

Conversational Structured Hybrid 3D Virtual Environments

Pablo Almajano
Artificial Intelligence Research
Institute (IIIA)
Spanish National Research
Council (CSIC)
Universitat de Barcelona
Barcelona, Spain
palmajano@iia.csic.es

Enric Mayas
Universitat de Barcelona
Barcelona, Spain
enric.mayas@gmail.com

Inmaculada Rodriguez,
Maite Lopez-Sanchez
Volume Visualisation and
Artificial Intelligence (WAI)
research group,
Universitat de Barcelona
Barcelona, Spain
{inma,maite}@maia.ub.es

ABSTRACT

Structured 3D Virtual Environments are 3D virtual spaces where some users' interactions are regulated by a subjacent Organisation Centered Multi Agent System (OCMAS) —an Electronic Institution (EI). They are task-oriented hybrid systems, where staff (organisational) software agents support —based on activities' specification and current EI state— human users in their task achievement. The contribution of this paper is a conversational task-oriented structured 3D environment, where users interact with staff bots (i.e. the embodiment of staff agents) using natural language and, as a result, the communication is improved. With this aim, we extend the Artificial Intelligence Mark-up Language (AIML) with special tags to enable the complex flow of task-oriented conversations. They are characterised by different conversation' states —such as asked, responded, failed or confirmed— and by data types from the EI ontology. We evaluate the usability of our conversational proposal and compare it to a previous command-based interaction system. As expected, data analysis on users' skills —command-based novice vs expert profile— suggests the co-existence both conversational and command-based user-agent interaction styles. But overall, results show a higher users' satisfaction with the conversational approach which, in average, also performs better in terms of efficiency, effectiveness and errors.

Categories and Subject Descriptors

H.5.2 [Information Interfaces and Presentation (e.g. HCI)]: User Interfaces (H.1.2, I.3.6); H.5.3 [Information Interfaces and Presentation (e.g. HCI)]: Group and Organization Interfaces

1. INTRODUCTION

3D Virtual Worlds (VW) are persistent Virtual Environments (VE) initially conceived as spaces where people interact for the sole purposes of socialising and entertaining. Nevertheless, the interactive 3D Internet aims to engage peo-

ple in 3D digital experiences not only for these pleasant purposes but for serious ones, such as educational, shopping or collaborative product design.

Hybrid Structured 3D Virtual Environments enable the interaction of multiple on line users and computer programs (i.e software agents) within serious activities. Based on a combination of an Organisation Centered Multi Agent System (OCMAS) approach [4] and a 3D interface, these systems allow humans to fulfil specific complex tasks by providing them with an immersive 3D VE as well as by regulating (i.e. structuring) their interaction with other participants.

Specifically, we use Virtual Institutions (VI) as Hybrid Structured 3D Virtual Environments which combine an Electronic Institution [3](EI, an OCMAS infrastructure) and 3D Virtual Worlds. Briefly, EIs model organisations that structure participants' interaction by defining communication protocols they must follow in order to perform tasks within specific activities. These protocols are based on the speech act theory [10] so that uttered illocutions count as actions. When running a VI, in addition to the organisational specification, the EI keeps track of its current state, which includes, for instance, current number of participants. Although EI participants can enact different roles, this paper focuses on (i) participants that are software staff agents enacting institutional roles that support organisational tasks, and (ii) human users enacting external roles that join the organisation to perform these organisational tasks.

Regarding 3D VW, they provide immersive scenarios where users participate and intuitively follow the ongoing activities. A user controls an avatar in the VW and interacts with other users and staff bots (i.e the embodiment of staff agents) in order to achieve her/his goals. Thus, user-agent interaction style becomes key for enabling task completion. VWs offer chat windows to perform textual user-agent interactions. A previous work on a task-oriented 3D virtual environment [1] deployed a command-based user-agent interaction style within such chat windows. Nevertheless, command-based interaction is error-prone, specially for users that are not familiar with command-based systems.

Natural language interactions constitute a suitable alternative to command-based systems. Embodied conversational agents are popular representatives of this type of interactions. They are virtual characters which are able to en-

gage in a conversation with humans. Nowadays, they can be mostly found as virtual assistants that provide information to users in web environments. Specifically, Artificial Intelligence Mark-up Language (AIML) chatter-bots [12] are well known reactive bots which follow basic dialog structures defined in static files. They have been conceived to give general information to users under request. Additionally, they are able to ask the user about some general information (e.g. gender or name) and store her/his response in a non-typed memory (i.e. a list of string values).

However, human-agent interaction in VIs have a complex structure, and thus, conversations must follow the communication protocols in EI specifications. On the one direction, if a user initiates the interaction with a staff agent, this requires the corresponding staff bot to: First, identify user's task; Second, request required user entries and validate (and store) them based on both the EI specification and current state; and Third, to redirect the compiled institutional message to the staff agent so that the user's task can be completed. On the other direction, a staff agent initiates an interaction with a user by having its staff bot to send her/him a chat message in the VW.

