

## Accepted Manuscript

A case study on service-oriented architecture for serious games

Maira B. Carvalho, Francesco Bellotti, Riccardo Berta, Alessandro De Gloria,  
Giorgia Gazzarata, Jun Hu, Michael Kickmeier-Rust

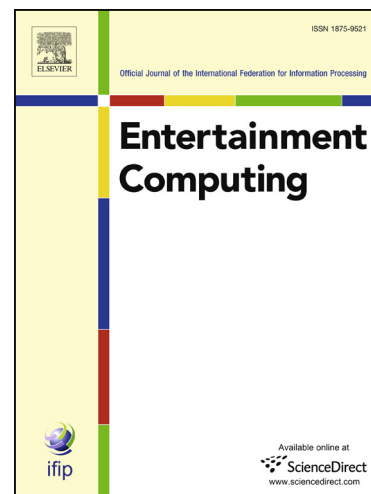
PII: S1875-9521(14)00044-5  
DOI: <http://dx.doi.org/10.1016/j.entcom.2014.11.001>  
Reference: ENTCOM 143

To appear in: *Entertainment Computing*

Received Date: 15 April 2014  
Revised Date: 21 October 2014  
Accepted Date: 12 November 2014

Please cite this article as: M.B. Carvalho, F. Bellotti, R. Berta, A.D. Gloria, G. Gazzarata, J. Hu, M. Kickmeier-Rust, A case study on service-oriented architecture for serious games, *Entertainment Computing* (2014), doi: <http://dx.doi.org/10.1016/j.entcom.2014.11.001>

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.



# A case study on Service-Oriented Architecture for Serious Games

Maira B. Carvalho <sup>a\*</sup>, Francesco Bellotti <sup>a</sup>, Riccardo Berta <sup>a</sup>, Alessandro De Gloria <sup>a</sup>, Giorgia Gazzarata <sup>a</sup>, Jun Hu <sup>b</sup>, Michael Kickmeier-Rust <sup>c</sup>

<sup>a</sup> DITEN - University of Genoa, Via Opera Pia 11A, 16145, Genoa, Italy

<sup>b</sup> Dept. of Industrial Design, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, Netherlands

<sup>c</sup> Knowledge Technologies Institute, Graz University of Technology, Inffeldgasse 13/5th floor, 8010, Graz, Austria

---

## Abstract

Service-Oriented Architecture (SOA) is a set of practices for architectural design of software that exploits services as loosely coupled components orchestrated to deliver various functionalities. The SOA paradigm is not well established in the Serious Games (SG) domain, but it is expected to provide benefits, particularly in reducing the conceptual and technological complexity of the development. In this paper, we propose and study the application of a SOA approach to SG development. We have used the SOA approach to develop an adaptive serious game for teaching basic elements of probability to high school and entry-level university students, called *The Journey*. Details of the architecture implementation are offered, as well as the results of an evaluation of the system using the Architecture Tradeoff Analysis Method (ATAM). Based on our experience, we argue that the SOA approach can make SG development shorter, more flexible and more focused.

*Keywords:* adaptive serious games, games for learning, serious games, serious games development, service oriented architecture, architecture tradeoff analysis method, ATAM.

---

\* Corresponding author

Email address: [maira.carvalho@elios.unige.it](mailto:maira.carvalho@elios.unige.it) (Maira B. Carvalho)

## 1. Introduction

Games are gaining increasing importance as educational and training tools. Serious Games (SGs) – as games used for purposes other than to simply entertain are often called [1] – have been shown to have a lot of potential in education [2–4], offering the possibilities of making learning more engaging and satisfying [5]. Among the benefits are their role in engaging and motivating learners [2] and their ability to expose learners to experiences that would be impossible, unsafe or at least impractical to reproduce in the real world [1,6,7].

However, there is still a long way to make SGs widely deployed, especially because of the high development costs. In this paper, we propose the application of a Service Oriented Architecture (SOA) approach to serious game development as a desirable and beneficial solution for the field, resulting in efficient development and high quality products. In SOA, software is built as a set of independent, loosely coupled components that provide self-contained functionalities (services) to other components and applications. By employing the core principles of SOA, such as modularization and compositionality, we expect to achieve flexibility in the development of serious games and to enable the reuse of software parts. The SOA principles are already widely and successfully employed in several areas of software engineering, but the examples in the SG domain are limited.

To illustrate the benefits of the SOA approach in SG development, we report a case study on the use of a web service based on the Competence-based Knowledge Space Theory (CbKST) [8] to develop an adaptive serious game for teaching basic elements of probability to high school and entry-level university students, called *The Journey*. The paper also presents the results of an evaluation of the system employing the Architecture Tradeoff Analysis Method (ATAM), using the conclusions of the evaluation to give an account of the constraints, benefits and changes in the programming paradigm that are relevant to SG development.

In short, the novelty presented here consists of a demonstration of how the SOA approach can improve the process of SG development through component reuse, and how it can enhance product quality by enabling the implementation of features that are still rare in SGs, such as adaptation techniques, learning analytics, social media integration, etc.

This paper is organized as follows. In the next section, we define Service Oriented Architectures and list its possible benefits and drawbacks in SG development. Section 3 presents a compilation of projects using SOA or similar approaches. In Section 4, we discuss the concept of adaptivity in learning environments and present the Competence-based Knowledge Space Theory (CbKST), explaining how it can be used in adaptive SGs. Subsequently, we describe the game *The Journey*, offering details of its architecture and implementation. Section 6 presents an ATAM evaluation of the system architecture and discusses the benefits and drawbacks of applying SOA in SG development. Finally, we present the discussion and conclusion of our work and pointers for future research.

## 2. Service Oriented Architectures

A Service-Oriented Architecture (SOA) is “a software architecture that implements business processes or services by using a set of loosely coupled, black-box components

orchestrated to deliver a well-defined level of service” [9]. It is a set of ideas, recommendations, policies and practices for architectural design. One of its goals is to employ modularization and compositionality to achieve flexibility and to enable the reuse of software parts, in an attempt to manage the complexity of large systems [10,11].

The benefits of using a SOA approach are many. Unlike the case of traditional library reuse, which requires replication of code, SOA supports reuse of the services themselves, which provides a significant benefit in terms of having up-to-date components without concerns about maintenance of the code. In addition, it supports such a level of abstraction that multiple services can offer the same functionalities, potentially giving the developer a wider choice of providers from which to obtain the service needed. Furthermore, SOA establishes standardized contracts between endpoints, placing formal obligations between consumer and provider and largely increasing reusability and interoperability. An implementation that complies to known web service standards (e.g. REST or SOAP) has additional benefits, such as standardization, technology/platform neutrality and automatic discovery and use [10]. The automatic binding of services removes compile-time dependencies; the interface definition happens in runtime, removing the need to alter the code every time when there is a change in the service provider. This provides flexibility in the development and improves maintainability [12,13].

In the specific case of game development, a SOA approach can bring the potential benefit of decreased interdependencies and usage dependent payment models [14]. Furthermore, it facilitates dealing with scalability issues, which is particularly relevant to online games in which several thousands of players interact in a common platform, as the increased load in the servers may bring performance concerns [14]. SOA also makes it possible to access games from simple devices, eliminating the dependency on the quality of gaming hardware. In addition, providing pervasive gaming experiences becomes easier, as support for different platforms is highly simplified if the core of the gaming experience is provided via a service in a centralized server [15].

