Agent Migration between Bodies and Platforms

Pedro Cuba
Instituto Superior Técnico
Av. Prof. Cavaco Silva, Taguspark
2744-016, Porto Salvo, Portugal
pedro.cuba@gaips.inesc-id.pt

ABSTRACT
Relational agents benefit from accompanying the user. As such, mobile devices are an attractive platform. However, depending on the task, there might be other types of devices for the task, originating the need of the agent migrating between such devices. But there is a lack of platforms that support the migration of agents between heterogeneous devices. This work addresses this lack of support. We have developed a platform that supports agent migration between heterogeneous devices and implemented a scenario of where an agent migrates between a robot and a mobile phone. This work also considered the influence that the presence of an agent in different types of devices might have in the user's ability to recognize it. We have evaluated the relevance of certain agent characteristics in identifying an agent in different devices. Results showed that personality was the most important characteristic in this matter.

Categories and Subject Descriptors
I.2.11 [**Distributed Artificial Intelligence**]: Intelligent Agents

General Terms
Design, Experimentation

Keywords
Embodied mobile agents, migration platform, agent identity, agent migration, identity cues.

1. INTRODUCTION
For several years researchers have been working in creating relational agents [1]. Applications of these relational agents can be shop assistants, advisors or other types of helpers. Someone who asks about important events in your life and suggests a walk in the park, possibly showing you the way to a park you don't usually go.

Traditionally these agents were located in a desktop computer, which prevents them to accompany the user to many different places and situations. This limits their ability to perform many of the tasks mentioned before. But mobile devices enable agents to perform such tasks. As the project Siri shows [2], mobile devices have advantages. They can be carried by the user, being always available. Moreover, the information people put in these devices, such as their contacts, schedule and tasks to do, can be used to provide better results or experience.

On the other hand, both desktop computers and mobile devices do not have the physical presence of robots, which can make certain interactions such as playing a game more enjoyable to the user [3]. Given that all devices have restrictions, a more interesting scenario is one where the agent is capable of migrating between different devices, making use of the capabilities each particular one has.

The platforms for creation of migrating agents which currently exist are designed for migrating agents between different software platforms. These do not take into account the possibility of migrating between different types of devices.

However, the fact that an can be present in different devices might cause some confusion to the user regarding its identity an agent is able to migrate between several devices might raise problems regarding its identity [4].

The identity issue is further aggravated because not all devices have the same capabilities. This can limit certain aspects of the agent's behaviour, capabilities or features that might be characteristic to that agent, therefore being vital to its identity and recognition by the users [5, 6]. For example, if we consider a robot, then its appearance cannot be changed, thus any agent that migrates to that robot is bound to have that appearance, regardless of what how he looked before. The presented problems concerning agent identity are particularly important when considering relational agents. If these relational agents are to act as a user's companion, then this user should recognize the agent as such, no matter the device where it is in.

The presented problems concerning agent identity are particularly important when considering relational agents. If these relational agents are to act as a user's companion, then this user should recognize the agent as such, no matter the device where it is in.

Taking this into consideration the focus of this work is on the research question: *"How to migrate an agent companion between heterogeneous devices, while preserving the perceived identity of the agent?"*

As a result this work had two distinct contributions: a migration platform that supports agent migration between heterogeneous devices; and a study regarding the perception of agent identity when in different devices.

Regarding the first contribution, we have developed a platform which supports such type of migration. The platform is generic so it can be used in different scenarios and with different types of devices. The use of such platform was demonstrated by implementing a scenario where the agent migrated between a mobile phone and a robot.

Then, having implemented scenario which included the migration of an agent between different devices, we conducted

the study presented as the second contribution of the work. This study considered the following hypothesis: "*Certain characteristics of an agent play a more important role in its identity. Preserving these characteristics across devices will lead to higher levels of recognition of the agent by users, even if others are not preserved.*"

To test this hypothesis we used the implemented system to evaluate different characteristics as identity cues and determine which have more influence in the perception of an agent's identity.

The remaining of this document is organized in the following way. In section 2 we present platforms and systems related to agent migration. After the related work, section 3 describes conceptual model of our work. Section 4 explains the implementation of both the migration platform and the implemented system. Then, in section 5, we present the experiment that was conducted regarding agent identity. Finally, section 6 contains the conclusions and future work.

2. RELATED WORK

The related work was divided into three subsections. First we will present some relevant platforms that are used to create mobile agents. Then, we move on to examples of embodied migrating agents. We will also present a study done concerning the issue of maintaining the identity of agents that migrate or change bodies. Finally we present a comparison of the presented systems.

2.1 Mobile Agents Platforms

An important platform is Telescript [7]. This was the first to provide the capability of transferring an executing program to a different computer. Using a single instruction the data, code and state of the program is transferred to another computer and the next instruction is executed at that computer.

Later appeared Agent Tcl [8, 9] which also has an single instruction to migrate. This platform provides services such as disconnected operation, enabling migration between two platforms that are not connected to the network simultaneous, by resorting to a third platform. After further development this platform gave origin to D'Agents [10] an agent migration platform that supported multiple programming languages (Java, Scheme, Tcl).

Another known platform is Aglets[11]. A distinguishing feature of this framework is that it provided an explicit notion of itinerary.

The platforms presented are important because they provide us with a perspective of how migration of agents is achieved and some examples of important platforms. But these platforms do not support migration between different devices. Moreover, they are also not designed to support agents with the purpose of interacting with the user, or establishing any kind of personal relationship. Given that we are particularly interested in the migration of this kind of agents and their identity, as perceived by the user, we will present some works on embodied migrating agents in the following section.

