

Article

# MOVICLOUD: Agent-Based 3D Platform for the Labor Integration of Disabled People

Alberto L. Barriuso \*, Fernando De la Prieta, Gabriel Villarrubia González ,  
Daniel H. De La Iglesia  and Álvaro Lozano 

BISITE Digital Innovation Hub., University of Salamanca, Edificio Multiusos I+D+i, C/Espejo SN,  
37007 Salamanca, Spain; fer@usal.es (F.D.I.P.); gvg@usal.es (G.V.G.); danihiglesias@usal.es (D.H.D.L.I.);  
loza@usal.es (Á.L.)

\* Correspondence: albarriuso@usal.es; Tel.: +34-923-29-44-00

Received: 18 January 2018; Accepted: 23 February 2018; Published: 27 February 2018

**Featured Application:** The main application of this work is to perform 3D simulations of different work environments, which can help to establish the accessibility problems that may occur in the corresponding real-world environments.

**Abstract:** Agent-Based Social Simulation (ABSS), used in combination with three-dimensional representation, makes it possible to do near-reality modeling and visualizations of changing and complex environments. In this paper, we describe the design and implementation of a tool that integrates these two techniques. The purpose of this tool is to assist in creating a work environment that is adapted to the needs of people with disabilities. The tool measures the degree of accessibility in the place of work and identifies the architectural barriers of the environment by considering the activities carried out by workers. Thus, thanks to the use of novel mechanisms and simulation techniques more people with disabilities will have the opportunity to work and feel comfortable in the environment. To validate the developed tool, a case study was performed in a real environment.

**Keywords:** multi-agent system; agent-based simulation; 3D representation

---

## 1. Introduction

According to the World Health Organization [1], around 15% of the total world population suffers from some kind of physical disability. From the total number of disabled people, 2% to 4% experience serious difficulties in performing daily activities. At present, disability is a factor that often impedes people from enjoying equal work opportunities, despite having all the necessary qualifications. This is due to the employers' negative attitudes, the lack of information and poorly adapted working environments. Statistics show that this is a serious problem, in developed countries around 50–70% of disabled people in working age are unemployed (reaching 80–90% in developing countries) [2].

Our society is becoming increasingly aware of the challenges faced by people with disabilities, including difficulties in finding employment. As a result of this collective awareness, large institutions and governments have created proposals related to the employment of people with disabilities. Many of these proposals have been put into practice, especially since the declaration of the *Standard Rules on the Equalization of Opportunities for Persons with Disabilities* issued by the UN [3]. We can find various concrete regulatory frameworks, which aim to promote social justice and facilitate access to work for people with disabilities. Two important frameworks can be distinguished, the *European Disability Strategy 2010–2020* [4] and the *Americans with Disabilities Act* [5]. Not only governments have taken initiatives, many companies are also committed to including and promoting people with disabilities in the private sector. An example of this is the *ILO Global Business & Disability Network Charter* [6], signed by 11 large companies worldwide.

Despite these efforts, there is still a long way to go before people with disabilities are given equal opportunities in employment. Thus, it is necessary to continue providing advanced technological tools that will enable company human resource (HR) managers to integrate employees with disabilities in the workforce. For successful management, it is of course necessary for HR managers to have the necessary experience and knowledge. However, advanced tools and software models would facilitate their work and make it even more effective. Social machines are an adequate computational model, since they allow for the development of computer mechanisms which facilitate social studies [7] and social interaction between human beings and computer systems [8]. In short, social machines are capable of balancing the human input (experience) and computational systems (computational capacity) [9]. The MOVICLOUD platform is part of the framework of this proposal. In the framework of this research, the MOVICLOUD platform is proposed: a multi-agent system (MAS) based on virtual organizations (VO) with the capacity to perform social simulations over a 3D environment. The use of organizational MAS allows the implementation of social machines, facilitating the work of humans (HR departments) and machines for the accomplishment of common objectives (integration of disabled people). Likewise, it also allows the inclusion of advanced techniques and algorithms such as social simulation techniques and the modeling of human behavior.

