Improvisational Multi-Agent Architecture: an Approach to Treat Unexpected Events Using Improvisation in Problem-Solving Process

Márcia Cristina Moraes
Universidade Federal do Rio Grande do Sul, Programa de Pós-Graduação em Computação
Pontifícia Universidade Católica do Rio Grande do Sul, Faculdade de Informática
Porto Alegre, Brazil, 90619-900
mmoraes@inf.ufrgs.br, mmoraes@inf.pucrs.br

and

Antônio Carlos da Rocha Costa
Universidade Federal do Rio Grande do Sul, Programa de Pós-Graduação em Computação
Pontifícia Universidade Católica de Pelotas, Escola de Informática
Pelotas, Brazil, 96010-000
rocha@atlas.ucpel.tche.br

Abstract

This paper aims to present an improvisational multi-agent architecture that includes improvisation as a capability for rational agents to handle problems that weren't anticipated in the planning of its course of action. Usually, when rational agents are in a problem-solving process they apply traditional re-planning techniques to solve unexpected problems. Re-planning techniques have several limitations such as possible exponential complexity and inadequacy for a world characterized by unpredictable events. Our proposal allows agents to give rapid answers to unexpected situations, independently of having explicit knowledge directly applicable to such situations. We allow agents to be capable of improvising behaviors using the resources readily available to them through an improvisation process based on analogy by similarity.

Keywords: Artificial Intelligence, Autonomous Agents and Multi-Agent Systems, Planning and Scheduling, Agent Architectures, Improvisational Agents

1 Introduction

Often, an artificial agent has the capabilities of reasoning, learning and communicating with other agents. Reasoning is used by a rational agent to solve problems in a complete and correct way. To do that rational agents plan their course of action in advance, trying to anticipate the future. However, real environments may change while an agent is reasoning about how to achieve some goal, and these changes may undermine the assumptions upon which the agent's reasoning is based. Agents in real, dynamic environments need to be receptive to many possible unexpected situations, which do not typically arise in a neatly sequential fashion. Agents need to reason about their actions [1]. To do that they have to know when new facts and opportunities happened and they have to adapt their selves to each current situation. Two alternatives broadly used in treating dynamic environments are probabilistic reasoning and re-planning that can be adapted through learning processes [2]. An alternative way, proposed by Hayes-Roth and Doyle [3] pointed out that when people are in front of unexpected situations, that demand a rapid and spontaneous answer, they use their capability for improvisation. For Hayes-Roth and Doyle in this kind of situation it is sometimes better to improvise than to re-plan.

The capability to produce spontaneous answers is particularly important for interface agents that can be personified through animated characters or human faces. Some researches have shown that users apply social rules to computers [4] [5]. So, as human agents, rational agents have to present coherent, interesting and believable behaviors, bringing the illusion of life to agents and making the user to suspend his disbelieves [6]. To do that, the agents have to be believable both in an expected and an unexpected situation.

To be believable in the first case, the agents can use what we call implicit improvisation. By implicit improvisation we mean the ability to incorporate predictable alternative courses of actions. In this way, we change planning to implicit improvisation because it is more reasonable to guide agents with an abstract course of behavior than to direct them with a complete course of actions. Implicit improvisation is treated using an approach based on constraint satisfaction.

In the other case, the agents can use what we call explicit improvisation. By explicit improvisation we mean the ability to treat an unpredictable known event using improvisation techniques explicitly. With explicit improvisation we aim to introduce the concept of improvisation in the problem-solving process of rational agents, treating improvisation in this case as a process based on analogy by similarity.

In this paper we present an improvisational multi-agent architecture that incorporates the changing of planning to implicit improvisation and the introduction of explicit improvisation in problem-solving for rational agents. The architecture is based on the ideas of Improvisational Theatre. So, agents can assume one of the three possible roles: director, actor and director-actor. The director-actor is a mixed role of the director and the actor. These roles imply two possible architecture's configurations: centralized and decentralized. The first one is composed by one director and several actors. The second one is composed by several directors-actors. In order to show how to incorporate the two kinds of improvisation in rational agents we are going to focus on the director's improvisation. As the process of implicit improvisation was already described in [7], this paper presents the explicit improvisation emphasizing the director's functions.

Related works on improvisation, such as Virtual Theatre Project [8], Oz Project [9] and Improv Project [10] have focused just on predictable known events, using approaches like planning and implicit improvisation. All these works focused the actors role and didn't exam the director role. Our proposal is a novelty because it treats explicit improvisation based on the directors' abilities of handling problems that weren't anticipated in the planning of the play.