2.2 Embodied Migrating Agents

A relevant system is ITACO [12]. It has the distinguishing feature of migrating to a variety of devices, including simple a table lamp. Although the migration is only simulated, they presented results indicating that the fact the agent accompanies the user allows the creation of a relation.

Another system is the Virtual Raft Project [4, 13, 14]. This system made a remarking effort in conveying the sensation of migration. This was effort consisted of a synchronized animation in which as a part of the agent's body left the screen of a computer it began appearing in the screen of the tablet pc (the devices used in the system).

We also considered AgentSalon [15], another system where an agent serves as a personal guide for exhibitions to a user. At certain kiosks, with large screens, these agents could migrate from the user PDA to that screen. There they would engage in conversation with other agents (from other users) based on user ratings of the exhibits visited.

The Mobile Fitness Companion [16] is a system where the agent motivates the user to perform physical exercises. At home it resides in a robot. But when the user goes outside to exercise he can take the agent on a mobile device. However, this system does not have migration between devices, but rather a central server where both devices download and upload data. And the agent makes this clear to the user in the interaction.

In the commercial arena, there is a good example of migrating agents, in the Sonic Adventures game[1]. This game was played on a console which had a special memory card which could be used as a portable console. The game itself had an area where the user was able to raise pets, by showing affection, feeding them, etc. However there were certain attributes that, in order to be improved the user had to take the pets in that memory card and play specific games that were not available in the main game console.

Although these systems present migration of agents between different bodies, none of them provides a generic platform that allows creating such agents. Furthermore, there was no concern in evaluating the user's perception of the agent identity in any of the systems, although some even have reasonably different devices.

3. CONCEPTUAL MODEL

This section contains a description of the model of our migration system. In the first subsection we will present different types of migration, the type of migration chosen in this work and why. Then, we describe relevant components of a migrating agent. After which, we explain the steps of the process of migrating an agent. Finally, we present the characteristics we chose to study as identity cues of the agent.

3.1 Types of Migration

In this work we will consider two perspectives under which you can classify agent migration: according to what the agent

[1] http://www.sonicteam.com/sonicadv2/sonic_e.html (Official Website)

migrates to; and according to what data pertaining to the agent is migrated.

In the first perspective, when we refer to what the agent migrates to, we are in fact considering the different "locations" where the agent can reside and which are the differences between them? In this perspective, we see two different cases. The first, which occurs in platforms for agent migration, is when the agent migrates between software platforms (while the type of device remains the same). For example, the migration of an agent between two desktop computers, that might have different operating systems. The second case is when the agent migrates between different types of body. Considering that an agent's body is the device it resides in. When the agent changes body, it is not only a matter of the software context that executes its code, it has to deal with resources that are characteristic to each device. For example, consider an agent that migrates between a desktop computer and a robot. Assume that the robot has an arm which is able to perform gestures and the computer has a virtual environment where the agent can perform animations. The way the agent is able to perform gestures in the robot will be different from the virtual animation that performed the gesture in the computer. Our model of migration follows this second case. More precisely, it aims at defining a model that supports agent migration between, not only software platforms, but also between bodies.

Regarding the second perspective, which considers the data transferred when an agent migrates. Most cases fall into one of two alternatives: i) the executable code is carried along with the agent and its knowledge to be executed at the destination platform; or ii) where only the state and knowledge of the agent is transferred between platforms and the destination platform has a version of the agents executable code.

In our work we opted for transferring only the state and knowledge of the agent. This choice was motivated, in great part, by two reasons. One of the reasons is related to the fact that our goal is migration between different bodies. Although, in general, it may seem a disadvantage the necessity of having a version of the agent's code in each platform, in this case it is easier to manage the access and use of resources that are specific to each particular body in the different versions of the code. The second reason is that by avoiding transferring binary files, the migration becomes more lightweight, consisting only of information that can easily be described in a textual and possibly human readable form.

3.2 Elements of a Migrating Agent

A migrating agent is composed of several elements. These elements can be divided into two groups. On one hand we have what the agent knows about the world and itself, its internal state. On the other hand we have the elements that provide the agent capabilities specific do the device it is residing in, the ability to migrate and its decision process.

Although every element can be seen as part of a migrating agent there is a difference between the two groups of elements. The first group, its knowledge, is the data that is able to be transferred between devices. The second group has to be provided by the software platform existing in each device. However, these are intended to be as identical as possible in every platform.

The elements that compose a migrating agent in this work are presented in Fig. 1. The migrating agent has an identity. The identity of the agent is influenced by the agent's characteristics. In this work, we have represented a small set of characteristics (voice, appearance, personality and memory of events). The reasons why these characteristics were considered, and what they represent in particular, will be discussed in the section Identity Cues (3.4). The way some of these characteristics are presented to the user, in a particular device, may depend on the device's capabilities and restrictions. Nonetheless, the agent should ideally have a representation of these characteristics, so it can attempt to present them as close as possible to its representation, when the device allows it.

The migration module is the main responsible for carrying out the migration of the agent between devices. It can either encapsulate the state of the agent and transmit it to another device, or receive such state from a device and restore it. In addition, it enables the agent to be aware of relevant steps and events in a migration process.

