ATAM-RPG: A role-playing game to teach Architecture Trade-off Analysis Method (ATAM)

Claudia Hidalgo Montenegro
Departamento de Informática
Universidad Técnica Federico Santa María
Valparaíso, Chile
chidalgo@outlook.com

Hernán Astudillo
Departamento de Informática
Universidad Técnica Federico Santa María
Valparaíso, Chile
hernan.astudillo@usm.cl

María Clara Gómez Álvarez
Facultad de Ingenierías
Universidad de Medellín
Medellín, Colombia
mcgomez@udem.edu.co

Abstract— **Teaching software architecture to undergraduate students is particularly hard because they typically have no experience with medium or large systems with competing stakeholders. A particularly hard case is ATAM (Architecture Trade-off Analysis Method), which allows the evaluation of architectural designs and quality attributes by competing stakeholders. This article describes ATAM-RPG, a role-playing game to support the teaching of ATAM by simulating stakeholder's interaction and trade-offs. The initial ATAM-RPG case incorporates the architecture, scenarios and design trade-offs of the Chilean national tsunami alert system (SNAM). The approach was tested by deploying the SNAM case in undergraduate courses; initial results show that ATAM-RPG was well-evaluated regarding trade-off description and understanding (and especially utility trees). Students also recognized the importance of exercising technically-based negotiation skills. We conclude that role playing games can be fruitfully used for software architecture education.**

Keywords— ATAM; software architecture; software engineering education; role playing games; architecture trade-off negotiation.

I. Introduction

Software architecture demands both technical and non-technical competences of software architects, especially trade-off detection and evaluation. Several methods have been proposed to evaluate, propose and negotiate quality attributes for a software system. ATAM (Architecture Trade-off Analysis Method) [1] is perhaps the best-known of these methods; it is a structured method with steps and techniques, designed to assess software architectures, and it supports the trade-off negotiation among stakeholders, who must reach agreement on priorities and scenarios.

Unfortunately, methods like ATAM are particularly hard to teach to students with little previous experience in developing medium and large scale systems, where they might have been exposed to legitimate competing stakeholder needs, and thus may be unaware that addressing actual trade-offs involves not just technical issues but also negotiating with other stakeholders. "Real world" large projects usually have stakeholders without a common authority which could force them to agree on issues, so they must negotiate not just the solution design, but frequently the problem definition itself.

This article describes ATAM-RPG, a role-playing game specifically designed to simulate an ATAM evaluation meeting, allowing students to practice both technical knowledge and social skills in the negotiation of software architecture. ATAM-RPG is parametric to an embedded case, and its current version includes the architecture, scenarios and roles of SNAM (Sistema Nacional de Alerta de Maremotos), the Chilean national tsunami warning system. The initial version of ATAM-RPG has been used in classes with two sets of practitioners undergoing certificate training, in Chile and Colombia.

The structure of this article is as follows: Section II describes the main concepts of this proposal like ATAM and role-playing games; Section III describes related work on games to teach software engineering; Section IV presents the problem description; Section V describes the solution approach ATAM-RPG; Section VI presents the SNAM case study; Section VII describes the ATAM-RPG case study results; and Section VIII summarizes and concludes.

II. Theoretical Framework

The main concepts underlying our proposal are: i) Software Arquitecture, ii) ATAM, and iii) Roles-playing games for teaching software engineering.

A. Software Architecture

The software architecture of a system describes its structure, which comprises software components and their relationships [2]. Garlan and Perry [3], [4] propose a broader definition that includes not only components and relationships, but also the principles and guides that control the design and evolution of the system over the time. In software architecture, a view represents one structural approach where the solution to one of several concerns is described [5]. Rozanski *et al.* [5]

978-15386-3057-0/17/$31.00©2017 IEEE

describe in detail how to define the architectural views. A brief description of the views is presented below:

- Context view: describe the interactions, relationships, responsibilities and dependencies between the system and external entities, people and systems.
- Functional view: describe functional capabilities, external interfaces, internal structure, and define functional design approach. This view allows specifying the functional elements at runtime.
- Deployment view: describe the physical runtime environment, the technical requirements of the nodes and the mapping of the software elements to such environment.

B. *ATAM*

Software architecture evaluation techniques provide support for decision making in early phases of development, and enable the creation of specifications and predictions as to whether a proposal meets the required architectural qualities. The literature records several techniques for software architecture evaluation [6], including SAAM (Software Architecture Analysis Method), ATAM (Architecture Trade-off Analysis Method), CBAM (Cost Benefit Analysis Method), ALMA (Architecture Level Modifiability Analysis), and FAAM (Family-Architecture Analysis Method).

Kazman *et al.* [7] present ATAM as a method to identify risks, since it can detect areas of concern in the architecture of complex software systems, because:

- It can be done early in the software lifecycle.
- It is relatively inexpensive and easy to perform, since it works with architectural design models.
- It produces analyses according to the level of detail of the architecture specifications.

ATAM [1] focuses on identifying how quality level goals are met, assessing trade-offs, and allowing stakeholders to prioritize quality attributes through a consensus-building activity involving project managers, architects, and project stakeholders. It has a 9-steps process shown in Figure 1 and described below:

1) Describe ATAM itself: the leader assessor presents the method to participants and define expectations.

2) Describe business drivers: the project manager presents the business drivers that will be primary on the project.