2 Improvisation and Analogy by Similarity

According to Spolin [11], improvisation is related to the spontaneity to act in a world that is in constant motion. In this way, the idea of improvisation is implicitly appeals to informal and spontaneous behaviors, that had had no previous preparation. One of the first kinds of improvisational theatre was in the Commedia Dell'Arte. In Commedia, dramatic text was replaced by *canevas*, a kind of plan which contains only the main facts in a sequence that makes possible free improvisation by comedians [12]. In traditional theatre, on the other side, representation has always been thought of as something organized in advance. Chacra [13], however, pointed out that improvised and planned representation are only different poles of the same subject, determined by degrees that make the theatrical presentation more or less formalized or improvised. If actors intend to use improvisation they explicitly are integrated in what is called Improvisational Theater. So, they don't prepare in advance all their actions and speeches, they consider the moment of spontaneity. The moment of spontaneity acts in two kinds of improvisation already mentioned: implicit and explicit improvisation. Implicit improvisation involves text improvisation (which occurs through the set of phrases and non-verbal behaviors that are left open in the play for the free interpretation of the actors) and personality improvisation (which is related to the way an actor interprets one character, considering its various physical and psychological characteristics). Explicit improvisation involves problem-solving improvisation which occurs when an agent doesn't have a plan that can be immediately applied to an unexpected situation. In this case, the director is responsible for helping actors to find solutions to unexpected problems. The director needs rapid answers to unexpected situations and he uses readily available resources to produce that answers. One approach that gives rapid answers and uses past experiences (resources) to produce new solutions for unexpected problems [14] is analogy by similarity. So, in order to handle problems that weren't anticipated, the agents implement problem solving improvisation as a process based on analogy by similarity.

In the absence of relevant specific information, traditional position in analogy studies has been that the most similar analogous brings, with higher probability, a correct solution. Russell [15] shows how a statistic analysis can be executed in order to produce a successful analogy, using only the supposition that there are some relevant characteristics in both source and target descriptions. Russell [15] and Davies [16] uses the theory of determination to provide a notion of relevance. This theory supports that as known similarities are (partially) relevant to inferred similarities, the analogical inference is guaranteed to be (partially) justified. In this way, Russell [14] proposes that at least one aspect of analogical reasoning consists of a probabilistic process of partial determinations. Through this probability the most similar analogous is guaranteed to be the most suitable analogy. When there is a higher number of relevant attributes it is necessary a closer overall match to ensure that relevant similarities are indeed present.

A simplified model for analogy in a database is: there is a target T described by m attribute-value pairs, for which we want to find a value for another attribute Q. There are several sources $S_1, ..., S_n$ (analogous) that have values for the attribute Q as well as for the m attributes known to for the target [14]. The similarity s is defined as the number of matching attribute values for a given target and source and the difference is defined by $d = m - s$. Assuming that there are r relevant attributes to find out the value of Q, $p(d,r)$ is defined as the probability that a source S, differing from the target in d attributes, matches it on the r relevant attributes. $p(d,r)$ is calculated using a simple combinatory argument [14]. Let N_m be the number of choices of which attributes are relevant such that S matches T on those attributes. Let N to be the total number of choices of relevant attributes. Then, [14] states that:

$$p(d, r) = N_m / N = \binom{m-d}{r} \Big/ \binom{m}{r} \quad (r >= 1)$$

This probability will guarantee that the most similar analogous is the most suitable analogy.

3 Improvisational Multi-Agent Architecture

This section presents an improvisational multi-agent architecture based on the ideas of Improvisational Theatre. In the proposed architecture, agents are responsible for the creation and presentation of some contents. In order to do that, they can assume one of the three possible roles: director, actor and director-actor. The director has to coordinate actors, informing to them their courses of actions, called scripts, and help actors to solve unexpected problems. The actors have to follow the director's instructions while improvising behaviors appropriate to each situation. The director-actor is a mixed role that agents can take, where they manage both responsibilities, of actors and directors. The architecture is improvisational in the sense that it includes improvisation in both actor and director functions. This makes possible the incorporation of implicit improvisation (including text and personality improvisation) related to predictable and known events and explicit improvisation (meaning problem-solving improvisation) related to unpredictable known events. Actors and director execute these kinds of improvisation in different levels of abstraction. The director is involved in the preparatory phase of improvisation and actors are involved in the execution phase.