The contribution of this paper is the addition of the *conversational* interaction mechanism to the Hybrid Structured 3D Virtual Environments. Particularly, staff agents, embodied as staff bots, interact with humans by means of an interaction mechanism that incorporates a new conversation system to the previous command-based system. In the latter the staff bot just understands user messages in a command-based language. In the former the staff bot converses with users in natural language and it is able to manage task-oriented conversations with the aforementioned functionalities. To do so, it extends the Artificial Intelligence Mark-up Language (AIML) with special tags to enable our task-oriented conversations. We name the resulting extension Task-Oriented AIML, which is created based on the EI specification and its flow is controlled by considering EI current state. The interaction mechanism is evaluated in terms of user effectiveness, efficiency, errors and satisfaction when performing structured tasks.

This paper is structured as follows. In next sections we present related work (Sect. 2), introduce our Conversational Structured 3D VE (Sect. 3) and present task oriented conversations (Sect. 4). We then show in Sect. 5 the results of the evaluation in the context of v-mWater, a virtual market for the trading of water rights; and its comparison with previous command system. Finally, we conclude with some remarks and future work.

2. RELATED WORK

A variety conversational systems appear in the literature. From them we may highlight tutoring systems, which are virtual tutors that follow pedagogical principles [8, 6]. On the one hand, the aim of AutoTutor [6] is to fulfil, in a specific order, student's expectations related to Newtonian physics. The virtual tutor is a 3D character with a textual interface that stores its knowledge in a curriculum script and uses Latent Semantic Analysis as a pattern-matching algorithm. Authors empirically evaluate the learning gains of students and the quality of the dialogues (by a variation

of the Turing test). Alternatively, conversations in our system follow complex protocols and need to consider system specification and current states. Additionally, our aim is to support users to achieve goals in the system rather than to have simulated humans, and thus, we evaluate the usability of our system instead of conducting a Turing test.

On the other hand, CHARLIE [8] uses an AIML (Artificial Intelligence Mark-up Language) mechanism to maintain a general conversation with students. The bot user interface is a pop-up window with a text area dedicated to the conversation. Specifically, particular keywords link AIML templates with a test task, where students can ask for predefined/personalised tests. Similarly, our bot is AIML-based, but we have extended the language to support task-oriented conversations. Moreover, our user interface is a 3D Virtual world that simulates real environments and provides an immersive experience to human users.

Other textual interfaces extend AIML with rule engines [5, 9, 13]. First, VISTA [13] (Virtual Interactive Story Telling Agent) uses a XML-based web interface to answer users' questions about specific stories. Those queries not covered by AIML patterns are processed by a logic-based engine over a dynamic knowledge base. Second, Persona-AIML [5] implements personality in chatter-bots so that they change the treatment to the users according to their mood and attitude. Persona-AIML was tested in a stand-alone application in Internet Relay Chat and in the web. Third, the work in [9] proposes a web-based multimodal conversational module that detects topic changes requested by the user and smooths the transition between topics by asking for user's confirmation. Our extension of AIML also includes new tags to support task-oriented conversations. Nevertheless, instead of using a rule engine, our bot verifies user's responses based on EI specification and current state. Moreover, these works have web-based interfaces in contrast to our 3D Virtual World interface, which manages simultaneous conversations with multiple users.

In the line of 3D virtual environments, KomParse [7] implements a bartender as a conversational 3D character in a commercial 3D massive multi-player on-line game. The bartender recommends and sells drinks to customers, and also entertains them with small talks. The knowledge base is a biographical ontology of celebrities acquired applying semantic web technology. The natural language understanding component uses a dialogue context memory and a dialogue state, and processes users' entries to select the appropriate answer. Furthermore, a Bayesian classifier is used if the input has not rule nor pattern associated. The main difference with our proposal is that, whereas KomParse is meant to entertain users in a particular game, our system models general "serious" applications, where staff agents successfully support users to perform structured tasks.

3. CONVERSATIONAL STRUCTURED 3D ENVIRONMENT

As previously introduced, Virtual Institutions paradigm allows the creation of Hybrid Structured 3D environments. VIXEE [11] is the Virtual Institutions eXecution Environment that connects an Electronic Institution (EI) and several 3D Virtual Worlds (VW). It enables the validation of