3) Present the proposed architecture: the architect describes the arquitecture emphasizing the accomplishment of business drivers.

4) Identify architectural approaches: the architect presents possibles architectures proposals.

5) Generate a utility tree describing quality attributes: the participants elicitate utility requirements of the system (performance, availability,security, usability, among others). These requirements will be specified by scenarios and prioritized.

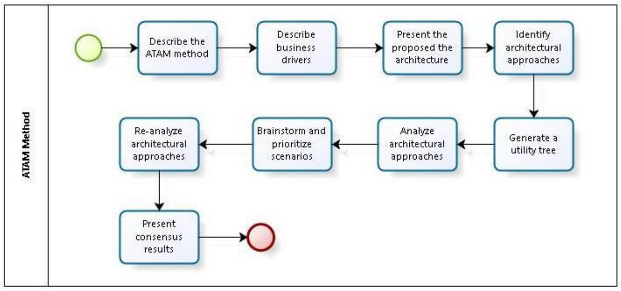

ATAM Method
Describe the ATAM method
Describe business drivers
Present the proposed the architecture
Identify architectural approaches
Generate a utility tree
Analyze architectural approaches
Brainstorm and prioritize scenarios
Re-analyze architectural approaches
Present consensus results

Figure 1. Steps of ATAM Method

6) Analyze architectural approaches: in this step the participants identify risks, high priority scenarios, and the results verification against stakeholders needs.

7) Brainstorm and prioritize scenarios: the stakeholders group elicitate a more complete set of scenarios. Such scenarios are prioritized by vote.

8) Re-analyze architectural approaches and priorities: the participants identify risks for prioritized scenarios (step 7) to identify sensibility and commitment points.

9) Present consensus results: in this step the results are presented to all the participants, specially to stakeholders.

These steps can be iterated in sessions along one or more days, meeting stakeholders until they finally agree on the architecture.

An important aspect in ATAM is the use of one level of quality characteristics (attributes) refinement, called utility tree, for prioritizing scenarios based on a particular quality characteristic. Such tree is a tool for promoting discussion and negotiation of the architectural qualities of the system in analysis.

C. *Roles- playing Games*

Roles-playing games allow students to participate in a simulated reality, using instructional material prepared specifically for the educational purpose of these games.

The basic elements of all role-playing games are:

- Roles to be interpreted by participants: each participant assumes a role in the problematic situation. In the case of software engineering such roles can be: project manager, architect, developer, tester, among others.
- Participants interaction: role-playing games should promote the interaction between participants to reach the proposed challenges.
- Story or situation for the action: The game should be framed by a specific situation.

In the case of role-playing video games, they must have a graphical interface, display the virtual place where the participants are, have tools to allow participants roles selection and interaction (chat, emotions, slate).

III. Related Work

There are several proposals for software engineering education using serious games or RPGs.

Hainey [8] proposes a ludic approach to teach elicitation and analysis of software requirements, using RPGs and case studies on paper; as we'll see later, our proposal integrates both methods into a software tool. This approach is similar to ATAM-RPG as both focus on teaching specific informatics areas, are based on case studies, and use multimedia elements to immerse in the case study simulation; but they differ in that [8] supports a single player interacting with non-player characters that follow fixed scripts, whereas ATAM-RPG allows multiple simultaneous players interacting with each other, and the teacher does not need to follow a fixed script but guidelines to conduct the simulation.

Rusu *et al*. [9] promote the learning of software engineering basic concepts with a 5-step game built by undergraduate students as a joint development project, and the creation of a computer game is used to show students in basic and secondary education the software life cycle and the basic tasks that software engineers carry, in a playful manner. It is like ATAM-RPG in its technical yet basic approach to teach informatics, but it differs in target group (single player versus multiple simultaneous players), use of RPGs for case studies, and support to teach ATAM.

Constantino *et al.* [10] use simulation-based games to teach specialized topics of operations management that allow decision-making regarding machines capabilities, maintenance strategies, or other production improvement decisions. It is similar to ATAM-RPG regarding use of a simulation game played in group and competition to achieve the best result in the activity, but it differs in that it is not directly linked to informatics but to engineering, does not use RPGs or case studies, and cannot be used to teach ATAM.

In this context, teaching students all the ATAM steps presented seems to be better through computer games than traditional expository classes. Sitzmann [11] shows that computer games enhance simulation-based learning and knowledge retention between 9% and 20% over traditional learning in education. Purarjomaldlangrudi and Ghapanchi [12] also conclude that serious games are effective and useful in teaching computer instrument and also note that most educational serious games mix rankings RPGs and simulation.

There are also some e-learning proposals to teach software architecture modeling. Calderón and Cedeño [13] present a teaching activity in two classes of software engineering, where students (8 groups in each class) were divided on the basis of a three-layer architecture; using UML sequence diagram representing changes to be performed on the architecture for a specific scenario, students were successful in modifying it yet maintain the three layers constraint.

Summarizing, there is a wide range of approaches and tools to teach social interaction methods, such as role playing, case studies, Socratic method, card games, and other group activity approaches; however, none includes social and technical components at once, which is essential to ATAM.

IV. Problem Description

There is growing interest in technology-supported education, especially for its ability to work for both local and remote audiences. These technologies include aspects of e-learning and b-learning (semi-remote education), and more recently m-learning (education through mobile devices) [14]. Tools usually found in educational settings are Educational Content Managers (LMS, Learning Management Systems), Virtual Learning Environments (VLE, Virtual Learning Managements), and Content Managers (CMS, Content Management Systems) [14].

