Preserving story choreographies across multiple platforms

An approach to platform-independent reuse of characters' behaviors for games, simulations, animations and interactive videos

Demetrius Lacet
Universidade Aberta
INESC TEC
Portugal
demetriuslacet@hotmail.com

Filipe Penicheiro
INSESC TEC
Universidade Aberta
Portugal
filipe.penicheiro@inesctec.pt

Leonel Morgado
Universidade Aberta
INESC TEC
Portugal
leonel.morgado@uab.pt

António Coelho
Universidade Porto
INESC TEC
Portugal
acoelho@fe.up.pt

ABSTRACT

This article presents an approach that allows the reuse of choreographies of digital stories, regardless of the platform for which they have been developed. Nowadays, story choreographies are linked to either game technology, simulations or interactive animations for which they have been implemented, so they become unavailable as these platforms become obsolete. This limits their reuse, because to perpetuate these digital stories requires new and repeated efforts and investments in its re-creation for new technological platforms that appear in substitution of the previous ones. We propose a methodology to safeguard the semantics of this "narrative choreographies" in an independent way, allowing them to be automatically adapted to new technological platforms when the original ones become obsolete.

CCS CONCEPTS

Information systems -> Multimedia information systems

KEYWORDS

Content production; e-learning, digital storytelling, training, choreographies, multiplatform

ACM Reference format:

D. Lacet, F. Penicheiro, L. Morgado and A. Coelho. 2019. Preserving story choreographies across multiple platforms. An approach to platform-independent reuse of characters behaviours for games, simulations, animations and interactive videos. In *Proceedings of ARTECH 2019, 9th International Conference on Digital and Interactive Arts (ARTECH 2019),* October 23-25, 2019, Braga, Portugal. ACM, New York, NY, USA, *4 pages.* https://doi.org/10.1145/3359852.3359916

ARTECH 2019, October 23–25, 2019, Braga, Portugal

ACM.ISBN.978-1-4503-7250-3/19/10…$15.00 https://doi.org/10.1145/3359852.3359916

1 Introduction

Outside the entertainment industry, one of the challenges of widespread adoption of interactive animations, interactive videos, games or computer simulations is the low reuse of software components that constitute its content, when the platforms or technological solutions for which they were developed become obsolete. For example, in a serious professional training game, the behavior exemplified for an employee and the clients he/she attends (that is, the choreography of the training narrative) in a classic 3D simulator should be able to be used in other immersive or mixed reality technology, without having to reprogram it totally. The use of a model-based architecture to specify this type of choreography has been proposed as one of the ways to tackle this challenge [4], as mentioned in section 2. The proposal presented here is based on the semantic specification and automatic adaptation of choreographies to the various platforms. Specifically, we present the methodology created for this purpose and an example of application of this methodology: an independent animation service based on safeguarding the semantics of choreographies independently, which allows them to be automatically adapted to the new platforms when the original ones become obsolete.

2 Related work

The reuse of software components on different technological platforms have been the focus of different research, particularly those related to the development of serious games [1][2] and character animations, here called "virtual choreographies". Producers, creative artists and programmers are concerned with the investment needed in the creation of these components and are trying to multiply their useful life span, achieving greater sustainability and better cost-effectiveness of projects [3]. One of the problems identified in the literature is that virtual choreographies are linked to a specific technological solution [4]. Whether this technology solution is a game / simulation / animation produced from scratch or based on a platform like virtual worlds or a game engine, when it becomes obsolete the choreographies are not reused because they were implemented with specific code to the original solution.

This technological “linkage” is part of the wider problem of reusing components in the information technology sector [5] and is also related to the vendor lock-in problem [6], although in the present case, more than the supplier, the issue is the dependency of a technological solution or technological platform.

In the video game industry, there have been reports of this problem at least since 2005, following the assessment of the use of the game Spectrum Warrior [7] in light infantry commanders training by the US Army, who found that they had to face the behaviors of the characters of the game as something already developed and static: "Initially, instructors may make time for using a new training game. However, if the game cannot be modified according to their experience, their changing training needs, or the emerging conditions of the current operational environment, then the game will not be utilized "[8].

One way to overcome the limitations of a technological “linkage” in this context may be the use of a model-driven architecture (MDA) applied to virtual choreographies, as proposed by Silva et al. [4]. In this context, choreography is understood as "the description of a set of actions that must or can be performed by a group of participants, including the objectives to be achieved and any restrictions that may exist" [4].

The possibility of choreographing actions independently of the destination platform makes them more reusable, which allows for greater sustainability than currently exists.

3 Issues and Objectives

The present proposal is based on aspects of the xAPI specification (Experience Application Programming Interface) [9], oriented to the post facto registration of the activities that take place in interactive environments. However, we intend to be able to specify choreographies of narratives as precursors to their occurrence, not as a record of what happened. Thus, as in xAPI, we explain the elements that compose a virtual choreography in standardized statements, composed of actors, locations, objects, actions and context (Figure 1).