Each agent, independently of its role, is organized around a two-level architecture. The higher-level contains processes related to the agents' cognitive capabilities and the lower-level, processes related to perception and action on the environment. In order to support the two kinds of improvisation mentioned before, the definition of the higher-level is based on the improvisational director's processes [11] and is composed by knowledge acquisition, intentions building and problem-solving improvisation, as shown in figure 1.

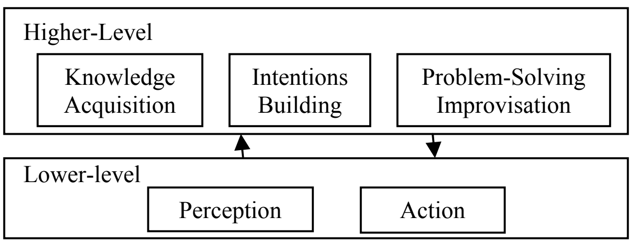

Figure 1: Improvisational Multi-Agent Architecture

Implicit improvisation is executed through knowledge acquisition and intentions building modules as a process of constraint satisfaction that already were described in [7]. The knowledge acquisition obtains information about the actors' courses of actions, composed by a sequence of, more or less abstract, activities and contents to be presented through those activities. Based on that, the intentions building module constructs intentions for the actors. The intentions are also called abstract scripts because they are abstract plans that guide, but don't determine, the agent's behavior. Explicit improvisation is executed by problem-solving improvisation module as a process of analogy by similarity. The next sections present how explicit improvisation is performed in the problem solving process, considering the director role.

3.1 Problem-Solving Improvisation Module

After acquiring knowledge, building abstract scripts of behavior and sending them to actors, the director waits for a perception. A perception structure is composed by: a type of perception, agent identification (id) and object description. If the type of perception is fail, the agent id indicates which agent asked for problem-solving and which intention (abstract script of behavior) needs improvisation. The object description has the same set of attributes of a means object (see below). A fail action is generated by an actor when occurs an event that is unpredictable for its course of action. This event can generate two different situations. In the first case, it may happen that the agent has a piece of knowledge that can be directly applied to solve the problems cause by the situation. So it could use re-planning, although this it sometimes not the most appropriated solution, as we mentioned before. In the other case, re-planning is not applicable because the agent has no knowledge on how to behave in the situation. In both situations the problem can be solved by improvisation based on analogy by similarity.

3.1.1Ends Objects and Means Objects

We assume that, in an analogy, both source and target objects are described through a generic object model. In this model, the objects are of two kinds: ends and means. Each kind of object is described using sets of attributes.
Objects classified as ends are related to the agent's goal, in a specific course of action. Objects classified as means are related to the way a specific course of action is performed. The set of attributes for ends are: *objective*, *effect* and *particular characteristics*; and the set of attributes for means are: *possible uses* and *particular characteristics*. The possible uses group has a list of uses for an object. The particular characteristics group has a set of attribute-values pairs that specify the characteristics of the object. The values for attributes (objective, effect, list of possible uses and particular characteristics) depend on the application domain.
One ends object can be related to a means object by its particular characteristics. For instance, one ends object can have as particular characteristic the name of a means object that should be part of its course of action.

3.1.2 Kinds of Improvisation for Problem Solving

Improvisation for problem solving can occur in two ways: using an *improvised means* as an alternative way to reach the current goal or using an *improvised ends* as an alternative goal that reaches the current goal through some side-effects. In the first case we have improvisation at the level of means, and in the second case we have improvisation at the level of ends.

3.2 Problem-Solving Improvisation Module's Architecture
Figure 2 shows the architecture of the *problem-solving improvisation* module, with its inputs and output. The module receives as input the description of the object that provokes the fail, the objects that represent agent's knowledge and the intention that must be reconsidered. The description of a problem object follows the same pattern of a *means* type of object, so it is composed by possible uses and particular characteristics. Based on these inputs, the module executes the intention reconsideration using a process of analogy by similarity. To do that, the module architecture is composed by three sub-modules: *identification of improvisation kind*, *analogy building* and *transforming analogy into intention*.