Educational applications in general – not only SGs – can also benefit from the application of SOA [16,17]. In addition to the points listed above, the most relevant advantage is the possibility to reuse educational components and domain-independent features (e.g. shared user profiles, knowledge databases on learning topics, natural language processing dialog services), which could potentially be deployed as web services that could be composed and invoked by a learning application or game when needed.

There are, nevertheless, challenges in adopting a service-oriented architecture. Quality assurance and testing module integration tends to be more difficult when developing SOA applications [9]. In addition, a service can be practically unusable if its interfaces lack clarity or are badly documented. Finally, extra attention has to be given to service descriptions, as they are the way to advertise the capabilities, interfaces, behavior and quality of a service, providing the required information for discovery, selection, binding and composition with other components [18].

### 3. Related work

Service-based architectures are already widely and successfully employed in several areas of software engineering, including game development. There is an increasing availability of service-based tools for game development, such as cloud-based infrastructure for building, deployment and distribution [19], platforms providing social connectivity to games [20] and services that provide generic gaming features such as achievements, leaderboards and cloud saving [21,22].

Although there are clear benefits in employing service-based architectures to SG development, the examples of deployments of SOA-based SGs are limited. This is true even if there are several gaming-related services already available and a large number of (non-educational) digital games already utilizing those services.

While not necessarily SOA-based, decoupling the content of the SG from the underlying gaming software is a way of facilitating the extensibility of SGs and to support domain experts in the creation of content, which can then happen independently of the development of the game itself [23]. The project *Travel in Europe (TiE)*, for example, proposes an architecture style that supports both code reuse and consistent interaction modalities across games [24–26]. The *MetaVals Serious Game*, a game for practicing basic finance concepts, consists of a modular database and an independent graphic interface, with a management interface that facilitates configuring the game to different contexts [27].

Authoring platforms also aim to reduce the complexity of game development. The *eAdventure* game platform serves as an authoring platform for educational point-and-click adventure games, executing games defined in a specialized markup language [28,29]. The authoring tool *Puzzle-it* divides the process of developing games into content authoring and core engine development, making it possible for instructors to create content for the games via the authoring tool without needing to be concerned about engine behind the games [30].

When it comes to the actual usage of SOA in SGs, examples available are very few.

While the game itself has not been developed, a Service-Oriented Architecture was the approach of choice for an envisioned gaming platform based on mobile augmented reality, called *MARL*. In this system, on-demand location-based instruction would be delivered through a head-mount display by a virtual instructor. The complete *MARL* game service would be composed of subsystems that would provide visual, human computer interface, and training services, allowing for the lower level objects to be encapsulated by the higher level interfaces, making it easier for improvements in the algorithms to be incorporated into the service [31].

The *Rashi Intelligent Tutoring System* teaches human anatomy through a problem-based environment. *Rashi* is built as a web service architecture that supports on-demand requests for small chunks of specific knowledge, instead of requests for an entire case specification at once, giving developers flexibility to develop lightweight inquiry tutors that run efficiently over the web [32]. On top of the same existing service structure for the original (2D) inquiry system, the researchers built a 3D game in which the student is a doctor who must diagnose a patient in a virtual hospital. Despite being limited to a specific type of learning (i.e. problem-based inquiry), *Rashi* constitutes one of the very few examples of service-based architecture for game-based education, demonstrating well the benefits of the SOA approach for SG development.

The Serious Games Society has developed the *Serious Games Web Services Catalog* [33], a repository of web services with the associated documentation and example applications. The catalog does not host any of the services, but rather acts as a showcase for services available from the providers, serving as a central hub for publishing and discovery of the currently available technologies. The effort aims to facilitate the communication between Serious Game developers among themselves and with educators, encouraging the application of the SOA approach to SG development.

#### **4. Implementing competence-based adaptation in Serious Games**

In order to investigate the development of SOA-based SGs, we have focused on competence-based adaptation in the learning environment, which is a highly relevant pedagogic feature [5,34], not yet widely employed in the SGs domain. Implementing adaptation in SGs, especially in low to medium scale projects, is costly, both in terms of conceptual and technological complexity as well as in design and implementation efforts. Therefore, efforts should be made to make it easier and cheaper equipping SGs with features to enable intelligent adaptation techniques for learning. Thus, a benefit is expected by applying a SOA approach, as it would allow using a single user profiling module supporting adaptivity for different SGs.

##### *4.1 Adaptive Serious Games*

Adaptivity in the context of learning and education refers to the adjustment of the instruction according to the learner's responses and performance, in an attempt to achieve superior performance of tailored tutoring when compared to regular teaching [35]. In game-based learning, equipping SGs with adaptation capabilities aims to create more effective games by providing an adequate level of challenge to the player, neither too hard, causing frustration, nor too easy, causing boredom [5].

Experimental findings have shown that equipping SGs with adaptive features results in superior educational gains and gaming experience [36]. The same results have been observed by a meta-review of more than 300 scientific articles on the educational efficacy of computer games: the vast majority (90%) of the games that reported non-trivial educational results displayed some form of educational adaptation or personalization [37].

In-game adaptation for learning can be different than the style of adaptation implemented in traditional virtual learning environments. Some authors propose the implementation of a subtle style of in-game adaptation, based on assessing the continuous interaction of the learner with the game, using observable evidences (i.e. the process of play itself) to infer knowledge, skills or other attributes. The objective is to not disrupt the game flow with tests or interventions that are not smoothly embedded in the game [38,39].

##### *4.2 Competence-based Knowledge Space Theory (CbKST)*

The Competence-based Knowledge Space Theory (CbKST) is an approach to formative, competence-centered assessment that comes from the non-numerical and non-linear approach of the Knowledge Space Theory (KST) [8]. The idea of CbKST is to

assume a finite set of roughly atomic competences – that is, well-defined, small scale descriptions of aptitude, ability, knowledge, or skill – and a prerequisite relation between those competences, which defines the competence model of the domain.

Due to the prerequisite relations, not all subsets of competences are possible competence states. This structure is an advantage from the computational point of view, and it also discriminates different learning paths for moving from having no competences to the state of having all possible competences. In accordance, a person's level of knowledge, ability or proficiency is described, at least theoretically, by exactly one competence state.

The structural model of the theory focuses on unobservable competences, making hypotheses about the brain's black box. By utilizing interpretation and representation functions, these unobservable competences (or, in other words, what is "in the brain") are mapped to evidences or indicators, relevant for a given domain. Such indicators can be all sorts of performance or behavior, and not only test items. The interpretation function ( $p$ , in Figure 1) assigns a set of competences required to solve a task to each of the indicators. Conversely, by utilizing a representation function ( $q$ ), a set of indicators is assigned to each competence state. This assignment induces a performance structure, which is the collection of all possible performance states. Due to these functions, unobservable competences and observable performance can be linked in a broad form where no one-to-one correspondence is required. This means that an entire series of indicators can be linked to underlying competence states.

CbKST accounts for the fact that indicators such as test items cannot be perfect evidence for the latent knowledge or ability. There is always the possibility that a person exhibits a correct behavior or activity just by chance. In turn, a person might fail in a test item although the necessary knowledge or ability is actually available. Because of this, CbKST considers indicators on a probability-based level: mastering a test item suggests having the underlying competences with a certain probability. Conceptually, this view constitutes a probability distribution over the competence structure.