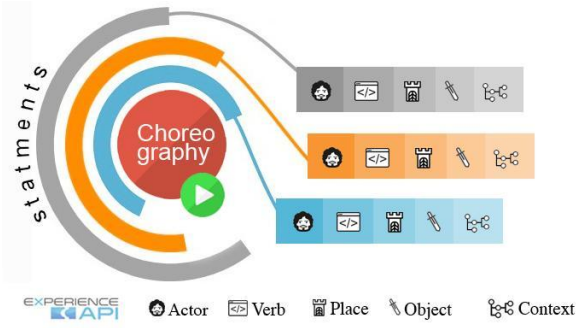

Figure 1: Elements of a choreography divided into metadata.

Still following the xAPI specification, the semantic space of these statements is defined by a singular specification, called "recipe". However, our idea is to use two types of recipes: the "choreography recipe" (Fig. 2, label 2) for the narrative semantic space; and the "platform recipe" (Fig. 2, tag 3) for the semantic space of each reproduction platform.

In this proposed model, a semantic mapping was included between choreography recipes and platform recipes (Fig. 2, label 4). Through it, a recipe for choreography can be mapped to any number of platform recipes. And vice versa: a platform recipe can be mapped to any number of choreography recipes.

Through these mappings, an automatic service can convert any choreography to any platform, provided there is a mapping between the respective recipes (Fig. 2).

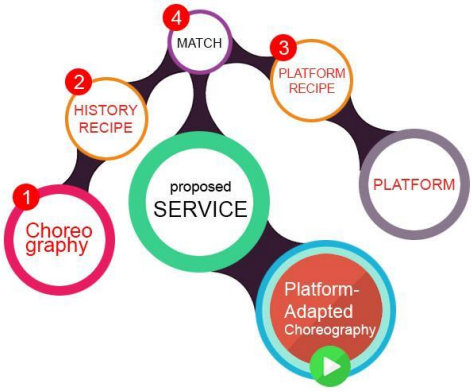

Figure 2: Diagram of the elements of the methodology

4 Prototype

We exemplify the concept from a recipe of choreographies for histories of Arthurian mythology (Fig. 3). In these recipes we have a character (King Arthur), two locations (Avalon Castle hall and Avalon Castle tower) and a verb described by the xAPI specification as ("is at") indicating the current location of the character.

```
{
   "actor":[
      {
         "id":"A01",
         "name":"Artur",
         "objectType":"Cavaleiro"
      }
   ],
   "verb":[
      {
         "id":"http://chic.mog-
technologies.com/exapi/verbs/is_at",
         "name":"is at"
      }
   ],
   "place":[
      {
         "id":"P01",
         "name":"Salao"
      },
      {
         "id":"P02",
         "name":"Torre"
```

```
      }
   ]
}
```

Figure 3: JSON with the choreography recipe structure for stories from Arthurian mythology.

In this example, we could create choreographies with endless sequences with King Arthur in the hall or tower. Figure 4 shows a choreography with two statements: King Arthur is in the hall; King Arthur is in the tower. But it would be enough to add other statements to make King Arthur return to the hall, return to the tower, etc. - each possible combination of the semantic elements of the recipe of Figure 3 in different sequences of statements of Figure 4 results in different choreographies.

```
{
   "receita":"receita_artur",
   "historia":[
      {
         "name":"A01",
         "verb":"http://chic.mog-
technologies.com/exapi/verbs/is_at",
         "place":"P01"
      },
      {
         "name":"A01",
         "verb":"http://chic.mog-
technologies.com/exapi/verbs/is_at",
         "place":"P02"
      }
   ]
}
```

Figure 4: JSON with choreography structure.

The platforms that want to be able to reproduce these choreographies will have to make explicit their semantic space, through a platform recipe. For example, Figure 5 exemplifies the recipe of a platform that allows one actor to be represented at two locations. The actor is simply identified by the identifier "1" and the locations by the identifiers "1p" and "2p". The available verb has the identifier "isat".

```
{
   "actor":[
      {
         "id":"1",
         "name":"avatar1",
         "objectType":"avatar"
      }
   ],
   "verb":[
      {
         "id":"isat",
         "name":"is at"
      }
   ],
   "place":[
      {
         "id":"1p",
         "name":"p1"
      },
      {
         "id":"2p",
         "name":"p2"
      }
   ]
}
```

Figure 5: JSON with the platform recipe structure

It is only the platform that knows how to interpret sequences of statements for this semantic space. For example, actor "1" can either be a simple flat image overlaying a background image, but also be an identification of a 3D feature with its own animations and associated artificial intelligence - only the platform knows what "actor 1" is. The same occurs with the interpretation of the verb "isat" and the places "1p" and "2p", as well as other semantic elements that may be present in a recipe.