Educational technologies have been deployed for general purposes, and there is significant recent work in using them to teach software engineering [14], but little is known of tools to teach software architecture. Undoubtedly, this partly stems from the fact that much informatics education is addressed to students with already developed technical background but with little experience in developing medium and large scale systems, making hard to convey the complexities of addressing simultaneously the technical, organizational and social aspects of developing large systems. Indeed, most software engineering education tools [14] aim to teach students how to collaborate (e.g. in a software development team) but not how to negotiate, much less using technical arguments and quality attributes to address external stakeholders.

ATAM (Architecture Trade-off Analysis Method) [1] is a well-known architecture evaluation method, which combines a high degree of social interaction and specialized negotiation techniques to allow stakeholders with competing (and relatively incompatible) goals to evaluate an architectural design using scenarios and attributes, and to negotiate a final design and valuation acceptable to all or most of them (and which then becomes the basis for further software development).

ATAM is especially hard to teach to students without previous socio-technical experience, unprepared to understand how competing stakeholder concerns will actually affect their own technical work, and without methodological tools to address these situations. This technically-grounded negotiation demands a practical understanding of both soft skills and technical knowledge, but most software engineering education tools in the literature are not meant to teach and exercise both skill sets simultaneously. ATAM is a clear example of a technique where practical application rather than expositive teaching are not just best, but actually required for effective learning.

V. Solution Approach: ATAM-RPG

ATAM-RPG is an educational game that combines role-playing and simulation. In this game each participant has a role (stakeholder, project manager, and architect) in an ATAM simulated meeting. For this meeting simulation, the application must support the ATAM method steps, and allows participants to view and modify architecture artifacts (like architecture models, utility trees, and scenarios). Finally, to enable teaching, the application must allows teachers or assistants to design cases that package story and roles.

The basic elements for ATAM-RPG, according to the role-playing games are:

- Roles to be interpreted by participants: there should be roles for all project stakeholders, with their own goals, possible opposite to those of other roles. Each student plays a role. Three additional roles are required for the ATAM mechanics itself (Moderator, Project Manager, and Architect), which should be taken by teachers, teaching assistants, or students with experience in software architecture.
- Participants interaction: The game should facilitate dialogue within the system, and establish a theme or objective that motivates participants to interact during the meeting.
- Story or situation for the action: the situation that has caused the meeting, i.e. some specific project for which an architecture is proposed and must be assessed.

The ATAM session simulator is a Web application, remotely accessible by teachers and students, built with PHP, JQuery, HTML5, and Flash.

Its key features are: (1) provide a graphical session simulation environment, (2) allow case designers to develop cases (including roles and initial artifacts, like architecture and utility tree), (3) provide tools for key ATAM artifacts, namely architecture modeling, utility trees, and scenario evaluation/priorization; (4) allow teachers to hold an activity, selecting (and possibly modifying) a case and assigning students to its roles, (5) allow students to play their assigned role in the simulated session, and (6) allow teachers to evaluate student performance according to learning goals.

In the next section we describe the ATAM-RPG game from two approaches:

- Key features or functionalities of the videogame
- Didactic elements of the videogame, like the learning goals and the procedure to prepare and apply the case study in the classroom.

A. *Key features or functionalities of the videogame*

The main features of the ATAM-RPG videogame are basically: (1) graphical design, (2) case development, (3) tools for ATAM techniques, and (4) student grading.

1) Graphical design: the graphical design is a fundamental aspect of a simulation. For the simulation of social interaction among the roles, participants should be able to see their virtual situation and choose how to express themselves. Not only they must see their characters in the conference room talking to each another, but they must be able to choose an emotion (seriousness, joy, or discomfort) since during negotiation of quality attributes there may be disagreements. Likewise, users should see the technical content of a session (architecture, utility tree, scenarios, or supporting material) as if was projected at the meeting room. All of this generates a sense of realism in virtual activity. Figure 2 shows screetshots of roles interaction in ATAM-RGP game.

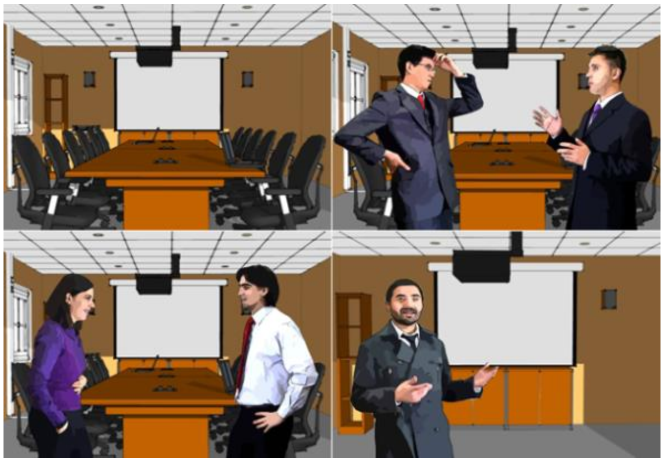


Figure 2. Role Play Environment Screenshots

2) Case Development: the case is decisive for the activity success, since it determines the "story" of the role-playing game: participants will know the system to be evaluated, the roles they can take, and their goals. Each project has its own set of stakeholders, defined in ATAM-RPG by the case designer; the standard ATAM roles (Moderator, Project Manager, and Architect) are also included. Each stakeholder role has a name, main goal (e.g. "reduce unofficial alert dissemination from 5% to 1%"), additional goals (e.g. "get consensus before deadline"), and a suggested phrase (e.g. "I need to be the only one with access to the alert bulletin").