The CbKST service, which is based on the theory described above, provides a software for formative assessment methods that can be integrated into learning activity sequences [40]. It has been applied, for example, to enable competence assessment in a web-based system for medical training [41], and also in the scope of the PROJECT03 and PROJECT04. It is part of the PROJECT01, specifically the PROJECT02. Details of its architecture, implementation and usage are described by Nussbaumer, Gütl and Albert [42]. The CbKST service is available in form of the PACKAGE01 educational software packages provided by the PROJECT03. The PACKAGE02 services of the PROJECT04 project, which were used for this study, are research prototypes and are available online [URL removed for anonymization purposes].

In the next section, we describe an adaptive serious game that has incorporated the CbKST services by using a SOA-based approach.

## 5. The Journey

*The Journey* is a serious game to teach basic concepts of probability theory to high school and entry-level university students, developed at RESEARCHGROUP01 as a

prototype implementation of a service-based adaptive SG, employing the CbKST service to implement basic adaptation features for learning. The objective of this development was to create a proof of concept that can be further developed later. In the game, the player represents the head of a group of hikers who wants to reach the top of a mountain chain. There are many ways to get there, so it is important to choose smartly to get to the end of the journey as fast as possible. Players have to understand how to calculate the probabilities of events that are related to the journey, and also use their knowledge to make the best possible decisions along the way.

When the player starts the game, she is first introduced to the story and the objective of the game. Next, the player reaches a point called a crossroad, in which she must choose between two paths, each with different probabilities of success and different lengths. The player has to calculate the probabilities for each way (Figure 2), and based on the results, she has to choose which way to go. If she calculates the probabilities correctly, the system decides, following the probability distribution of that path, if the group managed to go through the path or not. If the group was lucky enough, they move forward to the next crossroad; if not, they have to go back and try the other path, losing time. However, if the player calculates one or both probabilities wrongly, she is not able to proceed. In that case, the game shows her the correct answers and takes her back to the crossroad, presenting a new set of tasks. The process repeats until the end of the game.

There are two goals in the game: one external, which is communicated to the player, and one internal. The external goal is reaching the end of the journey as fast as possible. The internal goal is acquiring all the competences in the competence model (see section 5.1), and this is the goal to be reached for the game to end. The final score depends on how “fast” the player reaches the end (in the game’s internal time measure; real world time is not taken into consideration) and on the performance of the player in the tasks.

The prototype has been subjected to an early user test with 10 participants, aiming to assess the game’s pedagogical value and usability. The results of the evaluation of the learning effect were encouraging. In addition, no performance issues were found; minor usability problems were identified and will be fixed in future developments.

### 5.1 *Learning and assessment*

The competence model of the game is depicted in Table 1. For each competence in the model there is a set of predefined tasks. See below one example question, which refers to the competence “Probability space”. The values inside the curly brackets are generated by the game.

Up to now, {n} people tried to take this path, but only {x} of them managed to get through. Based on this sample, what is the estimated probability of arriving at your destination through this path?

The game provides guidance via the implementation of hints, offered to the player after 3 minutes of inactivity when in the Task screen. The game shows a graphical representation of the probability space and an explanation about how to apply the formula. If the player answers the task incorrectly, the game informs so and offers the correct answer and its explanation. The representation of the competence model itself is not exposed to the player.



The assessment in the game is based on the player's answers to the tasks, providing the features of an adaptive learning environment. The answers are forwarded to the adaptive assessment module, which holds the domain model and a representation of the competence state of the player. The service uses the player's answers to the tasks to update a model of the player's competences, and uses this information to send back to the game the ID of the next tasks that are appropriate to the user's level. The tasks are chosen randomly from that set of tasks for the competence.

The game ends, that is, the group reaches the top of the mountain, once the player has acquired all five competences.

## 5.2 Implementation architecture

*The Journey* has been developed following a Service Oriented Architecture (SOA) approach. It was designed to run locally in a machine with a working internet connection, as the connection with the adaptation service is implemented via network calls.

The game itself is a Flash Desktop Application running on Adobe Air. It is connected to a local SQLite database, which holds the game's tasks. The game uses Starling, an ActionScript 3 2D framework [43], for the graphical interface.

The CbKST service is a REST-based web service. REST services have generally a better performance than SOAP (the other popular protocol for the implementation of web services) due to their less verbose messages and because there is no need for message wrapping and serializations [44]. The messages exchanged with the service are basic XML files that contain very small amount of data (in the case of *The Journey*, typically no more than 100 bytes), resulting in interaction time between the game and the service in the range of 50-150 milliseconds per request in typical usage conditions, and of 150-900 milliseconds when simulating a 9.6 kbps connection.

The architecture features a very marked separation between game and the adaptation service, with the game's tasks and behaviors completely independent from the service. The game's interface with the service is well defined and has a limited scope: the service receives information regarding task completion only, and returns the list of competences acquired by the player and the suggestion for the next task. The service has no knowledge of the tasks themselves, except for the relationship between the tasks IDs and the competences in the model. In addition, the service does not hold any profile information about the players other than the learner ID, informed by the game when opening a learning session.

The game makes synchronous calls to service, which means that the game waits for the server's answer before continuing.

A simple, non-adaptive task sequencer class is included in the game, which can work as a back-up solution whenever the CbKST service is not available, making the game playable also offline. In this case, the sequence of tasks presented to the player is defined in a much simpler way: when the player responds correctly to a task, the back-up task sequencer assumes that the competence has been acquired and suggests, as next challenge, a task from the next competence in the competence model.

Currently, as it is a prototype with limited availability, the CbKST service is not registered in a service broker and can only be accessed by static binding, i.e. the service

interface and location have to be previously known and accessed directly. The service also has to be configured in advance with an XML representation of the learning domain competence model.

A typical interaction between game and service would be as represented in the Business Process Model Notation (BPMN) diagram in Figure 3:

1. The player starts the game. The game requests the service to open a game adaptation session.
2. The service initializes the session, with default values for each competence.
3. The player reaches a decision point (crossroad). The game asks the service for two tasks (or problems).
4. The service analyzes the current state of the competences of the player and suggests the IDs of two tasks.
5. The game pulls the tasks from the database and presents them to the player. The game sends the results of the tasks back to service.
6. The service updates the competence model of the player accordingly.
7. The player chooses among the two possible paths. The game decides if the player managed to progress in the chosen path or not and returns the result.
8. The game sends the player to the next crossroad and the process repeats.
9. When the player has achieved all the competences in the domain model, the service returns the message that no further tasks need to be performed. Game ends and the game adaptation session is closed.

Figure 4 depicts the whole system's service architecture using SoaML notation, in which it is possible to see how the components of the architecture work together. In the figure, there are two participants ("The Journey" and "CbKST service") and two services contracts that connect them ("Manage game adaptation session" and "Assess competences"). The connections between them represent their roles in that contract: "CbKST service" is the session manager and the assessment provider, while "The Journey" is the session requester and assessment requester.