The agent's decision process is responsible for the agent's reasoning. It determines how it reacts to certain stimuli, considering its state and knowledge. It defines the agent's decisions and behaviours. Although this is very characteristic of an agent, as defined in our model, it does not migrate between bodies. This must be provided by the software platform in each device. However, it should provide the same behaviour regardless of device, but know how to use the different competences available in that device.

The competences that the software platform provides are abstractions of a device's capabilities. More precisely, they represent what an agent is able to do in a particular device. For example, in a robot, with wheels or legs, there would be a competence that represented the robot's capability of moving from a place to another. As such, these belong to the group of element that must be provided by each software platform at a particular device.

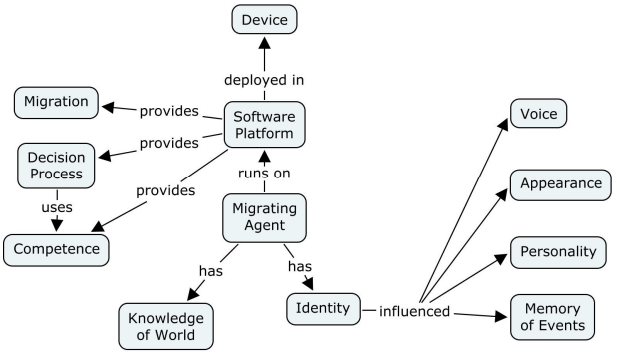

Fig. 1 Conceptual model of a migrating agent.

3.3 Migration Process

The migration process can be seen under two different perspectives: the perspective of an agent that is leaving a device (outgoing migration); and the perspective of an agent arriving at a device (incoming migration).

3.3.1 Outgoing Perspective

In our model of migration, when a migration is initiated on a device, the migration module informs the agent of that the process has started (step 1 in Fig. 2). As response, the migrating agent encodes its state in convenient format and provides that data to the migration module.

When the migration module receives the data representing the state of the migrating agent, it processes it into a format suitable for transferring between devices, adding information if necessary. This data is then transferred to the destination device. Afterwards, a response from the device is expected, confirming the success of the migration. Then, the migration module informs the migrating agent that the migration process has terminated and its state was successfully transferred to the other device.

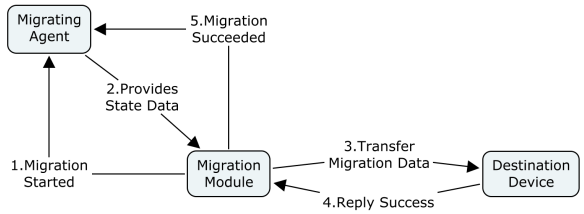

Fig. 2 Process of successful outgoing migration.

If any kind problem occurs during the migration process, then the migration module informs the migrating agent that it failed in transferring its state to the other device. This can happen, for example, if the device in question already contains an active agent.

3.3.2 Incoming Perspective

In this perspective, when the migration module receives migration data from another device it checks if it is able to receive a migrating agent. If the device is already occupied by an active agent then it refuses the migration and reports it back to the origin device. Otherwise, it accepts the migration by replying with a success message, as depicted in Fig. 3.

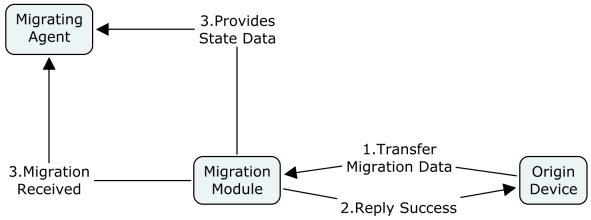

Fig. 3 Process of a successful incoming migration.

Then, the migration module processes the migration data, retrieving the state data of the agent that has migrated. In the next step of the process, the migration module informs the migrating agent that it has received a migration, and also provides the state data. Using this data the migrating agent restores the state it had on the origin device.

3.4 Identity Cues

Based on examples of the systems presented in the related work, and also considering the study of Martin [17], we decided to include in our model the following characteristics as identity cues: personality; voice; memory of past events; visual appearance.

Given the complexity of the notion of personality, we should be clear that in this work personality is modeled as a combination of the mood of the agent (e.g. happy, sad, etc.) and reaction to different events during an interaction with the user (e.g. shy, arrogant, etc.).

We also must not forget that if the user relies on these characteristics to create notion of the agent's identity, then the user should be given the opportunity to properly perceive each of these characteristics. This opportunity consists of an interaction with the agent for an appropriate amount of time.

4. IMPLEMENTATION

4.1 Scenario

In order to implement the concepts just described, we needed a scenario that would bring i) different bodies, ii) different platforms, iii) migration between the bodies and platforms and iv) direct interaction between the user and the agent.

The scenario that was chosen consists of a game between the user and the agent. More precisely, the agent and the user play chess against each other. Chess is one of the most known board games and many people know, at least, the basic rules. This makes chess a good choice because most people can interact with the agent without having to learn a specific game.

As we have mentioned, our scenario had to include different devices (and platforms) that the agent could inhabit. Therefore the agent is able to play chess in two different devices, which also have different platforms (Fig. 4): the iCat [18], a robot with a cat like appearance; and a high end mobile phone, running the Android operating system. In addition, the agent is able to migrate between both devices, carrying the state of the current game so that the user might resume playing on the other device.

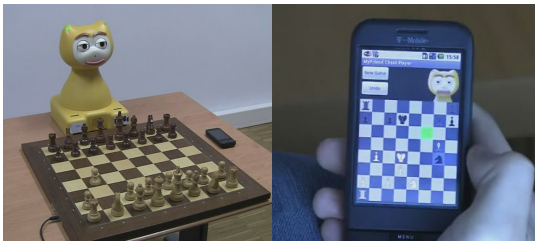

Fig. 4 Devices used in the scenario.