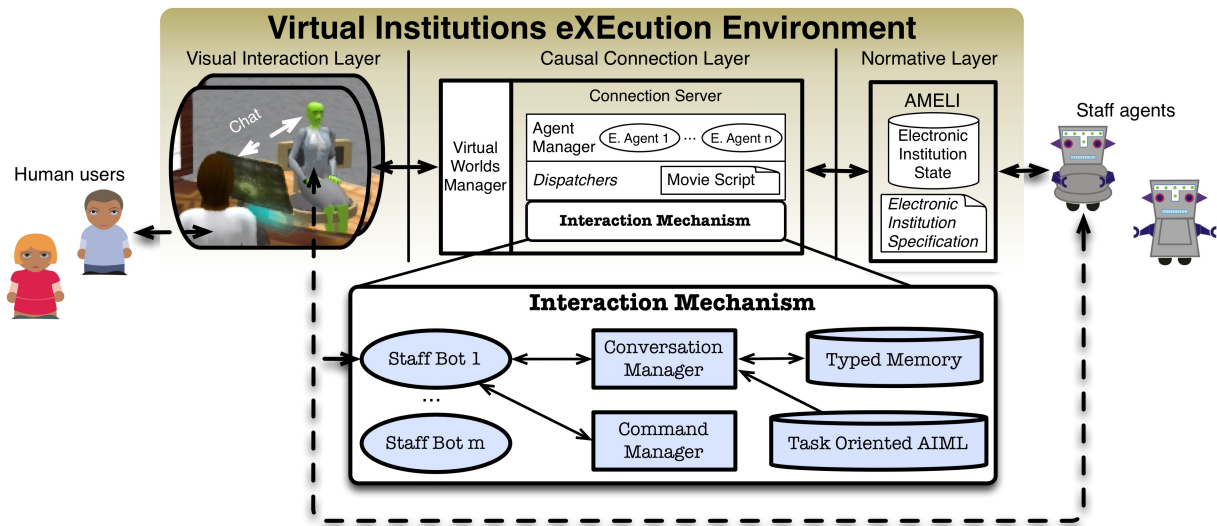


Figure 1: Conversational Task Oriented Architecture

those VW interactions which have institutional meaning (i.e. contemplated in the EI specification), and updates both VW and EI states to maintain a consistent state. This section introduces VIXEE and details the incorporation of a new user-agent (conversational) interaction mechanism.

3.1 Architecture

Top of Fig. 1 depicts VIXEE architecture structured in three main layers: Normative; Visual Interaction; and, as middle-ware, the Causal Connection Layer.

The **Visual Interaction Layer** is the 3D interface which represents an immersive space where users can interact and intuitively follow the progression of activities they are engaged in. Human users (human-alike icons on the left of Fig. 1) participate in the system by controlling the avatars (i.e. 3D virtual characters) that represent them in the virtual environment. Additionally, staff agents are visualised as staff bots in the VW (notice how a dashed arrow in Fig. 1 links robot icon on the right with staff bot character).

The **Normative Layer** on the right of Fig. 1 is composed by: AMELI [3]; the Electronic Institution (EI) specification; and its current state. AMELI is the EI execution infrastructure that regulates participant interactions by enforcing its EI specification on run-time. Briefly, this EI specification defines: participant institutional (i.e. staff) and external roles; activities where participants –enacting specific roles– perform tasks; and regulations and communication protocols associated to these activities. Communication (conversation) protocols are defined as finite state machines, where states represent different conversation stages and edges correspond to illocutions (i.e. institutional messages) that participants can interchange. These illocutions consist of a sender, a receiver and a content, which is expressed in terms of EI types (defined in an ontology) and whose specific values can come from user entries. Finally, staff agents (robot-alike icons in Fig. 1) are software programs connected to AMELI.

The **Causal Connection Layer** in Fig. 1 acts as middle-

ware. It includes both the VW manager (that mediates all the communication with the VW platform) and the Extended Connection Server (that does so with AMELI). The latter has three components: i) the Agent Manager, which represents humans as External Agents (represented as ellipses) in AMELI; ii) the Dispatchers and iii) the Extended Interaction Mechanism. The Dispatchers use the so called Movie Script mechanism to define the mapping between AMELI events and VW actions. On one hand, an event generated in AMELI triggers a VW action, and thus, the visualisation in the VW is updated. On the other hand, for each institutional action (regulated by the EI) performed by a human avatar in the VW, a dispatcher generates the corresponding illocution in AMELI.

The Extended Interaction Mechanism (zoomed in at the bottom of Fig. 1) supports human-agent interactions. In particular, each staff agent in AMELI has a staff bot (see Staff Bot labelled ellipses in Fig. 1) within this interaction mechanism that controls a staff bot character in the VW and that is endowed with a textual communication system. Human-agent communication in previous VIXEE versions were command-based, and they showed to be error-prone. Thus our proposal is to extend this with a new conversation-based system. Both systems are generic for any 3D VW (supported by VIXEE) and EI specification.

When using the command-based system, the staff bot just understands messages structured as commands, where the first word corresponds to the illocution to execute in the system and successive words map to its content. Specifically, the staff bot filters users' messages by comparing the first word with allowed EI illocutions and uses a Command Manager to validate its structured content. Commands are then directly mapped to illocutions to staff agents in AMELI. In the reverse direction, one illocution that the staff agent sends to the user is translated to one message sent by the staff bot to the user in the virtual environment.