3) Tools for ATAM techniques: a sidebar supports ATAM techniques and artifacts, like architecture models, utility trees, scenarios, and iterations to improve them. Teacher and students can see at all time the "initial state" of the case (including the initial architecture and utility tree) and the "current proposed solution" (including modified architecture views and prioritized utility tree).

4) Student grading: at the end of the activity, the teacher evaluates groups and individual students, reflecting: (1) if agreement could be achieved in the set time, and (2) how each student achieved their role's goals.

B. *Didactic elements of the videogame*

In this section we present some didactic elements of the videogame like the learning goals and the procedures for prepare and apply the case study in the classroom:

1) Learning goals: at the end of the game students should be able to:

- Identify the existence of conflicting goals among stakeholders and negotiation and priorization tecnhiques for obtain agreements.
- Recognize the importance of scenarios for system quality attributes like a technique for negotiation with stakeholders.
- Improve the understanding of all the steps of the ATAM method through its application in a real case study.

2) *Procedure for case study design:* the teacher should follow these steps to prepare the case study before apply the ATAM-RPG:

- Describe the case: the detailed case description was written using a viewpoint commonly used in role play: “what would you do if you were X”. It is recommended to use a real situation to increase the motivation of students to participate in the activity.
- Describe the roles: in the case study roles must correspond to actual stakeholders identified as relevant system actors with their own goals.
- Describe the goals for each role: it is necessary to establish a set of goals to achieve for each role. This step is very important because the goals drive the negotiation process between competing stakeholders. Additionally, the teacher should define allies and opponents and ensure the existence of conflicting goals.
- Pre-design dialogues by role: in the case study should be defined dialogues by role for each ATAM step. These dialogues can be modified by the participants and add flexibility to the game, otherwise the negotiation process would take longer.
- Define initial architecture design: the teacher must upload in ATAM-RPG three architectural views (contextual, functional and deployment views) for the system described in the case study. Such views should be consistent with each other and previously revised by the participant who assumes the role of architect. Such views could be modified during the game application.
- Define initial utility tree: the teacher should design an initial utility tree take into account the stakeholders' goals. Such tree will be modified by the participants through the identification of additional scenarios and the process of re-priorization and consensus.
- Predefine critical scenarios: in the initial utility tree, scenarios are missing because students must propose others during the activity through brainstorming (to train students in identifying architectural risks and ways of mitigating them). However, the case must include some predefined critical scenarios to anticipate possible problems during brainstorming; e.g. if students fail to propose scenarios, the teacher can show names and descriptions for possible scenarios from this list, which may also serve as an example to identify risk scenarios.

3) *Procedure for case study application*: once the teacher has created the case, they should follow these steps with the students:

- The teacher shows a brief overview of the case study and the system associated.
- The teacher explains the different roles and each student chooses a stakeholder role (in the ATAM-RPG game).
- The teacher emphasizes that the score of each participant depends on achieving their role goals and not exceeding allotted time.
- When the game starts, the teacher assumes the role of moderator and conducts the activity for each step of ATAM method.
- In each ATAM step, each participant defends their goals, expressed as values of quality attributes.
- In each ATAM step, the teacher (in the role of Moderator) calls vote when prioritizing quality attributes or modifying the system architecture, and calls brainstorming to decide additional scenarios.
- Once the allotted time is completed, the group evaluates the level of goals fulfillment for each role.

VI. Case Study: SNAM

The initial teaching case is based on the Chilean National Tsunami Warning System (SNAM, “Sistema Nacional de Alarma de Maremotos”). Tsunamis are an actual and frequent problem in Chile, and the population needs to be made aware of possible events about this nature. Also, there are previous studies [15] and [16] that explains the architecture of SNAM Systems and the people and organization involved in the correct operation of it, so we had the stakeholders needed to represent an ATAM meeting.

The following describes the application of each step of the procedure for case study design explained in the previous section.

1) *Case Description:* the case description starts like this [15]: “You belong to a unit of the Chilean Hydrographic and Oceanographic Service (SHOA), and your job is to ensure that your units' goals are represented in the development and future operation of the new National Tsunami System Alert (SNAM). One key subsystem is the Warning Decision-making Subsystem, which allow generation and monitoring of tsunami warnings; it operates as follows: ...” etc.

2) *Roles description:* seven roles were used: (1) CSN (National Seismic Center) informant official, (2) SNAM seismic scenarios DB provider, (3) SNAM seismic expert, (4) SNAM decision maker, (5) Alert dispatcher, (6) SHOA requirements officer, and (7) SHOA maintenance officer. Additionally, the are the typical roles of the ATAM method like the project manager and the architect.

3) Goals description: each role must be given its own goals (see example in Table I), and hence allies and opponents (fixed for each game application); indeed, a key task of the case designer is ensuring that there are conflicting goals.

4) *Predesigned role dialogues:* predesigned dialogues (see Table II) were developed for each ATAM step, for teacher and students. Participants can modify a dialogue before sending it (see Figure 3).