With this scenario we promote the interaction between the user and the agent through a chess game. The user can become aware of the agent's characteristics throughout the game by observing and listening to the agent and its reactions. In addition, the user interacts with the agent in two distinct devices, more precisely, the robot and the mobile phone.

4.2 Migration Framework

To support the migration considered, we've developed a simple and generic framework for agent body migration. This framework provides the necessary mechanisms for transferring the agent's state and its knowledge between different devices. It

also controls the activation when agents migrate to and from devices, in order to restrict the simultaneous presence of agents in a device. Both features were implemented according to the described conceptual model.

These features from the migration platform were obtained by extending the ION framework [19] with new components responsible for providing them. The ION framework is designed to create agent based simulations of virtual environments. It supports an event and request mechanism that provides an interface between components. This allows for integration, while maintaining a great degree of independence between them. This enables us to create a framework or a complete system in an extensible way, adding or replacing components if needed. This feature is useful to develop not only the migration framework, but also the systems which will use the migration framework. However, ION was implemented using the C#[2] programming language, which officially is only supported on Windows platforms. To support a wider variety of platforms, including Android, we ported the framework to Java. Thus, we were able to use ION in the creation of our migration platform.

4.2.1 Migration Architecture

The implementation of the migration platform (Fig. 5) consists of three types of components: the migration module; migrating objects; and migration aware components.

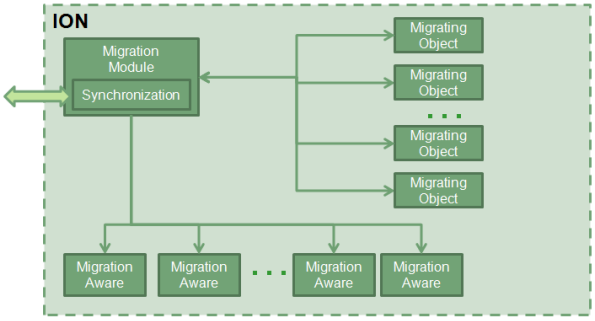

Fig. 5 Migration framework architecture.

Migration Module

The migration module is the main component in the migration of agents. It is responsible for compiling the migration data from all migrating objects, connecting to a device and transferring the migration data. While on the other device, it must receive the data, parse it into individual messages and provide those messages to the respective migrating objects, so that they can restore their state. In addition to this, the migration module must also inform migration aware components of the state of the migration process. To achieve this, the migration module was implemented by extending the ION Element. As such, a migration is triggered by a Migrate request. And to inform of the migration process it raises the following events: Migration Start, when a migration is initiated; Migration Complete, when an outgoing migration was successful; Migration Failed, when an outgoing migration failed; Message Received, containing the data of a Migrating Object; and Incoming Migration, when it receives a migration.

The task of connecting and transferring the migration data to another device is encapsulated in an internal component of the migration module, the synchronization module. This way, it abstracts the details of the technology and protocol used to transfer the data. Regarding the migration data, the format adopted by the migration module is XML. Among other advantages it is extendable and easy to parse.

Migrating Object and Migration Aware

The migration object and migration aware components were defined as an interface in Java, allowing a particular component in a system to assume both roles. The migrating object interface defines methods that must be able to translate the objects state to and from XML. The migration aware interface defines methods that are invoked in response to events indicating the state of a migration process.

The platform provides convenience methods that take care of registering each component in the migration module and invoking the appropriate methods in response to the respective events. In short, creating migration aware components or migrating objects is as simples as implement the respective interface and invoking the appropriate convenience method to register the component.

4.3 Migration Process

As stated in the conceptual model, the migration process can be divided in two parts, or perspectives: the outgoing perspective and the incoming perspective. The first shows the process of an agent leaving a device and the second shows the process of an agent arriving at a device. We will describe the implementation of the migration process from both perspectives.

4.3.1 Outgoing Perspective

The order that triggers the migration process was implemented as an ION request, the Migrate requests as mentioned earlier. This request is represented by step 1 in Fig. 6.

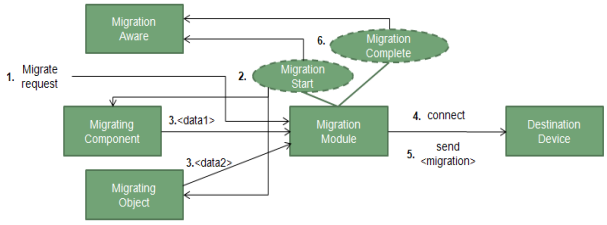

Fig. 6 Outgoing migration diagram.

When the migration module receives the migration request it raises an event to inform all migration aware components and migrating objects that a migration process to a specified device is starting (step 2). The respective event handlers are invoked by the ION simulation. In the case of migration aware components, the onMigrationStart method is invoked. In the case of migrating objects, the saveState method will be invoked and the returned XML element will be registered in the migration module (step 3).

[2] http://msdn.microsoft.com/en-us/vcsharp/aa336809.aspx

After gathering all the migration data, the migration module compiles this data into one XML document. Then, it connects to the destination device (step 4), using the internal synchronization module mentioned earlier. Upon a successful connection the migration module sends the XML document created to the destination device (step 5). After receiving a confirmation from the destination device, the migration module raises an event indicating that the process terminated successfully (step 6). This event causes the respective event handlers to be invoked, and as a consequence the onMigrationSuccess methods of the migration aware components will also be invoked. If the migration process failed at some point a MigrationFailed event would be raised, and the invoked method on the migration aware components would be onMigrationFailed.