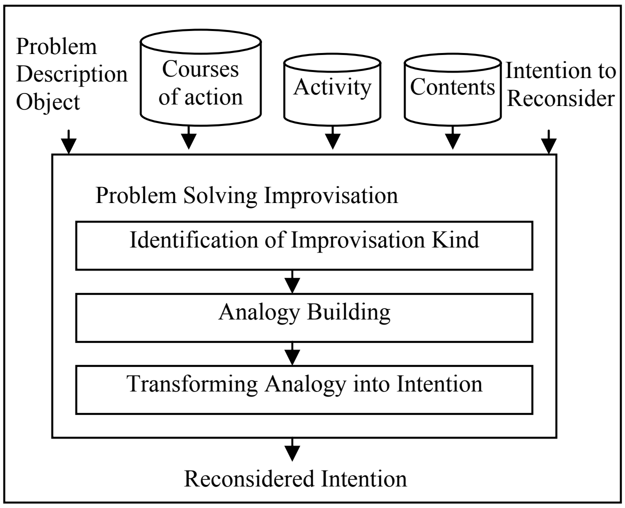

Figure 2: Problem-Solving Improvisation Module's Architecture

3.2.1 Identification of Improvisation Kind Sub-module

This sub-module receives the object representing the problem description and sends to the analogy building sub-module the type of improvisation to be executed, a list of possible analogous sources and the *problem description object*. In order to identify the kind of improvisation to be executed, the sub-module searches for *ends* objects (in the course of action objects databases) that have a goal compatible with one of the possible uses present in the *problem description object*. If one or more compatible objects are found, the kind of improvisation is of the *means* kind. In this moment, the sub-module searches for *means* objects (in the content objects database) that has possible uses and particular characteristics compatible with the *problem description object*. This process results in a list of possible analogous objects.

If it is no ends object are found, the kind of improvisation is ends improvisation. In this case, the sub-module searches for *ends* objects (in the course of action objects database), that has *effect* compatible with *possible uses* of the *problem description object*, building a list of possible analogous objects. If the list is empty, that is, there isn't any source object with effect compatible with *possible use* of *problem description* then we will have so-called improvisation without known effects. In this case, the sub-module searches for means objects (in the content objects database) that have *particular characteristics* similar to the problem object's *particular characteristics*. This process builds a list of possible analogous object.

In this way, the sub-module implements *means* improvisation and two kinds of *ends* improvisation: with known effect and without known effect. Both *means* and *ends with known effect* improvisations are due to a "strong analogy", while *ends without known effect* improvisation is due to a "weak analogy". Analogy is called strong when there is a relationship between possible uses of problem description objects and effects of source *ends* objects. Analogy is called weak when there is no such relationship, but there is some analogy between the particular characteristics. These two kinds of analogies always allow the agent to produce improvisations, even when the found analogy is not the most suitable one.

3.2.2 Analogy Building Sub-module

This sub-module uses the ideas presented by Russell [14][15] to select the most suitable analogous source object, given a target object description. So, we have to define the similarity *s*, the difference *d*, the relevant attributes *r* and the probability *p(d,r)* for each one of the possible analogous sources given by the identification of improvisation kind module.

The similarity *s*, target attributes *m* and relevant attributes *r*, are calculated for both *means* and *ends without known effect* improvisation, considering the set of attributes for particular characteristics and possible uses. In the *ends with known effect* improvisation, only the set of attributes of the *effect* kind is considered. This happens because when the goal changes, the particular characteristics need not necessarily be compatible, but the possible uses are compatible. In *means* improvisation, the current goal doesn't change, but the *means* can be different from the current one, so the sub-module takes the *possible uses* to search for similarities inside *means* objects.

After calculating the similarity of the analogous source, it must be calculated the difference among the *m* attributes that describe the target and a similarity *s*, as $d = m - s$. It is assumed *r* relevant attributes as an input argument and it is calculated *p(d,r)* as the probability of a source *S*, differing from its target in *d* attributes, matching with the *r* relevant attributes, using the formula presented in section 2. The supposition that there isn't any relevant information means that all attributes are equally relevant. The probability will guarantee that the most similar analogous is the

most suitable analogy. Although Russell proposed a probability function considering that *r* follows a probability distribution, we consider here that there is a fixed number of relevant characteristics *r*.
The sub-module applies the probability formula ($p(d,r) = N_m / N$) to each one of the possible analogous object contained in the set built by the *identification of the improvisation kind* sub-module. The chosen analogous object will be the one that has the highest probability of being similar to the target. At the end of its execution, the sub-module sends to the sub-module that transforms analogy into intention the kind of improvisation to be realized and the analogous source selected as the object most similar to the problem object.