The Participant diagram (Figure 5) represents the inner architecture of the participant "The Journey". The main Game Controller class is connected to the Service Connector class (highlighted), to the Database Connector and to the Internal Task Sequencer. The Service Connector implements the interfaces defined in the service contracts and makes the calls to the service that manages the game adaptation session and performs the competence assessment. The Database Connector accesses the local repository of game tasks. When a connection to the service is not available, the Game Controller makes the assessment requests to the Internal Task Sequencer class, which, as explained earlier, performs a simplified evaluation of the player's competences and assumes the role to tell the Game Controller which task sequence to show to the player. Finally, the Game Controller also accesses a package containing the Screens of the game; this package depends on a local copy of the Starling framework.

The CbKST service is a black-box component of which only the interfaces are known to a SG developer. These two interfaces and the methods that the service provides are depicted in the Service Interface diagram (Figure 6). Alternative implementations could

replace the CbKST service, as long as they implement the same interfaces established in the service contract.

The source code of the game is available online at [http://bit.ly/thejourney\\_code](http://bit.ly/thejourney_code).

## 6. Evaluation

We have conducted an Architecture Tradeoff Analysis Method (ATAM) analysis in order to identify possible points for improvements, from the architectural point of view. ATAM is a scenario-based method for assessing the quality of system architectures, which provides a structured way to identify risks, sensitivity points, and tradeoff points in the system. It was chosen for its ability to assess if the architecture of the system is in fact capable of delivering the desired quality attributes and fulfilling the software's business goals. In addition, ATAM is especially suited for evaluating systems in relatively early design stages and prototypes, serving as a valuable way for discovering weak spots that can be addressed in subsequent iterations of the development [45]. ATAM has been successfully used before to evaluate a game-based architecture, providing useful information to the developers [46].

The ATAM analysis is typically carried out in two phases. The first phase is architect-centric and focuses on describing and analyzing architectural documents, using scenarios to help determine if the architecture meets the desired functional requirements. The second phase congregates a more diverse group of stakeholders, including non-technical participants, to discuss and verify the results of the first phase [47].

This section summarizes the results of the ATAM analyses carried out with the prototype version of *The Journey* presented in this paper. The analysis focused mainly on the first phase (architect-centric), due to the small number of stakeholders involved.

### 6.1 Business drivers, quality attributes and scenarios

The main business drivers of the development of *The Journey* following a SOA paradigm were: (1) to be able to reuse the algorithm for in-game adaptive features for learning, in order to take advantage of existing services to make SG development cheaper, faster and more efficient; (2) to give freedom to the SG designer, providing the ability to easily incorporate various kinds of tasks in the game (e.g. questions and answers, item collection, dialog with characters) while maintaining in-game adaptation; (3) to have the possibility of supporting in the future the use of shared user profiles across different games or learning tools.

Based on the business objectives above, three main quality attributes were classified as high priority: performance during runtime, availability and modifiability. Table 2 represents the scenarios used to evaluate each of these attributes.

### 6.2 Risks, non-risks, sensitivity points and tradeoff points

From the direct comparison between the quality attributes and scenarios listed in Section 6.1 and the architectural approaches described in 5.2, a number of risks, non-risks,

sensitivity points and tradeoff points were identified and grouped according to their general themes.

From the viewpoint of the game performance, the analysis indicated that network latency problems are unlikely to happen due to the small size of the XML messages exchanged, and response times obtained in performance tests are low enough not to disturb the game flow. It has been added that, even so, the game could implement workarounds to minimize the risk even more, by using animations or adding other activities relevant to the game while it is waiting for the response from the server. The use of synchronous calls, although carrying the risk of delaying the game while waiting for the service response, increases the system reliability, as it ensures that the in-game adaptation behaves as expected.

It has been considered that the availability of the game is satisfactorily addressed, as the game has an internal task sequencer that is able to substitute the CbKST service when the service or a network connection is not available. There is still an associated risk, given that, if the connection fails permanently in the middle of the game, the current system has no way of reacting and switching to the internal module on the fly. This scenario has been considered unlikely; nevertheless, future versions of the software should address the risk by making it possible to switch between the two task sequencers in a dynamic and reliable way.

The modifiability of the system, particularly the possibility for future expansions, has also been positively assessed, although in this attribute there are some tradeoff points that need to be taken into consideration. The competence-based assessment is a robust, well-grounded and scientifically sound method for evaluating the player's performance, which is seen as sufficient for a wide variety of SGs. With the simple, limited scope interface offered by the CbKST service, a lot of flexibility is given to the game developer – it would even be possible to delegate the assessment to another service outside of the game, e.g. following an evaluation framework structure suggested by Serrano et al [48]. However, before including other types of task in the game, it is necessary to adapt their behavior to fit the atomic-competences model supported by the CbKST, e.g. by including a “translator” module able to connect the new types of tasks to the existing competence-based task assessment model. In addition, other types of assessment and adaptation (e.g. assessment of entertainment or of emotional states) are not yet supported and would require extra care to ensure that the different adaptation services would work well and consistently when combined.

The possibility of sharing the same user profile across different games or other learning tools may be obtained through a third component, a user profile manager service, that would ensure that different tools could access the same user profile without the risk of unpredictable results when two or more learning tools try to update the same profile at once.

Finally, a temporary risk identified is the fact that the binding between the game and the service is static due to the limited availability of the prototypes. Consequently, any changes in configuration have to be made manually. Furthermore, it is not currently possible to choose alternative services at runtime. This is, however, a temporary limitation that will be fixed in future developments.

## 7. Discussion

Using our experience in the development of *The Journey* and the subsequent ATAM evaluation of the architecture, we argue that the application of SOA in the development of serious games can result in shorter development times, and more focus and flexibility in the development. While the conclusions drawn in the ATAM analysis are directed to this specific case study, in this discussion we extrapolate these conclusions to consider also the applicability of SOA to SG development in general.

In the ATAM analysis, we confirmed the ability of the service-based architectural approach to fulfill the desired objectives (i.e. business goals), namely modifiability, availability and performance, in addition to modularity and flexibility in SG development, as already discussed in section 6.2.

Regarding the objectives that concern quality attributes related to the performance of *The Journey*, we concluded that the current implementation of the game is able to cope with the demands in typical usage situations. Nevertheless, the ATAM analysis helped identify possible improvements that would increase the overall quality and robustness of the system, especially related to the performance of the game in case of unexpected network/service unavailability, failure or slowdowns.

The ATAM analysis also allowed us to identify sensitivity points and tradeoffs that deserve extra attention. The issues identified in the analysis were particularly significant for the scenarios of future development of more complex SGs (i.e. games that incorporate different types of tasks or that adapt to other characteristics or states of the player) or richer learning environments that incorporate shared user profiles. In these cases, the current architecture is a good starting point for the development, but these environments would still require a non-trivial amount of work to ensure the smooth functioning of all the components.

Concerning general aspects of the use of SOA in SG development, we consider that the most immediate benefit is the possibility to reuse services, as it directly impacts the development times and the ability of the development team to focus their efforts in other aspects of game development (graphics, questions, game flow, etc.). In the case of *The Journey*, we could utilize the CbKST service – the result of intense research and development work of experts in the field of adaptation technologies – with a relatively small impact in development time.