4.3.2 Incoming Perspective

From the perspective of the receiving device, if the migration module is active then it is always listening for migration attempts from other devices. The port on which the migration module is listening is configured on the migration module configuration XML, the same file that contains the list of possible destination devices.

The migration process in this perspective begins when another device connects and sends the XML document with the migration state, which is received by the migration module. This is represented by step 1 of Fig. 7. If the migration module verifies that this device already contains an agent, then it immediately replies with a message refusing the migration and the process ends. Otherwise the reply is a message accepting the migration and the process continues.

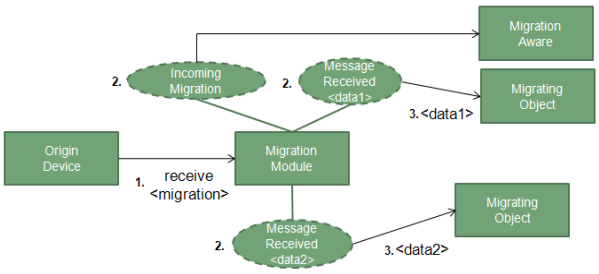

Fig. 7 Incoming migration diagram.

In the next step the migration module raises several events (step 2). One of the events signals that the migration of an agent was just received. This triggers the event handlers that will invoke the onMigrationIn method of the migration aware components.

The other events are raised according to the result of parsing the XML document that was received. The migration module parses the XML into the elements that represent the state of the several migrating objects and a Message Received event is raised for each element. This event contains not only the element itself, but also the type of message. The type is used to identify the XML element contained by the message. The event handler, invoked in response to the Message Received event, will compare the type of the message with the value returned by the getMigrationTag method of the respective MigratingObject. If the type matches then the restore state method is invoked using the XML element contained in the Message Received event (step 3). The result is a set of migrating objects which have a state identical to the state at the origin device at the time of the migration.

4.4 System Architecture

The architecture of the implemented system is presented in Fig. 8. More precisely, it represents the architecture implemented in the mobile phone. In the case of the robot the system was based on the work of Leite *et al.* [20], with some minor changes and the addition of a component similar to the migration module, in order to support migration.

The mind module and the game engine components are responsible for the decisions of the agent. Together they determine how the agent reacts and which moves it plays. These will be described in section 4.4.1.

The user and board components represent the relevant elements of the agent's environment. They model the information that the agent needs to know about its environment. A more detailed description is presented in section 4.4.2.

In this architecture we have implemented some competences, according to the notion introduced in the migration framework. This way it enables us to the reuse a great part of the system if later we extend the scenario to include new devices.

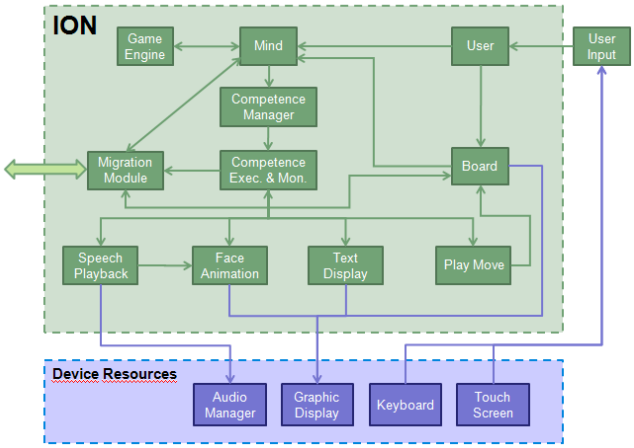

Fig. 8 System architecture.

All of the competences implemented in this system are described in 4.4.4, consisting of four different competences: face animation; speech playback; text output; and play move.

The competences are not directly connected with the mind of the agent. The competence manager and the competence execution and monitoring modules mediate the requests between the agent mind and the competences. The former decomposes the high level orders from the mind into specific competence requests. The latter takes those competence requests, issues them to the respective competences in the right order and controls the success or failure of their execution. Both these components are described in more detail in section 4.4.3.

4.4.1 Mind Module and Game Engine

The mind determines the agent's decisions and how it should react to certain events. All the actions the mind module decides to take are executed by scheduling the respective requests in the competence manager.

Although the agent only migrates upon the user's request, the mind module processes the user's request and decides that it will indeed attempt to migrate. In regard to migration, the mind module implements both the migrating object and migration aware interfaces.

When the mind detects that the user played his or her move, then the mind module plays its own move. After every user's move, the mind not only decides the agent's next move, but also the emotional reaction the agent will express. This emotional reaction is determined by the Emotivector, an affective system that was used in [20] and replicated in this system.

The agent decides its own moves and analysis its opponent moves through the game engine component. This component models the rules and heuristics of chess. Having the game engine as a separate component from the mind allows the mind to be game independent. This way, if we change the game the agent plays, the mind could be the same and only the game engine would have to change. This particular game engine was adapted from an existing open source project[3].

4.4.2 The Environment

The board component represents the state of the chess game on this architecture. Every time the board is changed it raises an event to indicate the change. The events are different according to the nature of the change (e.g. move undone, move played, piece captured). One of the components that is listening for these events is the board display. This component is responsible for showing the state of the game in the screen of the mobile phone.

The board is also a migrating object. Whenever there is a migration, the state of the game is also migrated, so the user can resume the chess match on the destination device.