In the framework of the labor integration of disabled people, simulations allow to discover in advance the problems that workers with disabilities will face before these problems happen. The two key elements to face this kind of difficulties are, on the one hand, the lack of accessibility of the buildings, and on the other hand, the need to adapt the different processes that are carried out in the company, so that they are adapted to people with disabilities but maintaining or increasing their level of efficiency or effectiveness.

To face these challenges, the MOVICLOUD platform proposes the use of a specifically developed 3D tool that allows to create three dimensional representations of the information extracted from an office environment. New mechanisms and techniques are employed to make realistic simulation models. The results of these simulations, which are interactive, facilitate the work of those responsible for HR, making decisions about the improvement of work processes that occur in the environment. For this purpose, a collaborative work framework between machines and humans is proposed, where experts can modify the simulations interactively, while the machines have the capacity to provide adequate and relevant information to the problem to be simulated, such as the identification of the different architectural barriers, the improvement of production processes, location of jobs, etc.

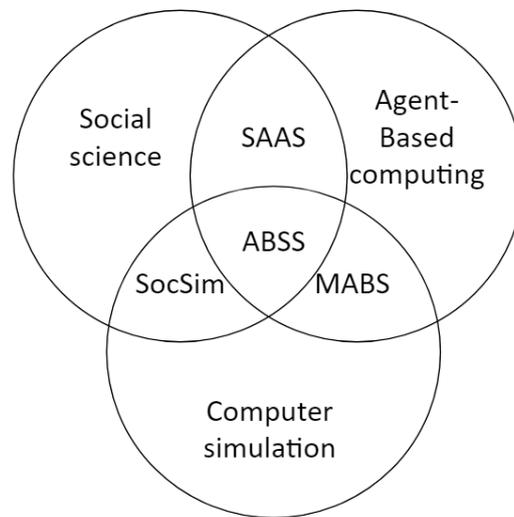
This paper is structured as follows: in Section 2 a review of the state of the art in social simulation based on agents and agent development platforms is made, Section 3 details the design of the platform presented in this article, Section 4 describes a case of study with which the platform has been validated and finally, in Section 5 the obtained conclusions are exposed, as well as the possible lines of future work.

## 2. Background

Performing simulations of heterogeneous systems where a large number of elements intervenes is a complex task, but has found in the agent-based modeling a technique that facilitates its design and implementation. For this reason, agent-based modeling has been widely used to model different active objects, such as vehicles [10,11], markets [12,13], people [14], industrial units [15], animals [16], health [17] etc. The phenomenon to be simulated is composed by a set of sequences, events or tasks that may exist in the environment, which will be configurable through the simulation model. These tasks are interrelated, since in the end, they depend on the order or time in which the events occur. The complexity lies in the fact that these models allow the simulation of changing and complex environments, so the use of open MAS is ideal, thus emerging the concept of Agent-Based Social Simulation (ABSS) [18].

ABSS is a discipline that deals with the simulation of social phenomena by using multi-agent computational models. Therefore, ABSS is a combination of social sciences, agent-based computing

and computational simulation. In recent times, other areas of study whose interests interact with each other have also emerged, as shown in Figure 1 [19].



**Figure 1.** Intersection of different areas within the Agent-Based Social Simulation (ABSS).

The use of the theory of agents in the framework of simulations is especially indicated when it is necessary to capture the different tasks, elements, objects or people of complex and dynamic environments. This is because these can be implemented without needing to have a great knowledge about the global interdependencies. Another reason is the ease they offer when building them when facing changes in the model, since generally it is only necessary to make local changes, not global [20]. This paradigm is used in many disciplines; some examples can be found in the field of health [21], urbanism [22], economics [23], architecture [24], criminology [25] or biology [26] among others.