In addition to the specific issues identified during the ATAM analysis, a few other general considerations must be taken into account when applying the SOA approach for SG design. First of all, the programming paradigm change requires adaptations in the style of development. Testing becomes even more crucial and at the same time more complicated, as the binding of services can result in unexpected results, causing errors and bugs that are harder to track.

The quality of documentation is even more important in a SOA approach than in traditional software engineering methods, as SOA assumes little or no communication between development teams, relying completely in the definition of the interfaces for the integration between the pieces of software. This is the reason why efforts such as the Serious Games Services Catalog are so valuable, so that documentation and examples of code are easily accessible to the developers.

Another issue with the SOA approach is the usability of the service itself, especially in cases where the service must be configured in advance. Often such customizations are made via configuration files, and in this case, the lack of administration user interfaces may prove to be problematic to the use of the service by third parties. Once again, good documentation is crucial.

## 8. Conclusions

In this paper we have discussed the benefits of adaptive features in SGs and defended the Service Oriented Architecture approach as a solution to make it easier for developers to enhance their SGs by incorporating existing services in their games with small overhead and added benefits. While there are many examples of games using services (e.g. casual games in social networks), in this work we focus on the use of services created specifically for educational SGs. We described the concepts, structure, software implementation and evaluation of *The Journey*, a serious game to teach probability to high school and entry-level university students, which employs the SOA approach and utilizes a service based on the Competence-based Knowledge Space Theory (CbKST) to implement in-game adaptation for learning. Unlike existing examples of SOA-based SGs (e.g. *Rashi Intelligent Tutoring System*), in which all components were built by the same team, the development process of the CbKST service was completely independent of that of *The Journey*. Furthermore, the adaptation features provided by the CbKST service are generic and can be utilized in a variety of learning settings, game-based or not.

Based on our experience, we argue that the SOA approach is able to provide SG developers with significant benefits, particularly increased flexibility and shorter development time. It is also relevant to mention that, by using service-based components, the developer has access to quality, always up-to-date components, created and maintained by experts in the field, and being exploited and tested by a variety of users. The SG developer may thus better focus on specific features and pedagogical aspects of the game itself.

Our work in *The Journey* is a prototype implementation of the SOA approach to SG development. We prioritized the deployment of adaptation for learning as it is an important and non-trivial feature for SGs, whose implementation from scratch would require prior knowledge on the theory of competence-centered assessment. We hope to have demonstrated how this complex theory can essentially be treated as a black box service that allows for easy addition of adaptive features in a SG.

The game's current implementation is a starting point that can be extended in several directions, exploiting the SOA paradigm. One possible development is the use of stealth assessment [8,39,49] to detect behaviors in the game, which can be mapped to competences in the competence domain. Another improvement involves exposing the competence model to teachers via an assessment interface or even to the players themselves, thus incorporating other pedagogical practices (e.g. self-reflection and formative feedback) to the existing game mechanics [50]. Moreover, other games could access the same competence model, as SOA enables sharing user profiles across different games. This paves the way to a richer learning environment, in which games can adapt according to achievements obtained in any

one of the connected learning technologies. Using the Architecture Tradeoff Analysis Method (ATAM) evaluation of the architecture, we have discussed how these and other future developments would affect the current architecture and which risks and sensitivity points would require extra attention.

Furthermore, currently only one feature of the game is deployed as a service. The SOA paradigm allows for the development of services implementing other generic, game- and domain-independent features such as user profiles, knowledge databases, dialog services, etc. Ultimately, the objective is to have a whole ecosystem of generic services that support cheaper and more efficient development of SGs, with dynamically bound services that would allow for much more flexibility in the development.

With this work, we hope to encourage a wider adoption of the SOA approach to SG development, as we believe that it makes it possible for developers to produce higher quality and feature rich games with relatively low effort.

## 9. Acknowledgements

[Removed for anonymity purposes].

## 10. References

- [1] T. Susi, M. Johannesson, P. Backlund, *Serious games - An overview*, Skövde, Sweden, 2007. <http://urn.kb.se/resolve?urn=urn:nbn:se:his:diva-1279>.
- [2] R. van Eck, *Digital Game-Based Learning: It's Not Just the Digital Natives Who Are Restless...*, *Educ. Rev.* 41 (2006) 1–16. doi:10.1145/950566.950596.
- [3] R. Blunt, *Do Serious Games Work? Results from Three Studies*, *eLearn.* 2009 (2009) 1. doi:10.1145/1661377.1661378.
- [4] F. Bellotti, R. Berta, A. De Gloria, *Designing Effective Serious Games: Opportunities and Challenges for Research*, *Int. J. Emerg. Technol. Learn.* 5 (2010) 22–35. doi:10.3991/ijet.v5s3.1500.
- [5] M.D. Kickmeier-rust, D. Albert, M.D. Kickmeier-rust, *Educationally Adaptive: Balancing Serious Games*, *Int. J. Comput. Sci. Sport.* 11 (2012) 1–10.
- [6] J.F. Knight, S. Carley, B. Tregunna, S. Jarvis, R. Smithies, S. de Freitas, et al., *Serious gaming technology in major incident triage training: a pragmatic controlled trial.*, *Resuscitation.* 81 (2010) 1175–9. doi:10.1016/j.resuscitation.2010.03.042.
- [7] V. Oliveira, A. Coelho, R. Guimarães, C. Rebelo, *Serious Game in Security: A Solution for Security Trainees*, *Procedia Comput. Sci.* 15 (2012) 274–282. doi:10.1016/j.procs.2012.10.079.
- [8] M.D. Kickmeier-Rust, D. Albert, *An Alien's Guide to Multi-Adaptive Educational Computer Games*, Informing Science Press, Santa Rosa, CA, 2012.

- [9] J. Hurwitz, R. Bloor, C. Baroudi, M. Kaufman, *Service Oriented Architecture For Dummies*, Wiley, 2007.
- [10] D. Sprott, L. Wilkes, Understanding service-oriented architecture, *Archit. J.* 1 (2004) 10–17.
- [11] W.M.P. Van Der Aalst, M. Beisiegel, K.M. Van Hee, D. König, C. Stahl, A SOA-based architecture framework, *Int. J. Bus. Process Integr. Manag.* 2 (2007) 91. doi:10.1504/IJBPIIM.2007.015132.
- [12] M. Stevens, Understanding Service-Oriented Architecture, *Developer.com*. (2005). [http://www.developer.com/design/article.php/10925\\_2207371\\_3/Understanding-Service-Oriented-Architecture.htm](http://www.developer.com/design/article.php/10925_2207371_3/Understanding-Service-Oriented-Architecture.htm) (accessed June 27, 2014).
- [13] T. Erl, *Service-Oriented Architecture: Concepts, Technology, and Design*, Prentice Hall PTR, Upper Saddle River, NJ, USA, 2005.
- [14] F. Arslan, Towards Service Oriented Architecture (SOA) for Massive Multiplayer Online Games (MMOG), in: *2012 UKSim 14th Int. Conf. Comput. Model. Simul., IEEE*, 2012: pp. 538–543. doi:10.1109/UKSim.2012.82.
- [15] M.M. Hassan, M.S. Hossain, A. Alamri, M.A. Hossain, M. Al-Qurishi, Y. Aldukhayyil, et al., A cloud-based serious games framework for obesity, in: *Proc. 1st ACM Multimed. Int. Work. Cloud-Based Multimed. Appl. Serv. E-Health (CMBAS-EH '12)*, ACM Press, New York, New York, USA, 2012: p. 15. doi:10.1145/2390906.2390912.
- [16] W. Chen, Web services - what do they mean to Web-based education?, *Int. Conf. Comput. Educ. 2002. Proceedings.* 1 (2002) 707–708. doi:10.1109/CIE.2002.1186051.
- [17] Z. Xu, Z. Yin, A.E. Saddik, A Web services oriented framework for dynamic e-learning systems, in: *Can. Conf. Electr. Comput. Eng. Towar. a Caring Hum. Technol. (CCECE 2003)*, IEEE, 2003: pp. 943–946. doi:10.1109/CCECE.2003.1226050.
- [18] M.P. Papazoglou, P. Traverso, S. Dustdar, F. Leymann, *Service-Oriented Computing: State of the Art and Research Challenges*, Computer (Long Beach, Calif). 40 (2007) 38–45. doi:10.1109/MC.2007.400.
- [19] Amazon Web Services, *AWS Game Development and Operation*, (2014). <http://aws.amazon.com/game-hosting/> (accessed July 22, 2014).
- [20] Facebook, *Games Developer Center*, (2014). <https://developers.facebook.com/docs/games/> (accessed October 20, 2014).
- [21] Lumos, *Lumos Powered*, (2014). <https://www.lumospowered.com> (accessed July 22, 2014).