Our focus is in the conversational system, that has the in-

intrinsic complexity of natural language dialogues, where users can express their intentions in a variety of ways and staff bots have to understand users' entries and support them in their task completion. To do so, a staff bot must: i) identify user's task; ii) if some data is required to complete the task, request the user accordingly; iii) validate user entries with respect to both the EI specification and EI current state, and store them in a typed memory; and iv) send a compiled illocution to the staff agent so that the user's task can be completed. In the reverse direction, each illocution the staff agent sends to the user should be expressed in natural language.

This bidirectional process requires a staff bot to dynamically control the flow and state of a task-oriented conversation with multiple users based on the user's entries, the staff agent's illocutions, the system specification and its current state. The staff bot delegates in a Conversation Manager the updating of the conversations' states, the storage of users' responses into a Typed Memory (with the types defined in the EI specification) and the interpretation of the conversational knowledge, which is based on the Artificial Intelligence Mark-up Language (AIML). As aforementioned, AIML has been conceived to program reactive chatter bots that follow simple conversations with users. We extend it and propose Task-Oriented AIML, which includes special tags to enable structured conversations in our system. Next section further explains this extension and the interactions among different participants in task-oriented conversations.

4. TASK ORIENTED CONVERSATIONS

The Artificial Intelligence Mark-up Language (AIML) is conceived to create chatter bots, i.e. reactive software programs that can engage in simple natural language dialogues under users' requests. AIML is in fact a XML dialect that encapsulates conversational knowledge in data objects called categories. Basically, each *category* is defined by an input question (i.e. a *pattern* for the user entry) and an output response (a *template* to generate bot's response). Categories are grouped in topics and may specify a topic change. Thus, bot designers can encapsulate conversational knowledge into topics and fix a conversation flow. Despite their simple structure and functioning, AIML chatter bots can give a response to almost any sentence the user can think of. It is a matter of expanding the conversational knowledge with more and more AIML files.

Nevertheless, AIML does not perform well in our structured 3D environment, where staff bots, beyond having basic conversational skills to welcome and farewell the user, need to be proactive in order to manage conversations that support users' task achievements. Previous section introduced the Interaction Mechanism, which has been extended to enable human-bot conversations in our system. In this section, we further detail the structure and deployment of these conversations and propose our Task-Oriented AIML, an extension of AIML to support task-oriented conversations.

4.1 Conversation Structure

The structure of a task-oriented conversation has a **welcoming** topic and one **task** topic for each supported task. **welcoming** is the default topic of the staff bot, uses standard AIML tags, and is devoted to greet, inform about the activity, and

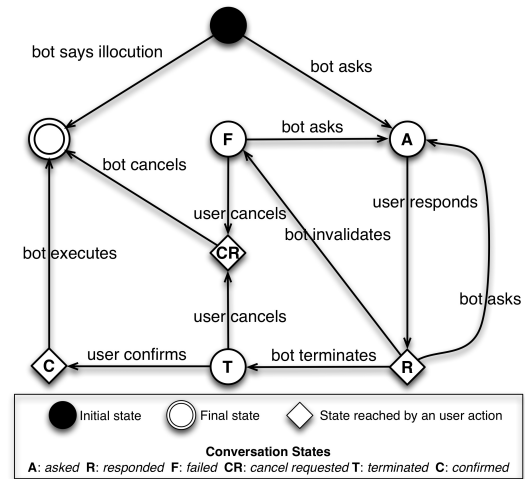


Figure 2: Conversational Finite State Machine for the task topic

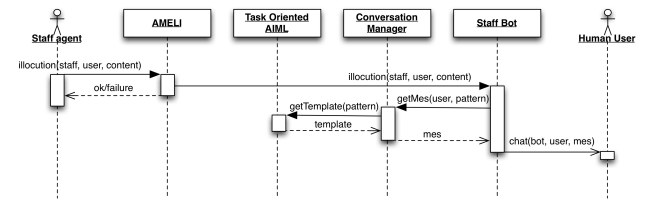


Figure 3: Agent to User interaction sequence diagram

farewell the user. Alternatively, the flow of the conversation in the **task** topic follows the structure defined by the finite state machine depicted in Fig. 2. There are two situations that trigger the change to this topic: i) the staff agent sends an illocution to the user and ii) the user sends a chat message to the staff bot.

The former trigger leads to a simple conversation where the staff bot processes the illocution that the staff agent has sent to the user within AMELI and sends the result to the user in natural language by means of the chat window. Specifically, Fig. 3 shows how the staff bot receives the staff agent's illocution from AMELI and computes the pattern that sends to the conversation manager. Afterwards, the conversation manager accesses the Task Oriented AIML to get the appropriate template, updates the conversation state and sends back the resulting message to the staff bot. Finally, the staff bot sends to the user a chat message that represents the illocution in natural language.