The Foundation of Intelligent Physical Agents (FIPA) is an international non-profit organization whose main objective is the development of different standards to promote the use of intelligent agents. The specifications made by FIPA have been grouped into three sets: FIPA 97, FIPA 98 and FIPA 2000. In these groups, the characteristics of agents and multi-agent systems are defined in relation to their management, communication languages, interactions between agents and human, mobility or security among others. The architecture proposed by FIPA for the development of multi-agent systems is defined in terms of abstract designs, as an interface. In this way, multiple implementations of this architecture can be possible, attending to the interests of concrete developments. The pursued idea is to obtain different implementations of the same architecture, which defines common elements of the different systems, so that they operate under the same specification, in an interoperable way. There are several agent development platforms based on FIPA specifications such as Lightweight and Extensible Agent Platform (LEAP) [27], FIPA-OS [28], Reusable Task Structure-based Intelligent Network Agents (RETSINA) [29] or MOLE [30]. However, the most widespread platform is JADE (Java Agent Development Environment) [31]. It is a framework for the development of multi-agent systems whose objective is to simplify the development of this type of systems through a set of system services and auxiliary agents, making use of the FIPA Interaction protocols, which facilitates the communication between agents. JADE is implemented in Java and is distributed as free software with Lesser General Public License (LGPL). The communication architecture of JADE offers an efficient and flexible message communication, where the platform is responsible for creating and managing a queue of arrival Agent Communication Language (ACL) messages, which are private for each agent.

Currently, the idea of modeling systems through the iterations that occur between their agents is opening the way to a new trend: modeling the behavior of the system from an organizational point of view. Following this trend, it is necessary to define concepts of rules [32], norms and institutions [33]

and social structures [34], which arise from the need to have a higher level of abstraction-independent of the agent-, which explicitly defines the organization in which agents coexist. In this context, the concept of agent society emerges [35], as the set of interrelated and interacting artificial entities, which are governed by certain rules and conditions. The architectures used to model and build multi-agent systems based on organizations must support the coordination between agents, so that they can dynamically adapt to changes in their structure, objectives or interactions [36]. By doing this, it is possible to allow the coexistence of agents in shared environments, in such a way that they can carry out their objectives. When an agent is part of a society of agents, it takes a collectivist vision, so it must take into account the global functioning of the system, how some agents influence others, so that he does not focus on itself. The current trend towards computer developments with collaborative work environments, distributed knowledge management, has sparked interest in virtual organizations: a set of people and institutions that need to coordinate resources and services across different institutional boundaries [37]. There is a clear analogy between human societies and agents; we can define a human society as a formation or social entity with a number of members that can be specified and an internal differentiation of the functions that are performed by these members [38] and as an agent organization a collection of roles, which maintain certain relationships among themselves, and that take part in patterns of interaction with other roles in an institutionalized and systematic way [39].

On the one hand, ABSS has been applied with good results for the design of work methods [40] or training of people with disabilities [41] and on the other hand, VO have also been previously applied to similar scenarios such as the monitoring of elderly and disabled people in geriatric residences [42]. However, the use of this techniques (together or separated) has not yet been applied to the integration of people with disabilities in their working place. Though, considering the current state of the art, ABSS together with the social aspects that VO provide (analogous to the human societies), are suitable techniques for modeling work environments where there are people with disabilities, since they allow modeling large and distributed problems where environments are open and dynamic and a set of actors with different roles interact. By following this kind of approach to perform simulations, it will be possible experiment on a model, which if is good enough, will respond in a similar way to the studied system. This would allow to carry out different models of experiments on multiple occasions, permitting to test and evaluate the different configurations, besides of being able to include random components. Thanks to this, it is possible to analyze the simulation context from different perspectives, obtaining much more realistic results from the analysis. In the case of the labor integration of disabled people, these techniques can be applied to the simulation of different work scenarios in order to detect, from these simulations, which are the obstacles that these people may find in their work, thus being able to take preventive actions to eliminate these limitations.