- [22] G. Hartrell, Introducing Google Play game services, Google Dev. Blog. (2013). <http://googledevelopers.blogspot.com/2013/05/introducing-google-play-game-services.html> (accessed July 22, 2014).
- [23] F. Bellotti, R. Berta, A. De Gloria, L. Primavera, Adaptive Experience Engine for Serious Games, *IEEE Trans. Comput. Intell. AI Games*. 1 (2009) 264–280. doi:10.1109/TCIAIG.2009.2035923.
- [24] F. Bellotti, R. Berta, A. De Gloria, V. Zappi, Exploring gaming mechanisms to enhance knowledge acquisition in virtual worlds, in: *Proc. 3rd Int. Conf. Digit. Interact. Media Entertain. Arts - DIMEA '08*, ACM Press, New York, New York, USA, 2008: p. 77. doi:10.1145/1413634.1413653.
- [25] F. Bellotti, R. Berta, A. De Gloria, L. Primavera, Supporting authors in the development of task-based learning in serious virtual worlds, *Br. J. Educ. Technol.* 41 (2010) 86–107. doi:10.1111/j.1467-8535.2009.01039.x.
- [26] F. Bellotti, R. Berta, A. De Gloria, A. D'Ursi, V. Fiore, A serious game model for cultural heritage, *J. Comput. Cult. Herit.* 5 (2012) 1–27. doi:10.1145/2399180.2399185.
- [27] M.M. Popescu, M. Romero, M. Usart, C.I. National, Using Serious Games in adult education Serious Business for Serious People-the MetaVals game case study, in: *Proc. 7th Int. Conf. Virtual Learn. - ICVL 2012*, 2011: pp. 68–72.
- [28] P. Moreno-Ger, J.L. Sierra, I. Martínez-Ortiz, B. Fernández-Manjón, A documental approach to adventure game development, *Sci. Comput. Program.* 67 (2007) 3–31. doi:10.1016/j.scico.2006.07.003.
- [29] J. Torrente, Á. del Blanco, P. Moreno-Ger, I. Martínez-Ortiz, B. Fernández-Manjón, Implementing accessibility in educational videogames with <e-Adventure>, in: *Proc. First ACM Int. Work. Multimed. Technol. Distance Learn. - MTDL '09*, ACM Press, New York, New York, USA, 2009: p. 57. doi:10.1145/1631111.1631122.
- [30] D. Pranantha, F. Bellotti, R. Berta, A. De Gloria, Puzzle-it: An HTML5 Serious Games Platform for Education, in: *7th Int. Conf. Edutainment 2012 3rd Int. Conf. GameDays*, Springer Berlin Heidelberg, Darmstadt, Germany, 2012: pp. 134–143. doi:10.1007/978-3-642-33466-5\_15.
- [31] J. Doswell, K. Harmeyer, Extending the `Serious Game' Boundary: Virtual Instructors in Mobile Mixed Reality Learning Games, in: *Proc. 3rd Digit. Games Res. Assoc. Int. Conf. - DiGRA 2007*, 2007: pp. 524–529. doi:10.1.1.190.2204.
- [32] M. Floryan, B.P. Woolf, Optimizing the Performance of Educational Web Services, in: *2011 IEEE 11th Int. Conf. Adv. Learn. Technol., IEEE*, 2011: pp. 399–400. doi:10.1109/ICALT.2011.126.
- [33] Serious Games Society, Serious Games Web Services Catalog, (2013). <http://services.seriousgamessociety.org> (accessed April 04, 2014).

- [34] V.J. Shute, Stealth assessment in computer-based games to support learning, in: S. Tobias, J.D. Fletcher (Eds.), *Comput. Games Instr., Information Age Publishers*, Charlotte, NC, 2011: pp. 503–524.
- [35] B.S. Bloom, Sigma of Problem : The Methods Instruction One-to-One Tutoring, *Educ. Res.* 13 (1984) 4–16.
- [36] M.D. Kickmeier-Rust, B. Marte, S. Linek, T. Lalonde, D. Albert, The effects of individualized feedback in digital educational games, in: *Proc. 2nd Eur. Conf. Games Based Learn.*, 2008: pp. 227–236.
- [37] M. Kickmeier-Rust, S. Göbel, D. Albert, 80Days: Melding Adaptive Educational Technology and Adaptive and Interactive Storytelling in Digital Educational Games, in: R. Klamma, N. Sharda, B. Fernández-Manjón, H. Kosch, M. Spaniol (Eds.), *Proc. First Int. Work. Story-Telling Educ. Games, CEUR Workshop Proceedings*, 2008.
- [38] C.M. Steiner, M.D. Kickmeier-Rust, E. Mattheiss, S. Göbel, D. Albert, Balancing on a High Wire: Adaptivity Key Factor of Future Learning Games, in: *An Alien's Guid. to Multi-Adaptive Educ. Comput. Games, Informing Science*, 2012.
- [39] V.J. Shute, F. Ke, Games, Learning, and Assessment, in: D. Ifenthaler, D. Eseryel, X. Ge (Eds.), *Assess. Game-Based Learn.*, Springer New York, 2012: pp. 43–58. doi:10.1007/978-1-4614-3546-4\_4.
- [40] A. Nussbaumer, C. Hockemeyer, A. E-learning, E. Platform, K. Space, W. Service, et al., A Generic Solution Approach for Integrating Adaptivity into Web-based E-Learning Platforms, in: *Proc. Int. Conf. Interact. Mob. Comput. Aided Learn. (IMCL 2007)*, Amman, Jordan, 2007: pp. 1–9.
- [41] C. Hockemeyer, A. Nussbaumer, E. Lövquist, A. Aboulafia, D. Breen, G. Shorten, et al., Applying a Web and Simulation-Based System for Adaptive Competence Assessment of Spinal Anaesthesia, in: M. Spaniol, Q. Li, R. Klamma, R.W.H. Lau (Eds.), *Adv. Web Based Learn. – ICWL 2009*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009: pp. 182–191. doi:10.1007/978-3-642-03426-8.
- [42] A. Nussbaumer, C. Gütl, D. Albert, Towards a Web Service for Competence-based Learning and Testing, in: *Proc. World Conf. Educ. Multime- Dia, Hypermedia Telecommun. (ED-MEDIA 2007)*, Vancouver, Canada, 2007.
- [43] Gamua, Starling Framework, (2013). <http://gamua.com/starling/> (accessed December 16, 2013).
- [44] P. Bianco, R. Kotermanski, P. Merson, Evaluating a Service-Oriented Architecture (CMU/SEI-2007-TR-015), 2007.
- [45] R. Kazman, M. Klein, P. Clements, ATAM: Method for architecture evaluation, 2000. <http://www.sei.cmu.edu/reports/00tr004.pdf> (accessed July 03, 2014).
- [46] A. BinSubaih, S. Maddock, Using ATAM to evaluate a game-based architecture, in: *Proc. 2nd Int. ECOOP Work. Archit. Evol. (ACE '06)*, Nantes, France, 2006.