TABLE I. GUIDANCE TABLE TO BALANCE ROLE GOALS

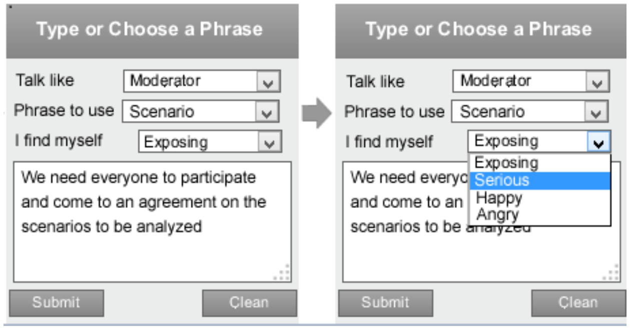
Guidance	Description
Role name	National Seismologic Center liaison officer
Goal	Reduce maximum sending time of technical data from 1’20” to 50”
Suggested phrase	“I need that the system allow me to send the information faster”
Allied roles	SNAM Seismic Expert
Opponent roles	SHOA Acquisitions Officer; SHOA Software Systems Maintainer
Expected opposing phrases	SHOA AO: “There is real need to invest in larger machines of alternative sending channels”; SHOA SSM: “If there are alternatives, there will be shared functions, so the system must be tested more thoroughly”
Possible scenarios	No Internet connection / No phone connection
Possible architecture changes	New interface for alternative channel in Functional View

TABLE II. PREDESIGNED DIALOGS EXAMPLE FOR SNAM CASE STUDY

Phase	Project Manager phrases (Teacher)	CSN Informant Official phrases (Student)
Presentation	Welcome! In this meeting, the proposed architecture for the SNAM will be presented taking into account your observations. Now, everyone presents yourself.	Hello, I am the CSN Informant Official and I have to inform to the SNAM and seismic experts about the technical data about seismic events.
Initial Architecture	Now, our applications architect will present the proposed architecture for the system.	I have a question about the architecture.
Negotiation	Now, we know the architecture and I want to know whether you have any objection or any aspect no considered.	Yes, I think that should be taken into account the maximum time to send technical data to 50 seconds. Due to___________.
Utility Tree	Now, we know the existent problems and we can use the utility tree to observe and analyze the viability of solutions and the desired priority.	I think that according to the difficulty and relevance of this attribute, it should be a priority of _____, due to________.

5) *Initial architectural design:*the initial architectural design for SNAM was described using three views: contextual, functional, and deployment [15]. Figure 4 shows the conceptual and functional views and Figure 5 shows the deployment view for SNAM system.

ATAM-RPG has interactive tools to modify utility trees and architecture views; changes can be seen immediately by all activity participants. Figure 6 shows the editing tool for architectural views.


Type or Choose a Phrase

Talk like: Moderator

Phrase to use: Scenario

I find myself: Exposing

We need everyone to participate and come to an agreement on the scenarios to be analyzed

Type or Choose a Phrase

Talk like: Moderator

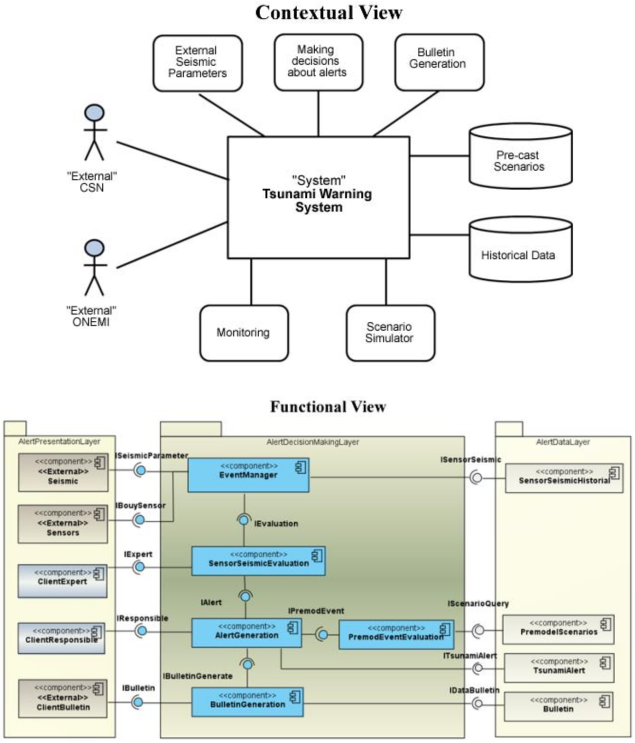
Phrase to use: Scenario

I find myself: Exposing

- Exposing
- Serious
- Happy
- Angry

We need everyo… and come to an… scenarios to be analyzed

Figure 3. Use of chat and predesigned phrases


Contextual View
External Seismic Parameters
Making decisions about alerts
Bulletin Generation
"External" CSN
"System" Tsunami Warning System
Pre-cast Scenarios
Historical Data
"External" ONEMI
Monitoring
Scenario Simulator
Functional View
AlertPresentationLayer
<<component>> <<External>> Seismic
<<component>> <<External>> Sensors
<<component>> ClientExpert
<<component>> ClientResponsible
<<component>> <<External>> ClientBulletin
ISeismicParameter
IBouySensor
IExpert
IResponsible
IBulletin
AlertDecisionMakingLayer
<<component>> EventManager
IEvaluation
<<component>> SensorSeismicEvaluation
IAlert
IPremodEvent
<<component>> AlertGeneration
<<component>> PremodEventEvaluation
IBulletinGenerate
<<component>> BulletinGeneration
AlertDataLayer
ISensorSeismic
<<component>> SensorSeismicHistorial
IScenarioQuery
<<component>> PremodelScenarios
ITsunamiAlert
<<component>> TsunamiAlert
IDataBulletin
<<component>> Bulletin

Figue 4. Contextual and Functional view for SNAM system

6) *Initial Utility Tree:* the utility tree for the initial architecture (see Figure 7) was developed from the stakeholders goals. Since ATAM allows (and expects) that utility trees be modified/rebuilt when scenarios are added and/or re-prioritized, the tool allows its interactive manipulation to reflect stakeholders agreements as the activity occurs.