The user component represents the user's action on the agent's environment, providing an abstraction of the different methods of input user might use. At the moment the methods available in the mobile phone are hardware keys or the touch screen. But this allows extending the user's way of input in the future, without changing any components within the ION simulation.

4.4.3 Competence Managing and Execution

The competence manager, as said before, receives high level requests from the mind module and decomposes them into a sequence of requests for the competences. This sequence is then scheduled on the competence execution and monitoring (CEM) for execution. For example, the mind schedules a greet request on the competence manager. The competence manager decomposes this request into two lower level requests, one to perform a smile animation and another to say a form of greet to the user (e.g. "Hello Maggie.").

This way, the competence manager allows the mind module to work at a higher level of abstraction of request. However, some of the requests that are scheduled by the mind cannot be decomposed because they already represent a request at the competence level. In particular cases, the mind schedules a request only to say something, which cannot be decomposed lower level actions.

The CEM is responsible for executing the requests scheduled by the competence manager. More precisely, it checks which available competences are able to handle the request and schedule it to be executed by them. Then it monitors the completion of the execution of the request and whether it was successful or not. This result is relayed back to the mind through the competence manager. When the request scheduled by the competence manager is a "plan" of competence actions, the CEM ensures that all competences are executed in the right order, or simultaneously if that is the case.

These two components (the competence manager and the competence execution and monitoring), by working together, create the bridge between the abstraction level at which the mind operates and the lower level at which competences operate.

4.4.4 Competences

Face animation

This competence is responsible for performing the animation of the agent's face, which is the visible area of the agent's body on the mobile phone screen. The set of animations provided by this component are identical to the animations available on the robotic platform. The appearance of the agent is also identical. The competence also simulates lip-synch, which can be used simultaneously with face animations.

Speech Playback

The speech playback competence provides the agent the ability to speak. But it has the limitation that it only supports a predefined set of sentences. To produce speech it relies on pre-recorded files with the spoken sentences. The sound files used were recorded using the same speech synthesis engine and voice used on the robotic platform. This competence interacts with the face animation competence in order to simulate lip-synch with the spoken sentence.

Text Display

The text display competence handles the same type of requests as the speech playback competence. But instead of speaking the sentence contained in the request it presents it on the screen of the mobile phone.

Play Move

The play move competence is responsible for changing the board component according to the move the agent decided to make. This is a very straightforward implementation because in this system making a move is achieved by changing the state of the board component. However, competences should provide an abstraction of the implementation of features or abilities they provide in a specific device.

[3] http://honzovysachy.sourceforge.net/

5. EVALUATION

5.1 Goal

In the beginning of this document the question to which we proposed to provide an answer was: "How to migrate an agent companion between heterogeneous devices, while preserving the perceived identity of the agent?"

In regard to preserving the identity of the agent, we have put the hypothesis that certain characteristics of an agent play a more important role in its identity than others. If these characteristics are preserved across devices, then this will lead to a better recognition of the agent by users, even if others are not preserved.

To prove this hypothesis, the goal of the evaluation is to determine the level of influence that each of the proposed identity cues has in the user's perception of the identity of agents. The result of the evaluation should allow us to identify which, if any, of these identity cues by influences the user significantly towards the identification of an agent in two different bodies. These results might be useful for the design of agents in general, but even more in scenarios where the agent migrates to devices with different capabilities and restrictions.

5.2 Method

In this experiment the independent variable was the characteristic that was kept identical in two cases of interaction with an agent (presented in different devices). The experiment had 5 different conditions according to the characteristic evaluated: voice; personality; memory; appearance; and control. Furthermore, given that we used a between-groups design for the experiment, originating a different test group per experimental condition.

In this experiment the measured dependant variable was how strongly did participants feel that the agent was the same in both interaction situations presented in their test condition.

The participants in these experiments were volunteers with no background restriction. Although we assume that all of them have at least a small familiarity with technology, given that the experiment used an online questionnaire. There were a total of 73 participants, which consisted mainly of teenagers and young adult males, as you can see in Fig. 9.

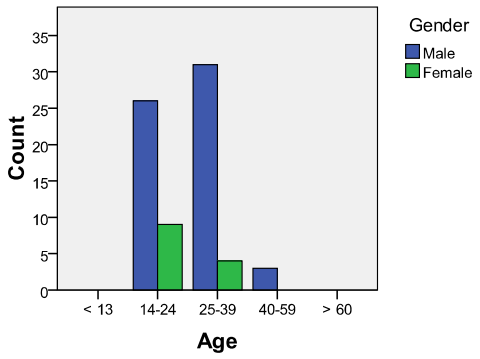

Fig. 9 Age and gender distribution of participants.

We used an online questionnaire elaborated specifically for this experiment. The questionnaire began with a short introduction mentioning an evaluation of chess characters, but nothing regarding the evaluation of identity or identity cues. Then followed two questions, regarding age and gender. After which, participants were asked to watch two different videos.

The first video contained parts of a chess game between a person and an agent in robotic embodiment (the iCat robot). This video was the same across all test groups. The agent shown was using a voice with an English voice with an American accent and its personality was pleasant and motivating to the player. The second video was also of a chess game, but this time it presented a virtual representation of the agent and a virtual chessboard in a mobile phone. This video was different according to the experimental condition. In each condition the agent presented, at most, only one characteristic in common with the first video, as shown in Table 1. For example, in the appearance experimental condition the agent in the second video looked the same, but had a different personality, voice and memory. The participants were not informed about the characteristics of the agent that were subject to change.