- [47] L.G. Jones, A.J. Lattanze, Using the Architecture Tradeoff Analysis Method to Evaluate a Wargame Simulation System: A Case Study (CMU/SEI-2001-TN-022, ADA399795), Pittsburgh, PA, USA, 2001.  
<http://www.sei.cmu.edu/publications/documents/01.reports/01tn022.html>.
- [48] A. Serrano, E.J. Marchiori, A. del Blanco, J. Torrente, B. Fernandez-Manjon, A framework to improve evaluation in educational games, in: Proc. 2012 IEEE Glob. Eng. Educ. Conf., IEEE, Marrakech, 2012: pp. 1–8.  
doi:10.1109/EDUCON.2012.6201154.
- [49] F. Bellotti, B. Kapralos, K. Lee, P. Moreno-Ger, R. Berta, Assessment in and of Serious Games: An Overview, *Adv. Human-Computer Interact.* 2013 (2013) 1–11.  
doi:10.1155/2013/136864.
- [50] S. Arnab, T. Lim, M.B. Carvalho, F. Bellotti, S. de Freitas, S. Louchart, et al., Mapping Learning and Game Mechanics for Serious Games Analysis, *Br. J. Educ. Technol.* (2014). doi:10.1111/bjet.12113.

### Figure captions

Figure 1: The CbKST model makes inferences about the brain's competence state by means of observable evidences.

Figure 2. The Task screen of *The Journey*.

Figure 3. Business Process Model Notation (BPMN) diagram of *The Journey*.

Figure 4. A SoaML Service Architecture diagram representing how the game component and the CbKST service work together in the system architecture.

Figure 5. A SoaML Participant Diagram depicting the inner architecture of the game component.

Figure 6. A SoaML Service Interface diagram displaying the two main interfaces of the CbKST service that are used in the implementation of *The Journey*.

ACCEPTED MANUSCRIPT

**Table 1**

ID	Competence	Description	Dependencies
1	Probability space	The learner understands the relationship of a collection of events and their probability measure within a sample space (the representation of all possible outcomes of the events) with values ranging from 0 to 1 (where 0 means impossibility and 1 means certainty of an outcome). The learner is able to estimate the probability of an event from data of observed outcomes. The learner is able to derive that $P(\bar{A}) = 1 - P(A)$ .	-
2	Probability of mutually exclusive events	The learner understands the relationship between two mutually exclusive events in the same probability space, which can be represented by the formulas $P(A \cap B) = 0$ and $P(A \cup B) = P(A) + P(B)$	1
3	Probability of non-mutually exclusive events	The learner understands that when there are two non-mutually exclusive events in the same probability space ( $P(A \cap B) \neq 0$ ), they can be represented by the formulas $P(A \cup B) = P(A) + P(B) - P(A \cap B)$ .	1
4	Probability of independent events	The learner understands that the probability of two consecutive and independent events is calculated using the formula $P(A \cap B) = P(A) \times P(B)$ .	1
5	Probability of dependent events	The learner understands that the probability of two consecutive events is calculated differently when the one of the events is dependent on the other. In that case, $P(A \cap B) = P(A) \times P(B   A)$ , and consequently, $P(B   A) = P(A \cap B) / P(A)$ .	2, 3, 4

Table 1. The competences targeted by *The Journey* and their dependencies.

**Table 2**

<b>Attributes</b>	<b>Scenarios</b>
<b>Performance</b>	The player's gaming experience is not affected by the software response times.
<b>Availability</b>	The game can be played in case an internet connection and/or the service is unavailable.
<b>Modifiability</b>	The game can be modified to include different task types (e.g. Q&A tasks, item collection, item sorting, dialog with characters, etc.).
	The game can be expanded to include other features (different input methods, teacher assessment interface, etc.).
	Related games can be created on the same learning domain.
	Related learning tools (games or otherwise) can be created, sharing the same user profile.

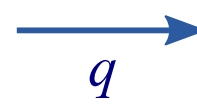
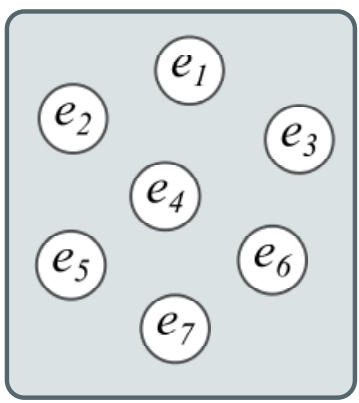
Table 2. The attributes and scenarios used during the ATAM evaluation of *The Journey*.

ACCEPTED MANUSCRIPT

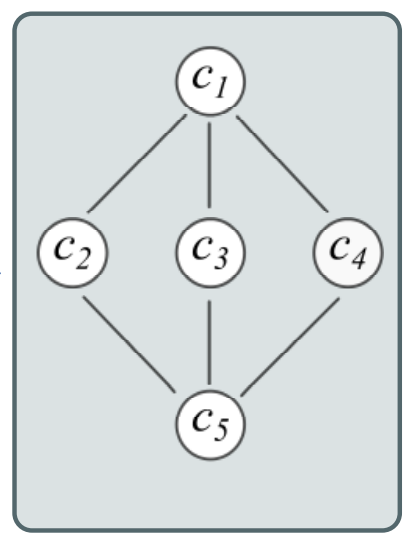
**Unobservable competences (brain's black box)**