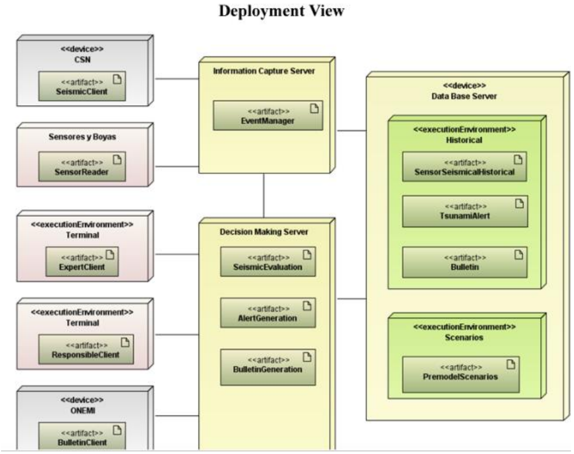
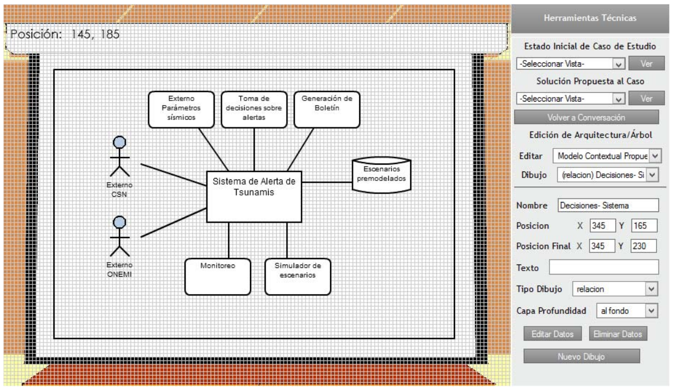


Figure 5. Deployment view for SNAM system



Posición: 145, 185

Estado Inicial de Caso de Estudio

-Seleccionar Vista-

Solución Propuesta al Caso

-Seleccionar Vista-

Edición de Arquitectura/Árbol

Editar: Modelo Contextual Propue

Dibujo: (relacion) Decisiones- Si

Nombre: Decisiones- Sistema

Posicion X 345 Y 165

Posicion Final X 345 Y 230

Texto:

Tipo Dibujo: relacion

Capa Profundidad: al fondo

Figure 6. Editing the architectural views

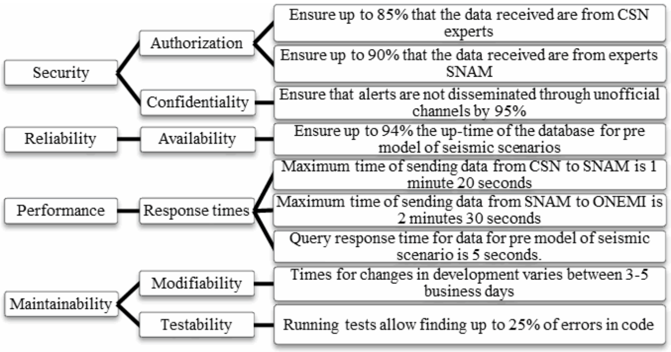

Security
Authorization
Ensure up to 85% that the data received are from CSN experts
Ensure up to 90% that the data received are from experts SNAM
Confidentiality
Ensure that alerts are not disseminated through unofficial channels by 95%
Reliability
Availability
Ensure up to 94% the up-time of the database for pre model of seismic scenarios
Performance
Response times
Maximum time of sending data from CSN to SNAM is 1 minute 20 seconds
Maximum time of sending data from SNAM to ONEMI is 2 minutes 30 seconds
Query response time for data for pre model of seismic scenario is 5 seconds.
Maintainability
Modifiability
Times for changes in development varies between 3-5 business days
Testability
Running tests allow finding up to 25% of errors in code

Figure 7. Utility tree for the initial architecture

7) *Predefined Critical Scenarios:*. table III shows the scenarios and modifications that could result in the evaluated architecture.

Once the case study is defined, the moderator must follow the procedure for the case study application explained previously to obtain results in terms of learning and level of student satisfaction.

TABLE III. POSSIBLE SCENARIOS AND MODIFICATIONS FOR THE EVALUATED ARCHITECTURE

Possible scenarios	Solutions	Possible modifications to architecture
- No Internet connection. - Failure of the primary server data premodeled scenarios. - Communication between servers is cut. - Denial-of-service attack. - Attempt to generate newsletter externally by unauthorized users.	- Add alternative channels to email.	- New interface for alternate channel on Functional View. - New internal or external mirror device data server to pre-modeled scenarios in Vista deployment. - New component with sophisticated algorithms search query interface between data and data processing component of pre-modeled scenarios in Functional View.