3.3.3 Transforming Analogy into Intention Sub-module

This sub-module receives the kind of improvisation and the source object that is the most appropriate analogy to the problem description object and transforms this analogy into an intention. If the kind of improvisation is means or ends without known effect, the intention to be developed is an update of the current intention. This update will contain the source object most similar to the object that caused the problem. This similarity is calculated through the probability formula presented in section 2.
If the kind of improvisation is *ends with known effect*, the sub-module builds a new intention based on the *ends* object chosen. As mentioned before, an *ends* object has a relationship with *means* objects by its particular characteristics. The sub-module activates the intentions building module that starting from the *ends* object and its relations to *means* objects, builds the new intention and sends it back to the transforming analogy into intention sub-module.

4 Case Study: SAGRES Virtual Museum

The SAGRES system is a virtual museum that aims to support learning processes through the interaction among visitors and also between a visitor and the museum resources [17]. SAGRES facilitates the organization of visits to museums, presenting museum's information bases in a way adapted to the visitor's characteristics. The improvisational multi-agent architecture is being used to create virtual guides and personal assistants to SAGRES. At this moment, we have already done a simulation of the *problem-solving improvisation* module for the presentation of three different contents that are related to *means* and *ends* (*with and without know effect*). The first simulation involves the presentation of the content named *archeological sites*. So the description of the problem object that arrives for *problem-solving improvisation* module has the structure presented in table 1.

Object Description					
Possible Uses	Particular Characteristics				
	Kind	Area	Sub-area	Location	Floor
Archeological Sites Dinosaurs Archeology	not informed	History	In the Past	Exposition	not informed

Table 1: Description of the problem object named *archeological sites*

The *improvisation for identification kind* sub-module executes, as described in section 3.2.1, and looks, in the course of action database, for an object compatible with one of the possible uses described in the problem object. It finds a course of action, which objective is dinosaurs, that is compatible with the possible uses dinosaurs. In this way, the kind of improvisation that is going to be execute is *means* improvisation. The sub-module starts to build a list of possible analogous, searching in the content databases for objects that have particular characteristics and possible uses compatible with the problem object particular characteristics and possible uses. It finds the possible analogous objects described in table 2 to 5. The particular characteristics of a content represents information that the SAGRES system uses to present some content.

Possible Uses	Particular Characteristics					
	Kind	Area	Sub-area	Location	Floor	Contents
Before the dinosaurs Pre-history Dinosaurs Earth evolution	Specific	History	Million of years	Exposition	second	Before... The dinosaurs... Lived...

Table 2: Structure of the content named *before the dinosaurs*

Possible Uses	Particular Characteristics					
	Kind	Area	Sub-area	Location	Floor	Contents
In the past Dinosaurs Earth evolution	Specific	History	Million of years	Exposition	second	In the past... Million of years ago...

Table 3: Structure of the content *in the past*

Possible Uses	Particular Characteristics					
	Kind	Area	Sub-area	Location	Floor	Contents
In the present Dinosaurs Earth evolution	Specific	History	Million of years	Exposition	second	In the present... Nowadays...

Table 4: Structure of the content *in the present*

Possible Uses	Particular Characteristics					
	Kind	Area	Sub-area	Location	Floor	Contents
Archeological sites	Specific	History	In the Past	Exposition	second	The sites... Archeology... There are...

Table 5: Structure of the content *archeological sites*

The sub-module *improvisation for identification kind* sends the list of analogous objects (described in table 2 to 5) to the *analogy building* sub-module. This sub-module verifies the kind of improvisation and starts to calculate the similarity, difference and probability for each one of the possible analogous objects. As the improvisation kind is of *means*, the *analogy building* sub-module starts to calculate the similarity based on possible uses and particular characteristics. The similarity for objects presented in table 2, 3 and 4 is 3 because only the area and location characteristics are similar to the description of the problem object and the dinosaur possible use is present in all analogous objects. The difference for these objects is 2 because there are 5 attributes to describe the problem object. Considering that the number of relevant attributes is 2 the probability calculated by the formula $p(d,r) = N_m / N$ is:

$$= \begin{bmatrix} m - d \\ r \end{bmatrix} \Big/ \begin{bmatrix} m \\ r \end{bmatrix} = \begin{bmatrix} 5 - 2 \\ 2 \end{bmatrix} \Big/ \begin{bmatrix} 5 \\ 2 \end{bmatrix} = \frac{3!}{2!(3-2!)} \Big/ \frac{5!}{2!(5-2)!} = 0{,}3$$