Therefore, in view of the limitations observed in the existing platforms, within the framework of this work, the MOVICLOUD platform will incorporate a novel simulation environment, based on 3D techniques that allow ABSS to be made from advanced models of physical spaces, that would allow to perform simulations in a closer way to reality. Next, the design of this platform is detailed.

### 3. MOVICLOUD Platform

Zambonelli et al. [39] introduce the idea that MAS must accept the participation of agents of all types, regardless of architectures and languages. This statement goes to the extreme in the framework of social computing, where humans and machines work collaboratively [9]. However, it is necessary to determine rules or social rules for agents, since a priori their behavior is not trustful. Thus, the idea of modeling systems through iterations has evolved to the point of being necessary to perform an organizational model, where artificial societies exist [43] in which humans and machines collaborate to achieve the common objectives of the society. Following this trend, it is necessary to define concepts of rules [32], norms and institutions [33] and social structures [34], which arise from the need to have a level of abstraction, which explicitly defines the organization in which the agents they coexist. The current trend towards computer developments in collaborative work environments

and with distributed knowledge management, has generated interest in VO: a group of people and institutions that need to coordinate resources and services across different institutional boundaries [37]. There is a clear analogy between human and agents' societies; we can define a human society as a foundation or social entity with a certain number of members that can be specified and where an internal differentiation of the functions that are performed by these members is done [38], on the other hand, an agent organization is a collection of roles, which maintain certain relationships among themselves, and that take part in patterns of interaction with other roles in an institutionalized and systematic way [39]. Thanks to these capabilities provided by the VO of agents, it is possible to model social machines, creating artificial societies of humans and machines that share tasks, objectives and norms. Human and artificial agents in a MAS try to cooperatively carry out tasks in the environment of the system to which they belong. The tasks that can be performed within the system will be established by the knowledge and skills of the agents and the restrictions imposed by the environment.

Laying on this theoretical base, the development of a tool for modeling simulations of human processes that take place in different environments in a flexible and dynamic way has been carried out. The proposed system is based on a novel approach, consisting of the use of MAS based on VO and a three-dimensional representation environment. The combination of both strategies is key to the purpose of the simulations to be carried out since on the one hand, agents can implement different tasks, objectives, purposes, etc.; while the three-dimensional environment allows the physical modeling of environments. In this work, a simulation tool oriented to model productive processes for workers in office environments is presented. Thus, this approach allows simulations to be carried out on the work environment itself, as well as the visualization of the results of the simulations. The possibility of integrating the tools for the generic construction of 3D environments into a platform is a differentiating factor with respect to platforms such as MASON [44], Repast [45], Pangea [46] or Swarm [47] that allow the visualization of the interactions between the agents themselves, as well as with the environment. These models and tools allow to model a wide variety of environments and tasks. In fact, they have even been applied in previous studies, obtaining good results [48], however, although they can be used as a basis for the construction of simulations with 3D elements, they are not specifically designed for it, so they do not offer the proper utilities that facilitate the deployment of these simulations in an agile and simple way.

The main novelties included in the developed platform are (a) the possibility to easily model 3D environments using a tool based on *cloud computing*; (b) the use of virtual organizations that model the behavior of people in a work environment (making use of the FIPA ACL communication mechanisms provided by JADE); (c) the design of a visualization platform in the field of office environments, which offers a vision closer to reality than other generic agent platforms. In addition, through a case study is intended to validate the use of this tool to obtain useful information to reduce accessibility barriers found by disabled people based on simulations carried out in the three-dimensional environment.

In the following Sections, the presented architecture is described in detail: in Section 3.1 an overview of the components architecture is done, Section 3.2 describes the implementation of the organizational model, Section 3.3 shows what the structure of the agents that intervene in the simulations is like and finally, Section 3.4 describes how the communication between the agent platform and the three-dimensional visualization tool takes place.