The choreography recipes (Figure 4) and platform recipes (Figure 5) are mapped by associating their elements, using the "attribs" element of the xAPI specification to assign this mapping its own semantics. As can be seen in figure 6, the mapping of the actor "A01" with the actor "1" has the attribute "id_receita.name", that is, this mapping is associated to the text "Artur" of the choreography recipe. Other mappings can occur, such as "P01" to "1p" with the attribute "Hall" and even with other types of values or more advanced degrees of meaning, such as behavioral and decision rules, for example

```
{
   "actor":[
      {
         "id_receita":"A01",
         "id_plataforma":" 1",
         "attribs":[
            {
               "value":"id_receita.name"
            }
         ]
      }
   ],
   "verb":[
     … //resto do mapeamento
```

Figure 6: JSON with choreography structure

The proposed service can generate the choreography of Figure 4 in statements adapted to the semantic space of the platform of Figure 5. For example, instead of mentioning that A01 is in place P01, the generated statement will indicate that 1 (attrib Artur) isat 1p (atrib Hall).

If the platform described by the recipe in figure 5 is, for example, a text adventures system, this statement can generate the textual description:

```
ARTUR IS AT HALL
```

However, if the platform were a comic book system (CB), it could show the same statement as an image in one of the frames of the CB, without even needing to use the attributes: King Arthur would correspond to the image of the system for the actor "1" and the Hall would be the image that the system had for the location "1p" (Figure 7).

Note that these images do not have to be really King Arthur and a Hall: they can be just pins of board games, avatars and predefined spaces of the system, etc.: each platform recipe presents the semantic space that it has and each mapping makes the feasible specification.

Figure 7: (a) CB platform (b) concept in 3D game

In the same way, if the platform were a three-dimensional game, the same statement would correspond to presenting the avatar "1" in place "1p", eventually with the attribute "Artur" to be used as a label overlapped to the avatar (fig 7b).

4 Final Reflections

The approach presented in this paper allows that the choreographies created for one platform can be reproduced in others, provided that the respective semantic spaces are defined and mapped. This opens up the possibility of increasing its useful life and stimulating the creation of virtual choreographies of games, simulators and interactive animations in teaching and training environments. We anticipate that this freedom also allows the analysis of the choreography according to different visions, for the purpose of evaluation, study or others, according to the generalization of the method.

ACKNOWLEDGMENTS

This work is co-financed by the ERDF – European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 and the Lisboa 2020 under the PORTUGAL 2020 Partnership Agreement, and through the Portuguese National Innovation Agency (ANI) as a part of project CHIC «POCI-01-0247-FEDER-024498»".

REFERENCES

[1] W. van der Vegt,., E. Nyamsuren, W. Westera,, (2018). Making Serious Games with Reusable Software Components. In Joint Int. Conference on Serious Games (pp. 13-16).

[2] I. A. Stănescu, A. Stefan., T. Lim., F. Belotti, J. Baalsrud Hauge,.R., Wendrich, G. Neagu,,. (2014). Strategies and Tools to Enable Reuse in Serious Games Ecosystems and Beyond. In Let's build the future through learning innovation.: Procs. of the 10th Int. Scientific Conference "eLearning and Software for Education" Bucharest, April 24 - 25, 2014 (Vol. 1, pp. 363-372).

[3] E. Folmer. (2007) Component Based Game Development – A Solution to Escalating Costs and Expanding Deadlines?. In: Schmidt H.W. et al. (eds.) Component-Based Software Engineering. CBSE 2007. Lecture Notes in Computer Science, vol 4608. Springer, Berlin, Heidelberg

[4] E. Silva . N. Silva., L. Morgado (2014) Model-Driven Generation of Multi-user and Multi-domain Choreographies for Staging in Multiple Virtual World Platforms. In: Ait Ameur Y. et al. (eds) Model and Data Engineering. MEDI 2014. Lecture Notes in Computer Science, vol 8748. Springer, Cham.

[5] B. Weide W. Ogden., S. Zweben (1991). Reusable Software Components. Advances in Computers, 33(C), pp. 1-65.

[6] G. C. Silva., L. M. Rose, R. Calinescu, (2013). Towards a model-driven solution to the vendor lock-in problem in cloud computing. In 2013 IEEE 5th Int. Conference on Cloud Computing Technology and Science, v. 1, pp. 711-716). IEEE.

[7] Full Spectrum Warrior [software]. (2004). California: THQ

[8] S. Beal, (2005). Using games for training dismounted light infantry leaders: Emergent questions and lessons learned. Research Report 1841. Fort Benning, GA, EUA: U.S. Army Research Institute for the Behavioral and Social Sciences.

[9] A .Corbi,. D. Burgos, (2014). Review of current student-monitoring techniques used in elearning-focused recommender systems and learning analytics. The Experience API & LIME model case study. International Journal of Artificial Intelligence and Interactive Multimedia, 2 (7).