The second trigger leads to a more complex situation that starts when the staff bot recognises the user entry as a request to perform a task. This is done by using standard AIML categories that redirect particular user entries to a **task** topic. At this point, the staff bot takes the initiative by asking to the user the data needed to complete the task (see *asked* state in Fig. 2) and waits for the user response to be processed in the *responded* state. If the user entry is not valid, the staff bot informs the user, and the con-

versation state becomes *failed*. In this state, the staff bot repeats the question (reaching again the *asked* state) until the user requests the cancellation (*cancel requested* state), provoking the staff bot to finish the task. Alternatively, if all information is successfully gathered, the staff bot terminates the questions (*terminated* state) by presenting to the user the obtained information and requesting her/his confirmation before executing the actual illocution in the system. At this point, the user can request the cancellation with the same consequences as for previous *failed* state. Otherwise, the staff bot executes the related user illocution within the system, informs the user accordingly, and changes the topic back to *welcoming*.

The staff bot is thus an automata that selects the next action to perform based on user entries, illocutions defined in the Electronic Institution (EI) specification, and its content allowed values at current state. Fig. 4 represents the corresponding VIXEE’s sequence diagram. Each time a user sends a message to the staff bot, it gets from the conversation manager the user’s current task, the expected type in her/his response (if any), and the current conversation state. Then, it processes the user’s entry and computes the next AIML pattern. First, if the state is *responded*, then the staff bot tries to recognise a value in the user entry (message). To do so, it gets the definition of the related type from the EI specification and the allowed values at runtime from the EI state, and tries to find such a value in the user’s entry. If found, it sends the valid value to the Conversation Manager, which stores it into its typed memory. Afterwards, if the conversation state is *confirmed*, the bot executes the related illocution where the sender is the user, the receiver is the staff agent and the content is the stored data asked to the user.

Finally, based on the result of the previous process and the current conversation state, the staff bot computes the pattern which will lead to the next state and sends it to the conversation manager. The conversation manager interprets the Task-Oriented AIML template with the given pattern, updates the conversation state, and returns the resulted message (sentence) to the bot, which is the one that the staff bot will send to the user. Next we explain how we have extended AIML.

4.2 Task Oriented AIML Knowledge

As previously mentioned, the knowledge representing a task is encapsulated in a single topic. Moreover, we have extended AIML so that the conversation manager is able to control the flow of the conversation. Specifically, we have included two new AIML tags that are located inside the *template* tag: *taskresptype* and *taskstate*. Next we show a general example:

```
...
<template>
  <taskstate>state</taskstate>
  <taskresptype>responsetype</taskresptype>
  ...
  expression
</template>
...
```

First, the expected format (type) of the user response for this task is indicated in the tag *taskresptype*. Its *responsetype* value corresponds to a type specified in the EI ontology.

Second, *taskstate* indicates the state of the task-oriented conversation, and can be one of the conversation states defined in Fig. 2. The staff bot is able to redirect the conversation to the desired state because the categories that represent the conversation states have defined specific patterns. In particular, the final state changes the topic of the conversation to *welcoming*. This has been done with regular AIML by adding the *think* tag inside the *template* tag as in the following example:

```
...
<think><set name="topic">welcoming</set></think>
...
```

Following this schema, Fig. 5 shows an actual conversation within *v-mWater* prototype (introduced in next Sect. 5), where a tester user converses with the Registration Bot to perform a register task. Specifically, the user registers a water right with identifier *wr1* at a price of 25€. It also shows an extract of the AIML code that enables such a conversation. Categories in the *welcoming* topic change to the *register* task topic whenever the user mentions “register”. Furthermore, this fragment considers task state *asked* and requested entry EI types *wrid* and *price*.

5. EVALUATION

This section evaluates the usability of the proposed conversational mechanism, by assessing and comparing it with the command-based approach. Specifically, we have performed user tests¹ using *v-mWater* prototype, a Structured 3D Virtual Environment which implements a virtual market for the trading of water rights. In the following we define general test objectives. Next, we detail test methodology. Last, we present and discuss results.

5.1 Test objectives

The main goal is to assess the overall usability of the task-oriented conversational approach in Hybrid Structured 3D environments. To do so, we focus on different usability criteria such as effectiveness, efficiency, errors and satisfaction and compare them with the command based approach. We also aim to open some discussion about the hypothesis that *users’ skill* in command based systems may affect his *experience* with both command and conversational interaction mechanisms. Additionally, this usability study will allow us to detect design problems -in both structure and content - of task-oriented conversations in structured 3D virtual environments.