VII. RESULTS

ATAM-RPG has been used in live classes with a total of 16 professional, graduate-level Informatics students at Universidad Técnica Federico Santa María (Chile) and Universidad del Quindío (Colombia).

Before each activity, an expository class of ATAM was given and study material was delivered. Then, the teacher and the students used the computer equipment of the class to access ATAM-RPG, where they followed the steps for the case study application, selecting the roles and according the goals of each role. Then they interacted with each other enacting the negotiation and voting process through the different ATAM steps.

After 1 hour in the activity, students answered two surveys to assess their perception of effectiveness of this method for teaching ATAM and their satisfaction.

1) *Perception of effectiveness:* Table IV shows the questions evaluated with the average score received for each of them, where the following rating scale used as follows: 1) strongly disagree, 2) disagree, 3) neutral, 4) agreement and 5) strongly agree. In general, students agreed that ATAM-RPG was a contribution to the class and helped to better understand ATAM. Regarding practical skills, they also agreed (less strongly) on the contribution to value negotiation and to improve skills to generate utility trees. Finally, they felt even less strongly that their skills to interpret software architecture models had improved; clearly, the ATAM-RPG focus on negotiation needs to be complemented with a similar focus on architecture reading.

TABLE IV. SURVEY RESULTS FOR PERCEIVED EFFECTIVENESS

Evaluated Concept	Average
I believe that the application is a contribution to the course	4.3
Now I understand ATAM better	4.2
I recommend this software to anyone to understand ATAM	4.1
Now I value the importance of negotiation	3.9
I have now a better skill to generate utility trees	3.9
I have now a better skill to interpret architecture descriptions	3.7

2) User satisfaction: the student satisfaction survey was divided into four areas (see Table V), with general scores (in a 1-5 scale) of (a) *case study*=4.4; (b) *negotiating commitments*=3.2; (c) *architecture description and utility tree*=4.4; (d) *application design*=4.0. For the area *Case study* the results shows that such cases are an appropriated tool for subjects that comprises theory and practice but a detailed explanation of the roles ant the steps to obtain help is important. This point is an aspect of improvement for future applications of the ATAM-RPG game. Now, for the area *Negotiation commitments* the students recognize the importance of the teacher support in the negotiation (average score = 4.2) and the proper behavior of the students (average score = 4.3). However the overall evaluation is poor and we conclude that do more emphasis on a negotiation process based on technical criteria is important.

For the area *Architecture description and utility tree* the results shows a high understanding of architectural views, scenarios and utility trees. This is because ATAM-RPG is a game for strengthening the assimilation of such concepts previously presented by the teacher in expository classes. Finally, for the area *Application design,* some concepts like usability, chat log presentation and usefulness of chat tools have poor scores. This shows the need of continuous improvement of ATAM-RPG videogame for facilitate the interaction between game participants.

Moreover, one unexpected result of the evaluation was that students found that they had learned the value of negotiation in defining software architecture for a multi-stakeholder project, but they also felt that negotiation was not well supported. We attribute this to a newly gained awareness of negotiation as a concern, but not complemented with a notion of what negotiation is or how it could be supported.

TABLE V. SURVEY RESULTS FOR USER SATISFACTION

Case Study	Average	Negotiation	Average
General evaluation	4.4	General evaluation	3.2
Explanation of case study	4.2	Behavior of participants	4.3
Explanation of roles to choose	3.9	Support from teachers in the negotiation	4.2
Explanation of steps to be performed by help	3.9	Chat predefined phrases chosen for the role	3.9
Architecture description and utility tree	**Average**	**Application Design**	**Average**
General evaluation	4.4	General evaluation	4
		ATAM steps distinction	4
Ease of understanding for scenarios and utility trees	4.3	Visual organization of tools on the display	3.7
Understanding architecture views	4.1	Usability	3.3
		Chat log presentation	3.3
Ease of understanding of architectural models	3.9	Usefulness of chat tool with people in full body image	3.1

Students felt that their understanding of ATAM had improved and that ATAM-RPG complemented well the lectures they attended. We attribute this to two aspects: (1) ATAM is simulated step-by-step, and (2) the teacher had previously delivered ATAM theoretical content: ATAM-RPG was not created to impart new knowledge, but to improve knowledge already gained with a practical activity on a method hard to grasp precisely due to its high level of practical content. Thus, we recommend to deliver basic ATAM notions by exposition (for live classes) or by requiring previous resources reading (for distance education) before using this game as a practical activity

VIII. CONCLUSIONS

This article describes ATAM-RPG, a Web-based role playing game (RPG) to support teaching of ATAM (Architecture Trade-off Analysis Method) to informatics students in classroom or remote settings. ATAM-RPG is based on cases; case designers must define case-specific roles (plus the standard ATAM roles Moderator, Project manager, and Architect) and artifacts (architecture descriptions, utility trees, and scenarios). An ATAM-RPG activity requires that a teacher takes on the Moderator role, assign roles to students, conduct along the process, and evaluates participants for achieving their role's goals using technical arguments and social skills.

A first case was developed based on the Chilean National System of Tsunami Alarms (SNAM). The approach and tool were tested in a case study that deployed this specific case in a professional certification class.

As usual with RPGs, activities with strong motivational components help ultimate success: students showed great interest in the activity, enthusiasm in the interpretation of their roles, and competitiveness to meet their goals. They not only faced their teacher evaluation, but also the scrutiny of their peers.