The similarity for the object described in table 5 is 4 and the difference is 1. Considering that the number of relevant attributes is 2 the probability calculated by the formula $p(d,r) = N_m / N$ is:

$$= \begin{bmatrix} m - d \\ r \end{bmatrix} \Big/ \begin{bmatrix} m \\ r \end{bmatrix} = \begin{bmatrix} 5 - 1 \\ 2 \end{bmatrix} \Big/ \begin{bmatrix} 5 \\ 2 \end{bmatrix} = \frac{4!}{2!(4-2!)} \Big/ \frac{5!}{2!(5-2)!} = 0{,}6$$

After examining all the object present in the list of possible analogous object, the sub-module *analogy building* chooses the object that has the greater similarity. In this case, the chosen object was the object described in table 5. The chosen object and the kind of improvisation are send to the next sub-module, *transforming analogy into intention*. As the kind of improvisation is of *means* the current intention has to be updated. *Transforming analogy into intention* activates the *intentions building module* to do the intention's update.

The second simulation involved the presentation of the content named *meteors*. So the description of the problem object that arrives for *problem-solving improvisation* module has the structure presented in table 6.

Object Description					
Possible Uses	Particular Characteristics				
	Kind	Area	Sub-area	Location	Floor
Meteors Planets	not informed	History	Earth Planet	Exposition	Not informed

Table 6: Description of the problem object named *meteors*

The *improvisation for identification kind* sub-module executes and find out that there isn't any objective in the course of action database that is compatible with one of the possible uses in the description of the problem object. In this way, the sub-module searches for objects in the course of action database that have some effect compatible with one of the possible uses in the description of the problem object. As there isn't any, the kind of improvisation is of *ends without known effect*. So the sub-module searches in the content database for some object that has possible uses and particular characteristics compatible with the particular characteristics of the problem object description. The resultant list is composed by objects described in table 2, 3, 4, 5 and 7.

Possible Uses	Particular Characteristics					
	Kind	Area	Sub-area	Location	Floor	Contents
Minerals	Specific	History	Earth Planet	Exposition	second	The minerals... Many year ago... Many minerals...

Table 7: Structure of the content *minerals*

The sub-module *improvisation for identification kind* sends the list of analogous objects (described in table 2, 3, 4, 5 and 7) to the *analogy building* sub-module. This sub-module verifies the kind of improvisation and starts to calculate the similarity, difference and probability for each one of the possible analogous objects. As the improvisation kind is of *ends without known effect,* the *analogy building* sub-module starts to calculate the similarity based on possible uses and particular characteristics. The similarity for objects presented in table 2, 3, 4 and 5 is 3 because only the area, location and floor characteristics are similar to the problem object description. The difference for these objects is 2 because there are 5 attributes to describe the problem object. Considering that the number of relevant attributes is 2 the probability calculated by the formula $p(d,r) = N_m / N$ is:

$$= \binom{m-d}{r} \Big/ \binom{m}{r} = \binom{5-2}{2} \Big/ \binom{5}{2} = \frac{3!}{2!(3-2!)} \Big/ \frac{5!}{2!(5-2)!} = 0{,}3$$

The similarity for the object described in table 7 is 4 and the difference is 1. Considering that the number of relevant attributes is 2 the probability calculated by the formula $p(d,r) = N_m / N$ is:

$$= \binom{m-d}{r} \Big/ \binom{m}{r} = \binom{5-1}{2} \Big/ \binom{5}{2} = \frac{4!}{2!(4-2!)} \Big/ \frac{5!}{2!(5-2)!} = 0{,}6$$

After examining all the object present in the list of possible analogous object, the sub-module *analogy building* chooses the object that has the greater similarity. In this case, the chosen object was the object described in table 7. The chosen object and the kind of improvisation are send to the next sub-module, *transforming analogy into intention*. As the kind of improvisation is of *ends without known effect* the current intention has to be updated. *Transforming analogy into intention* activates the *intentions building module* to do the intention's update.

The third simulation involved the presentation of the content named *human and primate evolution*. So the description of the problem object that arrives for *problem-solving improvisation* module has the structure presented in table 8.