5.2 Methodology

We followed both summative and comparative evaluation methods [2]. The summative method focuses on gathering mainly quantitative data related to the usability of the conversational approach. For the comparative evaluation we

¹We encourage the reader to watch <http://youtu.be/Vild9IfuhCY>

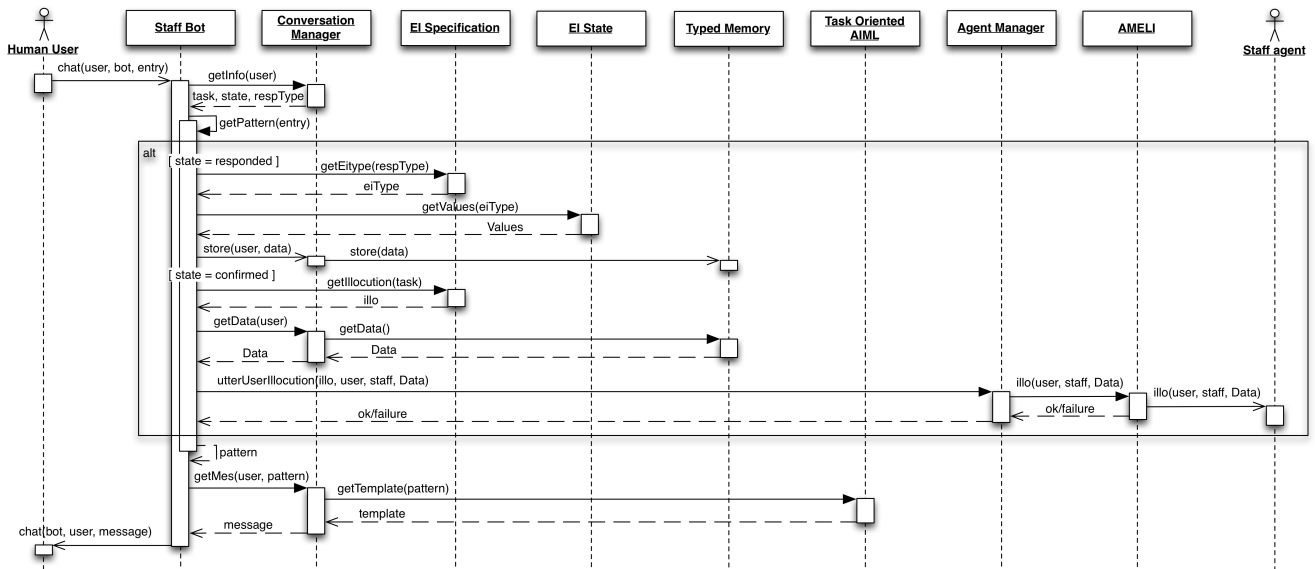


Figure 4: User to Agent interactions sequence diagram

```

<category>
<pattern>*</pattern>
<template>
<think><set name="topic">welcoming</set></think>
Greetings and welcome to the Registration building.
How can I help you?
</template>
</category>

<topic name="welcoming">

<category>
<pattern>* REGISTER *</pattern>
<template>
<think><set name="topic">register</set></think>
<srail>BOTASKDATA1</srail>
</template>
</category>
...
</topic>

<topic name="register">
<category>
<pattern>BOTASKDATA1</pattern>
<template>
<taskstate>asked</taskstate>
<taskresptype>wrid</taskresptype>
If you want to register a water right,
please give me its ID.
</template>
</category>

<category>
<pattern>BOTASKDATA2</pattern>
<template>
<taskstate>asked</taskstate>
<taskresptype>price</taskresptype>
What is your reference PRICE (in euros)
for this water right?
</template>
</category>
...
</topic>

```

Figure 5: Extract of a task oriented conversation to register a water right, and the related AIML code

conducted a within-subjects design, where each user tried each approach (conversational and command-based), measuring user’s performance for each approach.

We recruited 10 participants, half of them were selected with previous experience with command-based systems, and the other half were novices. Participants were a diverse population in terms of other characteristics such as age, sex, and occupation.

As Table 1 details, all participants were requested to repeat the same two tasks using conversational and command-based interactions. To mitigate carryover effect, half the participants started by using command-based interaction, whereas the other half started by using the conversational one.

The users were asked to perform the following tasks, literally:

Table 1: Within subjects experiment design

Participants	Task1, Task2	Task1, Task2
P1-P5	Conversational	Command
P6-P10	Command	Conversational

- Task 1: “Your goal is to ask the Information Manager about the last 2 transactions in the market”.
- Task 2: “Your goal is to ask the Registration Manager to register a water right, identified as wr1, for a price of 25€.”

Note that users performed the tasks in this order: task 1 followed by task 2. The reason was that the first task is a bit simpler than the second one and therefore we assumed novice users in command based interactions would encounter less difficulties (i.e. become less frustrated) trying first task one.

The evaluation team was composed by a moderator and an observer. The former guided the user (if needed), introduced the test, and gave the user the consent-form, task descriptions, and the post-test questionnaire. The latter took notes during the test. Tests took place at users' locations. The equipment consisted in a computer running both the VW server and the VW client. It also recorded user interactions and sound.

The test protocol consists of 4 phases. First, in the *pre-test interview* we welcomed the user, explained test objectives, and asked about their experience with command-based and conversational interactions. In the second phase, the *training*, the user played through a demo to learn how to move in 3D environments and interact with both objects and bots. This training part was mostly fully guided, except at the end, when the user could freely roam and interact in the demo scenario. The third phase was the *test*, the user performed the test tasks without receiving guidance unless s/he ran out of resources. Finally, the user answered a *post-test questionnaire* with both qualitative and quantitative questions, including a last open question for any extra comments the user could have.