After watching the videos the participants were required to indicate how much did they agree that the agent presented in both videos was the same, using a 7 point Likert scale. They were also given the opportunity to mention, in an open answer format, what they found alike or different between the agents in both videos.

5.3 Results

When processing the data gathered from the online questionnaires, we noticed that some participants provided an extremely low score on the similarity of the characteristic that was identical in both videos they had watch. If a participant did not find a certain degree of similarity between both videos, regarding the characteristic that was identical, then the score he provided regarding the identity of the agent did not reflect the influence of that same characteristic.

We have mentioned in our hypothesis that an agent has characteristics that play a more important role in its identity than others. As such, we were interested in determining the influence each characteristic had on the participants perception of the agent's identity. However, since the score obtained in the cases that we have mentioned did not reflect the influence of the characteristics that were kept the same, we have excluded these results from the statistical analysis. More precisely, we have excluded cases where participants indicated that the characteristic that was preserved in both videos, was completely different using a grade below 2 (the lowest the value was, the more different the characteristic was considered). For example, in the voice test group we did not consider the results of participants who provided a score of 1 or 2 for the similarity of the voice of the agent in both videos.

Finally, the results of the experiment are presented in Fig. 10. More precisely it shows, for each experimental group, the results of the scores provided by the participants when referring to the agent identity in both videos. There are some reasonable differences between the median scores of some of the experimental conditions.

Table 1Variations of the agent's characteristics in each of the experimental conditions.

Characteristic	First Video	Second Video				
		Appearance Condition	Memory Condition	Personality Condition	Voice Condition	Control Condition
Appearance	Cat	Cat	Human	Human	Human	Human
Memory	Don't recall game	Don't recall game	Recall game	Don't recall game	Don't recall game	Don't recall game
Personality	Friendly	Arrogant	Arrogant	Friendly	Arrogant	Arrogant
Voice	American accent	British accent	British accent	British accent	American accent	British accent

The scores in the Personality and Appearance groups are the highest. On the other hand, the scores in the Control and Memory groups are identical and the lowest of the experiment.

According to the Shapiro-Wilk test the scores of the memory and voice group do not follow a normal distribution ($D(9) = 0.811$, $p = 0.27$ and $D(10) = 0.813$, $p = 0.16$), which indicated that a non-parametric test is the best choice [21]. However, we were not able to conclude that the identity cues that we used significantly influenced the perception of an agent's identity using the Kruskal-Wallis test ($H(4) = 6.25$, $p = 0.181$). This means that, statistically, we could not determine that there is a difference between the level of influence each of the characteristics (considered in the study) has on the users' perception of the identity of the agent.

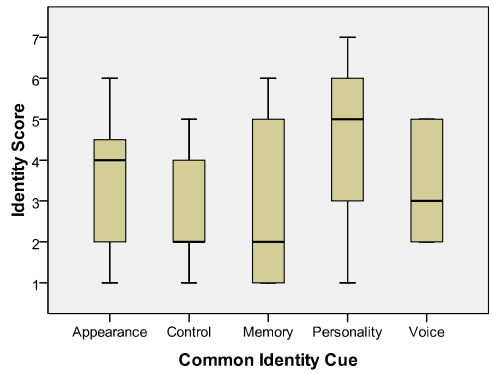

Fig. 10 Agent identity box plot per test group.

Despite the result of the previous test, we considered the personality condition, which had the highest median and compared it to the control condition separately, using the Mann-Whitney test. In fact, there was a significant difference in the agent's perceived identity ($U = 22.5$, $p = 0.034$). Furthermore, personality produced a medium size effect in increasing the perception of an agent's identity ($r = -0.47$).

5.4 Analysis

When we look at the results of the experiment (Fig. 10) we are able to see that personality (as considered in this work) had the stronger influence in the perception a person has of an agent's identity. As to appearance it also had reasonably high scores in influencing the perception of the agent's identity. We expected such an effect beforehand, because it was the only visual characteristic. Therefore it could be immediately observed by participants, independently of the interaction presented in the video or its length. Other characteristics required the user to retain its attention for a longer period of time in order to be perceived.

A relevant problem in this experiment was that many participants didn't observed or noticed the characteristics that we were evaluating in a proper manner. We can see this effect by looking at the number of participants whose results were not considered. This could be a consequence of, in a general manner, the characteristics not being very clear and not exposed in a way that participants could easily perceive them. In addition, it would be preferable to have participants physically interacting with the agents for a longer period of time, instead of just observing videos. But this alternative was discarded in order to have a greater control on the length of the experiment and to be able to reach a larger number of participants.

These reasons combined with the relatively small sample of participants are possibly responsible for not having significant quantitative effects of the identity cues over the perception of the agent's identity. Regardless, by looking at the obtained results we believe that they still indicate an effect of using these identity cues on the agent's identity and that we should not ignore it. This might be demonstrated by designing and conducting a different experiment, avoiding the problems mentioned above.

However, the results indicated that personality might be the best characteristic to use as an identity cue. It had the best results in maintaining the identity of the agent in the user's perspective. In fact, we've seen that if we compare it individually to the control group it has statistically significant improved results with a reasonable effect size. These results can also reinforce our idea that the other characteristics might also influence the perception of an agent's identity, even though they might have a weaker influence than personality.