Object Description					
Possible Uses	Particular Characteristics				
	Kind	Area	Sub-area	Location	Floor
Human and primate evolution Earth evolution Pre-history	not informed	History	In the past	Exposition	Not informed

Table 8: Structure of the content *human and primate evolution*

The *improvisation for identification kind* sub-module executes and find out that there isn't any objective in the course of action database that is compatible with one of the possible uses in the description of the problem object. In this way, the sub-module searches for objects in the course of action database that have some effect compatible with one of the possible uses in the description of the problem object. The sub-module finds two compatible objects, described in table 9 and 10. This kind of improvisation is *ends with known effect.*

Objective	Effect	Initial Activity
Dinosaurs	Pre-history	Virtual guide presentation

Table 9: Structure of the course of action *earth stages of evolution*

Objective	Effect	Initial Activity
Earth Stages of Evolution	Earth evolution Human and primate evolution	Archeological site

Table 10: Structure of the course of action *earth stages of evolution*

The sub-module *improvisation for identification kind* sends the list of analogous objects (described in table 9 and 10) to the *analogy building* sub-module. This sub-module verifies the kind of improvisation and starts to calculate the similarity, difference and probability for each one of the possible analogous objects. As the improvisation kind is of *ends without effect knwon,* the *analogy building* sub-module starts to calculate the similarity based on possible uses of the problem object description, comparing it with the effect of the possible analogous objects. The object presented in table 9 has similarity equal to 1, because only one of the possible uses of the problem object description is in the list of this object. As *m* is equal to 3, because there are three possible uses in the problem object description, the difference is 2. Considering that the number of relevant attributes is 1 the probability calculated by the formula $p(d,r) = N_m / N$ is:

$$= \binom{m-d}{r} \Big/ \binom{m}{r} = \binom{3-2}{1} \Big/ \binom{3}{1} = \frac{1!}{1!(0!)} \Big/ \frac{3!}{1!(3-1)!} = 0{,}33$$

The similarity for the object described in table 10 is 2 because two of the possible uses of the problem object description are in the list of this object. The difference is 1. Considering that the number of relevant attributes is 1 the probability calculated by the formula $p(d,r) = N_m / N$ is:

$$= \binom{m-d}{r} \Big/ \binom{m}{r} = \binom{3-1}{1} \Big/ \binom{3}{1} = \frac{2!}{1!(2-1!)} \Big/ \frac{3!}{1!(3-1)!} = 0{,}66$$

After examining all the object present in the list of possible analogous object, the sub-module *analogy building* chooses the object that has the greater similarity. In this case, the chosen object was the object described in table 10. The chosen object and the kind of improvisation are send to the next sub-module, *transforming analogy into intention*. As the kind of improvisation is of *ends with known effect* the current intention has to be removed and a new one has to be created. *Transforming analogy into intention* activates the *intentions building module* to build the new intention.

Analyzing the *means* and *ends* (*with and without known effect*) kinds of improvisation implemented in the *problem-solving improvisation* module we conclude that the *problem-solving improvisation* produces coherent answers for the tests applied, always choosing the source object that correspond to the most similar analogous.

As previously considered in section 3.2.1, improvisation of *ends without known effect* can produce a "weak analogy" and it does happened in our simulation. Although the produced answer was not directly related to the asked subject, it did have some relationship between the particular characteristics of the subject, indicating that they are in the same super set of contents. This shows that even when agents doesn't have complete knowledge of the subject, they can produce some plausible answer preventing their fail of execution.
In the "weak analogy" case, agents can express their lack of specific knowledge explaining that they don't know that subject but they do know another one that is similar to what was requested in certain characteristics. Doing that, agents can act like humans in the same situation, showing smart and believable behavior and consequently producing the so desired illusion of life.

5 Final Considerations

This paper presented one approach to include improvisation concept in problem-solving process for rational agents through the specification of an improvisational multi-agent architecture. Improvisation is treated like a process of analogy by similarity due to the strong relationship between these two concepts. We consider analogy and case-based reasoning as synonymous and analogy by similarity was chosen because we believe that its the theory of determination, that provides a notion of relevance, is a suitable methodology for improvisation.