5.3 Results and discussion

In this section we analyse test results and discuss the achievement of test objectives which, as introduced before, are mainly focused on usability criteria and user profile influence in task achievement. Tests results come from data collected from: post-test questionnaire, users comments, observer notes, and the review of the desktop and voice recordings while participants were performing the task.

Table 2 summarizes the seven questions included in the satisfaction post-test questionnaire, and Figure 6 depicts a compilation of users' answers. There, X axis shows each of the post-test questions and the Y axis shows average values (and standard deviation values) of answers considering a five-point Likert scale. Questions are formulated so that 1 corresponds to the most negative answer and 5 to the most positive.

Five post-questionnaire questions had double answer, one for the conversational approach and another one for the command-based, and two of them required a single answer about the conversational approach. Therefore, bar chart in Figure 6 shows five pairs of bars, dark blue and light blue for conversational and command-based respectively. Overall, the quantitative results we obtained from these five questions were very satisfactory, and the average answer for the conversational approach was higher than command-based. Individual questions (Q6, Q7) with averages of 4.7 and 4.6 show good results on the conversational bot's ability to understand the user and to give her/him meaningful responses.

Regarding *task effectiveness*, the conversational interaction style obtained a task completion rate of 100%. That was not the case for the command-based approach, where 30% of the participants failed at task 1 and 20% did at task 2. We consider a task failure when the participant could not complete the task without the help of the moderator or when the task was performed unsuccessfully (i.e. registering using an incorrect price). Additionally, users performed both

Table 2: Questions in post-test questionnaire.

Brief description	
Q1	I did not feel I needed help while talking to the bot.
Q2	I did not feel frustrated while talking to the bot
Q3	What the bot said to me made sense
Q4	I did know what to answer to the bot
Q5	How comfortable was the communication
Q6	I felt that the bot understood me
Q7	The bot had answers I expected

Possible answers are: 1: Never/ Very Uncomfortable
 2: Sometimes/Uncomfortable - 3: Regularly/Normal
 4: Often/Comfortable - 5: Always/Very Comfortable

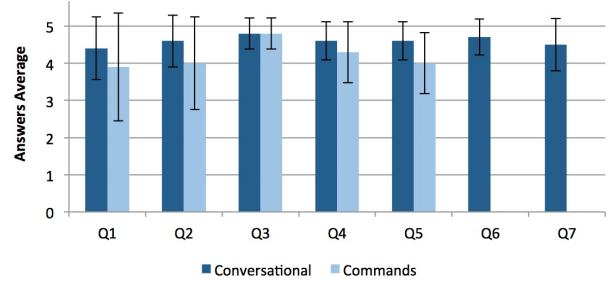


Figure 6: Post-test questionnaire results. X axis: questions from Table 2. Y axis: average (and standard deviation) values.

tasks making a lower average of *errors* in the conversational system (0.3 errors in average) than in the command-based (2.2 errors in average). This error difference was significant with a p-value = 0.01 obtained in a t-test.

Related to *efficiency*, we report results on the number of messages users needed to send to the bot to successfully complete the given tasks, either in the conversational or the command-based approach. They sent an average of 6,7 messages using commands and 9,7 messages when conversing using natural language. If we analyse these averages respect to the minimum number of messages needed for both tasks in command (that was 4 messages), and the minimum messages for conversational (that was 9), they represent a 168% in commands, and only 107% in the conversational approach. These results show that users who interacted by using natural language spent less effort pursuing their goal. This is also corroborated by the smaller number of errors made when conversing.

If we analyse results by user's skill in command based systems (i.e. expert or novice), satisfaction post-questionnaire shows that experts rate almost identically both methods (conversational with an average of 4.35 and command with 4.57) while for novices natural language is rated significantly higher, with an average of 4.65 compared to the average of 3.63 obtained in commands, with p-value=0.014. This result denotes that experts feel comfortable with both interaction methods, which is not the case for novices. When it comes to the number of errors in the command-based interaction, the difference between novices (with an average of 3.8) and experts (with an average of 0.6) was proved to be significant with a p-value of 0.01. Additionally, novices sent more messages than experts, with an average of 8.4 and 5 respectively,

again with a p-value of 0.01. If we analyse the results obtained in the conversational approach, data collected shows that both experts and novices made a similar number of errors and sent a similar number of messages, demonstrating that both user profiles behave similarly when using the conversational interaction approach. This data analysis on users' skills suggests the use of multimodal interaction, facilitating the coexistence of both interaction styles.

