Noël is a professor of University at Grenoble Institute of Technology in France. Coming from the Mechanical Design disciplines, his interests are mainly Computer Aided Design and Collaboration assistance for designers. His current projects concern definition of intuitive design office tools. Frédéric Noël was the coordinator of a European Research Infrastructure about Visualisation and Interactions called VISIONAIR providing Scientific Visualisation, Virtual Reality, and Augmented Collaborative Environments for Research activities, but also Ultra-High Definition networking facilities.