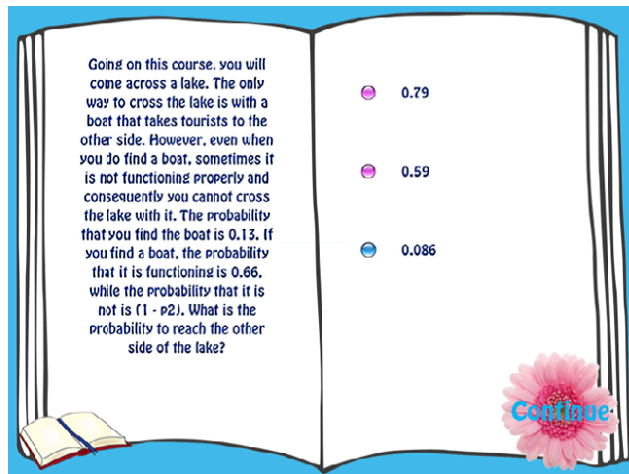


**Observable evidences**

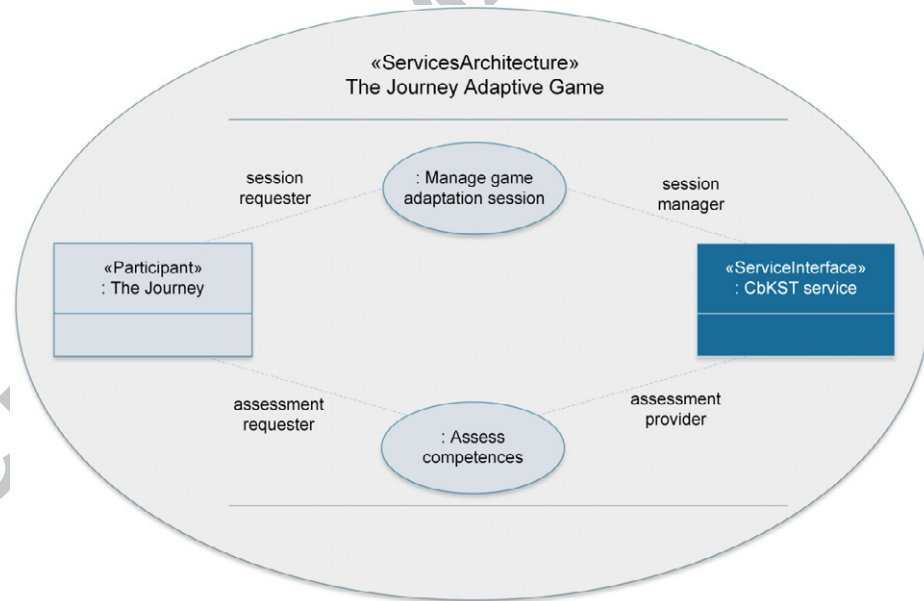
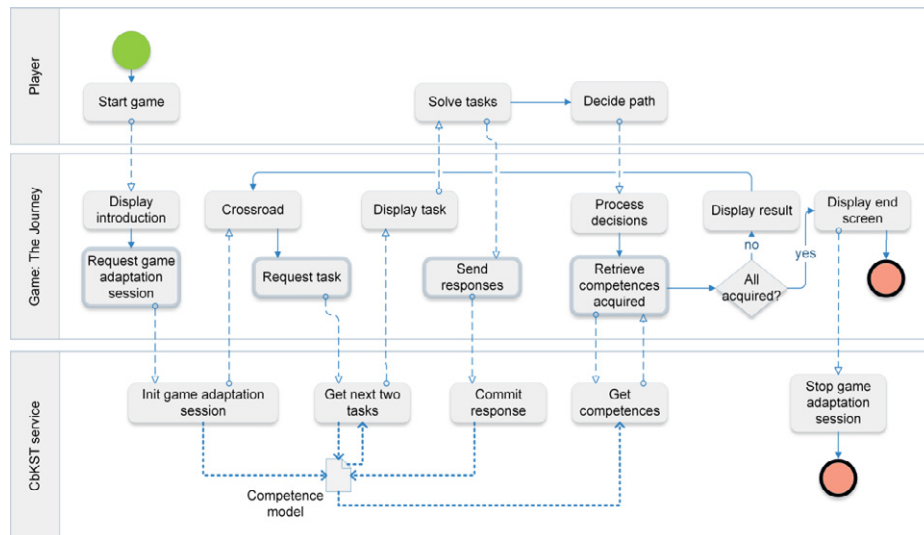


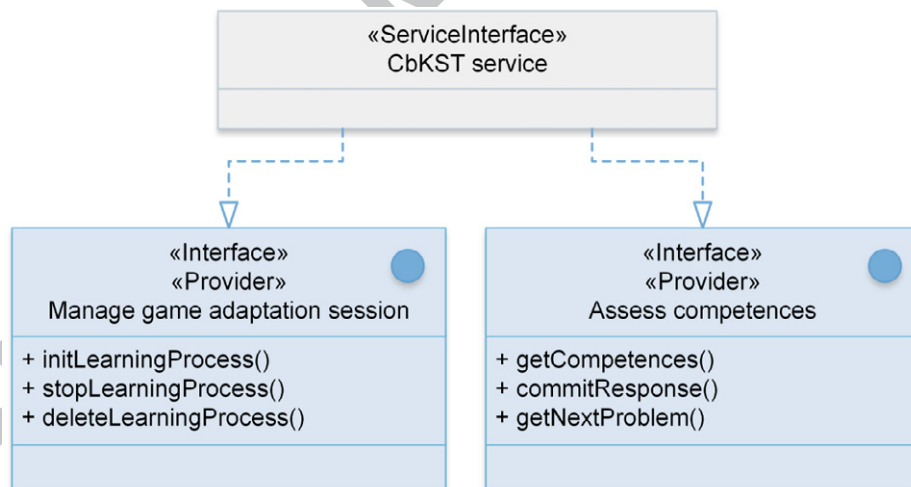
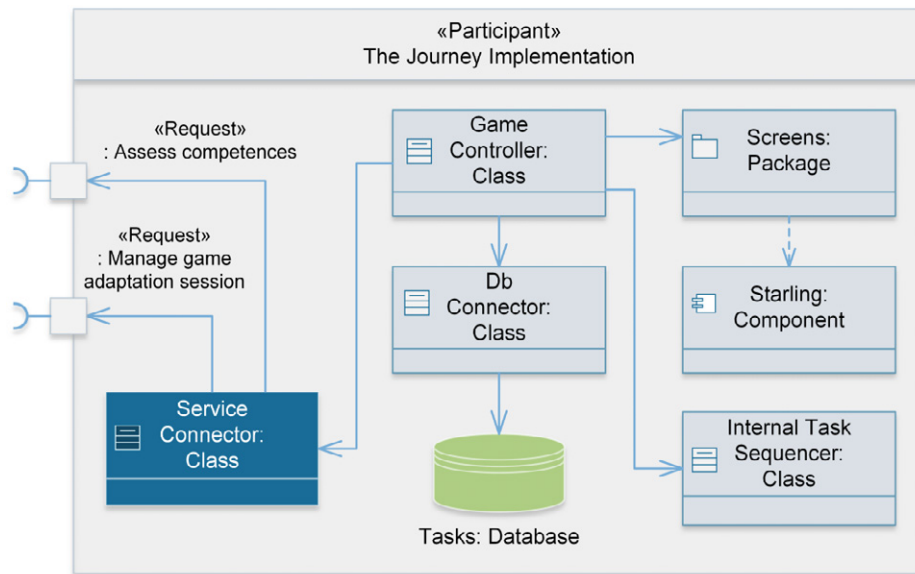
**Competence model**











## **11. Highlights:**

- We investigate the use of Service Oriented Architecture in Serious Games development
- We developed a prototype of a simple SOA-based adaptive Serious Game on probability
- We describe the game architecture and discuss benefits and drawbacks of the approach
- We argue that SOA can enhance SG development with small overhead
- Benefits include efficient development, cost, reliability and easier maintenance

ACCEPTED MANUSCRIPT