The proposed architecture is based on a traditional BDI architecture [18] and can be mapped into a BDI diagrammatic architecture as presented in Wooldridge [19] with one extension. The knowledge acquisition process represents the beliefs revision function. The intentions building contains both options and filter function. Perception represents input sensor and action represents action function. The extension is present in problem-solving improvisation that contains the characteristics of both options and filter functions to build an alternative course of action through analogy by similarity when something unexpected happens by accident and the agent doesn't know how to solve it.
Showing the mapping between our proposed architecture and Wooldridge's diagrammatic BDI architecture is important in order to insert improvisation as a fundamental characteristic of rational agents, improving their capabilities to handle problems that weren't anticipated in the planning of its course of action. The choose of a BDI architecture to show the possibility of using improvisation in rational agent has made due to the significance and recognition of BDI to describe rational agents behaviors. This overcomes the existent gap between improvisation and artificial intelligence and shows that improvisation is a natural problem-solving technique.
With the *problem-solving improvisation* module we extend the works previously done on improvisation, proposing an approach to treat unexpected know events through a director agent. Previous works only have concerned actors agents that are able to handle expected know situations trough planning and implicit improvisation.
The simulation performed in the SAGRES virtual museum shows that the *problem solving improvisation* module can produce relevant answers to unpredictable known events. However, that was only a simulation and it is not possible to do considerations about the efficiency of the *problem-solving improvisation* module. We intend to do other tests in order to compare the performances of the virtual guides' architecture with and without the improvisation modules and observe more complete and reliable results, including observations on the agent's ability to solve unexpected problems.

References

[1] Pollack, M. E. The use of plans. Artificial Intelligence, Vol. 57. Elsevier Science Publishers (1992) 43-68

[2] Russell, S.; Norvig, P. Artificial Intelligence: a modern approach. Upper Saddle River. (2003)

[3] Hayes-Roth, B., Doyle, P. Animated Characters. In: Autonomous Agents and Multi-Agent Systems, Vol. 1, Kluwer Academic Publishers, (1998) 195-230

[4] Ball, G.; Ling, D.; Kurlander, D.; Miller, J.; Pugh, D.; Skelly, T.; Stankosky, A.; Thiel, D.; Van Dantzich, M.; Wax, T.: Lifelike Computer Characters: The Persona Project at Microsoft. In: Software Agents. Menlo Park, California: AAAI Press. (1997)

[5] Koda, T. Agents with Faces: A Study on the Effects of Personification of Software Agents. Master Thesis, MIT Program in Media Arts and Sciences. (1996)

[6] Loyall, B.: Believable Agents: Building Interactive Personalities. PhD Thesis Carnegie Mellon University, Technical Report CMU-CS-97-123. (1997)

[7] Moraes, M.; Costa, A. C. R.: How Planning Becomes Improvisation? – A Constraint-Based Approach to Director Agents in Improvisational Systems. In: Advances in Artificial Intelligence, 16thBrazilian Symposium on Artificial Intelligence. LNAI 2507. Berlin: Springer. (2002)

[8] Hayes-Roth, B.; Browston, L.; Sincoff, E. Directed Improvisation by Computer Characters. Technical Report KSL-95-04 – Stanford University. (1995)

[9] Bates, J. The Nature of Characters in Interactive Worlds and The OZ Project. Technical Report CMU-CS-92-200 – Carnegie Mellon University. (1992)

[10] Perlin, K.; Golberg, A. Improv: A System for Scripting Interactive Actors in Virtual Worlds. In: Computer Graphics, vol. 29. (1996)

[11] Spolin, V. Improvisation for the Theater: A Handbook of Teaching and Directing Techniques. First Edition. Nothwestern University Press. (1963)

[12] Reverbel, Olga. Teatro: Uma Síntese em Atos e Cenas. Editora: LPM, (1987)

[13] Chacra, S.: Natureza e Sentido da Improvisação Teatral. Editora Perspectiva. (1983)

[14] Russell, S.: Analogy by Similarity. In David Helman (Ed.), Analogical Reasoning, Boston, MA: D. Reidel. (1988)

[15] Russell, S.: A Quantitative Analysis of Analogy by Similarity. In Proceedings of the Fifth National Conference on Artificial Intelligence, Philadelphia, PA: Morgan Kaufmann. (1986)

[16] Davies, T. R; Russell, S.: A Logical Approach to Reasoning by Analogy. In Proceedings of the Tenth International Joint Conference on Artificial Intelligence, Milan, Italy: Morgan Kaufmann,. (1987)

[17] Bertoletti, A. C.; Costa, A. C. R. SAGRES – A Virtual Museum. In: Museums and The Web 1999 Conference. USA: New Orleans, Louisiana. (1999)

[18] Bratman, Michael E.; Israel, David J.; Pollack, Martha E. Plans and Resource-Bounded Practical Reasoning. Computational Intelligence, 4 (4). (1988) 349-355

[19] Wooldridge, Michael. Intelligent Agents. In: Multiagent systems: A Modern Approach to Distributed Artificial Intelligence. Gerhard Weiss (editor). The MIT Press, Cambridge, Massachusetts. (1999) 27-77