6. CONCLUSIONS

This document presented scenarios of agent migration between different devices as a very attractive situation, in particular for

relational agents. But despite this there is a lack of platforms supporting migration between different devices. We also mentioned the obstacles that migrating an agent between different devices presents to the user's perception of its identity.

We've addressed these issues by presenting a model of migration which supports agent migration between different devices. Then we described the implementation of a migration platform according to this model.

This migration platform was used to implement a scenario where and agent plays chess with the user. To do so it could either use a robot or a mobile phone, being able to migrate between both. This scenario demonstrated the use of the migration platform and was used to conduct a study regarding agent identity.

The study that was conducted evaluated the influence of different characteristics of the agent in the perception of its identity. The agent in the robot was compared to when in the mobile phone, using different sets of characteristics in each device. The results showed that personality had a significant influence in the user's perception of the agent's identity. Although other characteristics, such as appearance and voice, were not statistically significant, the results indicated that these could also have a relevant influence in the perception of the agent's identity.

6.1 Future Work

In this work some the main improvements and courses of future work that we consider are relative to the migration framework and the evaluation. Regarding the migration framework an important feature that could be added is a device discovery service. This would allow the dynamic addition of host devices using only the devices. As to the evaluation of identity cues, it might prove fruitful a follow-up experiment, where the participants interact with the system directly.

7. REFERENCES

1. Bickmore, T. and R. Picard, *Establishing and maintaining long-term human-computer relationships.* ACM Trans. Comput.-Hum. Interact., 2005. **12**(2): p. 293-327.
2. Gruber, T., *Siri: A Virtual Personal Assistant*, in *Keynote Presentation at Semantic Technologies conference (SemTech09)*. 2009.
3. Pereira, A., et al., *iCat, the chess player: the influence of embodiment in the enjoyment of a game*, in *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems - Volume 3*. 2008, International Foundation for Autonomous Agents and Multiagent Systems: Estoril, Portugal.
4. Tomlinson, B., M.L. Yau, and E. Baumer. *Embodied mobile agents*. in *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*. 2006. Hakodate, Japan: ACM.
5. Duffy, B.R., et al. *Agent chameleons: Moving minds*. in *Proceedings of The IEEE Systems, Man & Cybernetics, UK-RoI 2nd Workshop on Intelligent Cybernetic Systems*. 2003.
6. Martin, A., et al. *Empowering Agents within Virtual Environments*. in *Proceedings of the IEEE Systems, Man, and Cybernetics, UK-RoI 3rd Workshop on Intelligent Cybernetic Systems*. 2004: IEEE Computer Society.
7. White, J., *Mobile Agents Make a Network an Open Platform for Third-Party Developers*. Computer, 1994. **27**(11): p. 89-90.
8. Gray, R.S., *Agent Tcl: a flexible and secure mobile-agent system*, in *Proceedings of the 4th conference on USENIX Tcl/Tk Workshop, 1996 - Volume 4*. 1996, USENIX Association: Monterey, California.
9. Gray, R., et al., *Mobile agents: the next generation in distributed computing*, in *Proceedings of the 2nd AIZU International Symposium on Parallel Algorithms / Architecture Synthesis*. 1997, IEEE Computer Society.
10. Gray, R.S., et al., *D'Agents: applications and performance of a mobile-agent system.* Softw. Pract. Exper., 2002. **32**(6): p. 543-573.
11. Lange, D.B. and M. Oshima, *Mobile agents with Java: The Aglet API.* World Wide Web, 1998. **1**(3): p. 111-121.
12. Ogawa, K. and T. Ono. *ITACO: Constructing an emotional relationship between human and robot*. in *RO-MAN 2008: Proceedings of the 17th IEEE International Symposium on Robot and Human Interactive Communication*. 2008: IEEE Computer Society.
13. Tomlinson, B. *A heterogeneous animated platform for educational participatory simulations*. in *Proceedings of the 2005 conference on Computer support for collaborative learning: learning 2005: the next 10 years!* 2005. Taipei, Taiwan: International Society of the Learning Sciences.
14. Tomlinson, B., et al. *The virtual raft project: a mobile interface for interacting with communities of autonomous characters*. in *CHI '05 extended abstracts on Human factors in computing systems*. 2005. Portland, OR, USA: ACM.
15. Yasuyuki, S. and M. Kenji. *AgentSalon: facilitating face-to-face knowledge exchange through conversations among personal agents*. in *Proceedings of the fifth international conference on Autonomous agents*. 2001. Montreal, Quebec, Canada: ACM.
16. Ståhl, O., et al. *A Mobile Fitness Companion*. in *The Fourth International Workshop on Human-Computer Conversation*. 2008.
17. Martin, A., *Dynamically Embodied Virtual Agents*, in *School of Computer Science and Informatics*. 2007, University College Dublin: Dublin. p. 320.
18. Breemen, A.v., X. Yan, and B. Meerbeek, *iCat: an animated user-interface robot with personality*, in *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*. 2005, ACM: The Netherlands.
19. Vala, M., et al., *ION Framework --- A Simulation Environment for Worlds with Virtual Agents*, in *Proceedings of the 9th International Conference on Intelligent Virtual Agents*. 2009, Springer-Verlag: Amsterdam, The Netherlands. p. 418-424.
20. Leite, I., et al. *Are emotional robots more fun to play with?* in *RO-MAN 2008: Proceedings of the 17th IEEE International Symposium on Robot and Human Interactive Communication*. 2008: IEEE Computer Society.
21. Field, A. and G. Hole, *How to design and report experiments*. 2003: Sage publications Ltd.