User tests also aimed to detect faults, shortcomings, or inconsistencies in the definition of AIML task-oriented conversations, as they may affect negatively user-bot interaction. As previously mentioned, task-oriented conversations are structured in welcome, task and farewell stages. Nevertheless, the staff bot did not welcome the user proactively, that is, the staff bot waited for the user to take the initiative in the greeting. Therefore, some users that were eager to ask the staff bot about the task they wanted to perform, received as response a greeting, being necessary for the users to repeat their request. Moreover, some users became confused by an interrogative expression used by the staff bot related to last transactions, since they assumed they had to provide transaction identifiers, which were unknown, instead of the number of transactions (i.e., how many). To avoid this or similar confusions, staff bots' AIML should be reviewed. Finally, when a task required the user to introduce several data, some users wanted to give all the data in a single sentence. They thought the staff bot could understand the entire sentence but this was not the case. This is another point to take into account in the revision of staff bots' knowledge.

6. CONCLUSIONS AND FUTURE WORK

Virtual Institutions are Hybrid Structured 3D Virtual Environments (VE) where participants (both humans and software agents) perform serious activities. These activities are specified (e.g. roles, participants' dialogs) at design time and are enforced at run-time by the execution infrastructure. Actually, staff agents are institutional agents in charge of facilitating organisational activities. Then, in many occasions users should interact with staff bots (the embodiment of staff agents) who support them in their task achievement.

This paper has presented the integration of a new conversational mechanism for user-agent interaction in Structured 3D VEs. To do so, we have proposed an AIML extension for dealing with task-oriented conversations, which are based on activities' specification and current system state. We have evaluated the conversational mechanism in *v-mWater*, a Virtual Institution for the trading of water rights. Test results have given good usability measures of efficiency, efficacy and user satisfaction for the conversational approach. We have also compared it with another interaction style already incorporated in the infrastructure which is based on commands. In the satisfaction post-test questionnaire the conversational approach has been better rated than the command-based one. Nevertheless, further data analysis, based on users' skills (in the commands-based approach), suggests us the coexistence of both approaches.

As on-going work we are re-designing AIML conversations to take into account tests results. We also plan to incorporate voice conversations that will further facilitate users' tasks.

7. ACKNOWLEDGMENTS

Work funded by Spanish CSD2007-0022, TIN2011-24220 and TIN2012-38876-C02-02 research projects.

8. REFERENCES

- [1] P. Almajano, E. Mayas, I. Rodriguez, M. Lopez-Sanchez, and A. Puig. Structuring interactions in a hybrid virtual environment: Infrastructure & usability. In *GRAPP'13*, pages 288–297, 2013.
- [2] D. Bowman, J. Gabbard, and D. Hix. A survey of usability evaluation in virtual environments: classification and comparison of methods. *Presence: Teleoperators & Virtual Environments*, 11(4):404–424, 2002.
- [3] M. Esteva, B. Rosell, J. A. Rodríguez-Aguilar, and J. L. Arcos. AMELI: An agent-based middleware for electronic institutions. In *AAMAS'04*, pages 236–243, 2004.
- [4] J. Ferber, O. Gutknecht, and F. Michel. From agents to organizations: An organizational view of multi-agent systems. In *Agent-Oriented Software Engineering IV*, volume 2935 of *Lecture Notes in Computer Science*, pages 214–230. Springer Berlin Heidelberg, 2004.
- [5] A. M. Galvao, F. A. Barros, A. M. Neves, and G. L. Ramalho. Persona-aiml: An architecture developing chatterbots with personality. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 3*, pages 1266–1267. IEEE Computer Society, 2004.
- [6] A. C. Graesser, P. Chipman, B. C. Haynes, and A. Olney. Autotutor: An intelligent tutoring system with mixed-initiative dialogue. *Education, IEEE Transactions on*, 48(4):612–618, 2005.
- [7] T. Kliwer, F. Xu, P. Adolphs, and H. Uszkoreit. Evaluation of the komparse conversational non-player characters in a commercial virtual world. In *LREC*, pages 3535–3542, 2012.
- [8] F. A. Mikic, J. C. Burguillo, M. Llamas, D. A. Rodríguez, and E. Rodríguez. Charlie: An aiml-based chatterbot which works as an interface among ines and humans. In *EAAEIE Annual Conference, 2009*, pages 1–6. IEEE, 2009.
- [9] K. Mori, A. Jatowt, and M. Ishizuka. Enhancing conversational flexibility in multimodal interactions with embodied lifelike agent. In *Proceedings of the 8th international conference on Intelligent user interfaces*, pages 270–272. ACM, 2003.
- [10] J. R. Searle. *Speech acts: An essay in the philosophy of language*, volume 626. Cambridge university press, 1969.
- [11] T. Trescak, I. Rodriguez, M. Lopez Sanchez, and P. Almajano. Execution infrastructure for normative virtual environments. *Engineering applications of artificial intelligence*, 26(1):51–62, 2013.
- [12] R. Wallace. The anatomy of a.l.i.c.e. In R. Epstein, G. Roberts, and G. Beber, editors, *Parsing the Turing Test*, pages 181–210. Springer Netherlands, 2009.
- [13] F.-Y. Wang, P. B. Mirchandani, and Z. Wang. The vista project and its applications. *Intelligent Systems, IEEE*, 17(6):72–75, 2002.