Initial results show that ATAM-RPG was well-evaluated regarding trade-off description and understanding, and utility trees were especially appreciated. However, it was found weaker in simulating trade-off negotiation, a counter-intuitive result for a role-playing game. This could have been attributed to lack of experience with RPGs, but one of the authors [17] has significant academic and commercial experience in developing RPGs. We attribute the results to a newly gained awareness of negotiation as a concern, but not complemented with a notion of what negotiation is or how it could be supported.

Finally, ATAM-RPG allows not only to teach ATAM, but also other contents that informatics students should know and practice:

- UML Modeling: A case study could be used to provide team roles, were they debate and ask for the professor, in a role of client, about the software requirements and then they could use the same tools provided in this software to draw the UML diagrams.
- BPMN Modeling: As before, the students could be the software development team and professor the client that provides the process information, so the students could draw the process diagrams through BPMN. The toolbox should be adapted to include the BPMN notation.

- Scrum Project: The students could auto-organized themselves choosing Scrum roles, drawing backlogs and burn down charts.

Thus, ATAM-PRG allows teaching using case studies, role-playing games and technical tools. To date it has been proven effective only in an academic setup, but improving the capability to train people in specific skills it could be use in corporate training.

The final version of ATAM-RPG can be seen at http://www2.toeska.cl/atam/; visitor can create new accounts on the same page.

Acknowledgment

The authors want to acknowledge the support of the participating students. This work was partially supported by FONDECYT Grant 1140408 (TacPat4SS).

References

[1] R. Kazman., M. Klein, & P. Clements. ATAM: Method for architecture evaluation (No. CMU/SEI-2000-TR-004). Carnegie-Mellon Univ Pittsburgh PA Software Engineering Inst, 2000.

[2] L. Bass, P. Clements, R. Kazman, *Software Architecture in Practice (3rd Ed.)*. Reading, Mass.: Addison-Wesley, 2012.

[3] M. Shaw and D. Garlan, *Software architecture. Perspectives on an Emerging Discipline*. Upper Saddle River, N.J.: Prentice Hall, 1996

[4] M. Shaw and D. Garlan, *Software architecture. Perspectives on an Emerging Discipline*. Upper Saddle River, N.J.: Prentice Hall, 1996

[5] N. Rozanski and E. Woods, *Software systems architecture: working with stalkeholders using viewpoints and perspectives (2nd ed.)* Addison-Wesley Educational Publishers., 2012.

[6] M. Babar, L.Zhu, L., & R. Jeffery. *A framework for classifying and comparing software architecture evaluation methods*. In Software Engineering Conference, 2004. Proceedings. 2004 Australian (pp. 309-318). IEEE, 2004

[7] R. Kazman, R., P. Clements, & M. Klein. Evaluating software architectures. Beijing: Tsinghua University Press, 2003

[8] T. Hainey. *Using Games Based Learning to Teach Requirements Collection and Analysis at Tertiary Education Level.* Ph.D. Thesis, University of the West of Scotland, 2010.

[9] A. Rusu, R. Russell, J. Robinson, A. Rusu. "Learning Software Engineering Basic Concepts using a Five-Phase Game." 2010 IEEE Frontiers in Education Conference (FIE 2010), Washington, DC, Oct 2010. DOI:10.1109/FIE.2010.5673327

[10] F. Constantino, G. Di Gravio, A. Shaban, M. Tronci. "A simulation Based Game Approach for Teaching Operations Management Topics." 2012 Winter Simulation Conference (WSC 2012), Berlin, Dec 2012, pp 1-12. DOI:10.1109/WSC.2012.6465028

[11] T. Sitzmann. "A Meta-Analytic Examination of the Instructional Effectiveness of Computer-Based Simulation Games." *Personnel Psychology*, 64(2), pp 489–528, Summer 2011.

[12] A. Purarjomaldlangrudi, A. H. Ghapanchi. "Applications of Educational Serious Games for Information Technology Related Subjects in Higher Education." *The International Management Review* (2013).

[13] M. E. Calderón, A. Cedeño. "Enseñanza del concepto de arquitectura de tres capas integrado con UML." I Jornadas de Docencia en Computación e Informática (JODCI 2013), San Pedro, Costa Rica.

[14] A. J. Moreno Guerrero. "El proceso de enseñanza-aprendizaje mediante el uso de plataformas virtuales en distintas etapas educativas". Revista INTEFP, August 2011. http://recursostic.educacion.es/observatorio/web/es/software/software-educativo/1007-monografico-el-proceso-de-ensenanza-aprendizaje-mediante-el-uso-de-plataformas-virtuales-en-distintas-etapas-educativas. Last visit: 2014-06-15.

[15] G. Pedraza-García, H. Astudillo, D. Correal. "A Methodological Approach to Apply Security Tactics in Software Architecture Design." COLCOM 2014.

[16] D. Yachan Galarce. *Módulo de soporte de decisiones para sistema integrado de predicción y alerta de tsunamis*. Memoria de titulación para optar al título de Ingeniero Civil Informático. Universidad Técnica Federico Santa María, 2014.

[17] C. Hidalgo. *Desarrollo de software en línea para la simulación de juegos de rol*. Tesis para optar al título profesional de Ingeniero Civil en Computación e Informática: Universidad del Mar, La Serena, 2007