### 3.1. Components of the Architecture

The general architecture of the system is made up of two large blocks: a MAS and a tool for designing and visualizing three-dimensional environments. On the one hand, the MAS is composed of a set of intelligent agents that allow to perform the simulation of the desired model. For the implementation of this component, the JADE platform [31] has been used, as it will be detailed later. The developed platform is oriented towards the modeling of the MAS as a VO of agents. From this perspective, it is able to define a structure, roles, and a set of rules that regulate the interactions between agents. To do this, a higher-level layer to JADE, responsible for managing the MAS, is included in the

system. Through this, it has been possible to provide the system with a set of additional functionalities that maximize the capabilities of MOVICLOUD, allowing a better interactivity and configuration on the part of human experts. These functionalities are detailed below:

- Ability to use **virtual organizations of agents**, which allow agents to adopt different topologies, such as hierarchies, oligarchies, groups, congregations, etc. [49]. VO allow the artificial representation of human societies in work environments. Thus, the structure of the multi-agent system in each simulation may vary, generating specific components depending on the environment to be modeled. Attending to the simulation that it is intended to be performed, is necessary that the multi-agent system would be able to adopt one of these topologies, so different types of agents must co-exist. JADE does not support the management of VO, so it has been necessary to introduce additional functionality with which to perform this task. To support this functionality, a database has been incorporated into the system in which the topology of the agents present in the system will be stored, making it possible to reuse it in the future.
- The tasks that the agents can carry out within the artificial societies that are generated within the framework of the simulations can vary, so that the objectives of the agents can be the same, but the decision-making process to carry them out may differ depending on the kind of disability. In this way, agents with advanced capabilities have been developed for **planning the distribution of tasks** according to the needs and configuration of the organizational structure. The user of the platform is responsible (through a series of functionalities implemented in the visualization tool) to model the different agents that act in representation of the elements that they want to model, thus allowing human-artificial collaborative work. For example, when defining an organization with a hierarchical structure, two kinds of agents will exist. The agents in charge of the decision-making are located at the higher level of the hierarchy—determining which tasks need to be carried out and who should carry them out—and the agents that have a more basic functionality—those in charge of carrying out the designated tasks—will be placed in the lower layers.
- **Communication with the 3D environment**, so that all the changes occurred can be reported between the agent platform and the three-dimensional environment in a bidirectional way.

Figure 2 shows in detail the different components present in the multi-agent system, as well as the relationships between them.

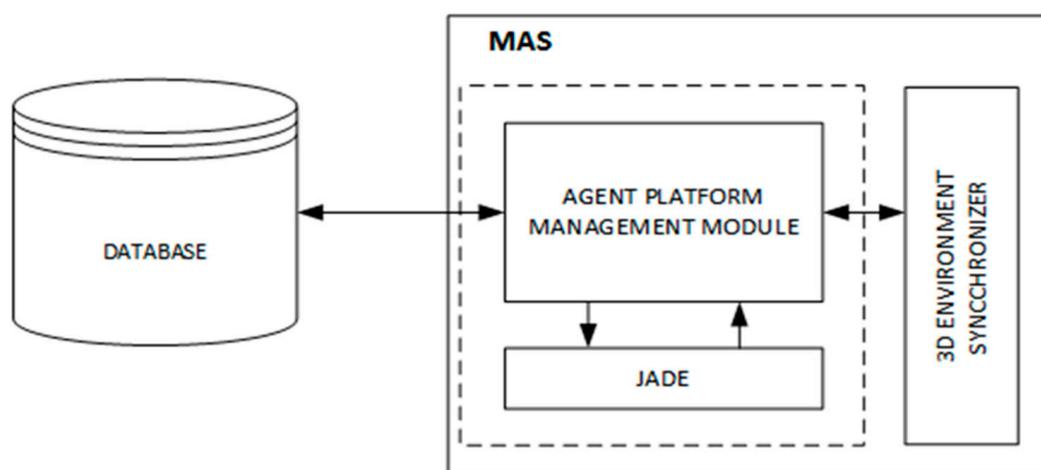


Figure 2. Multi-agent system (MAS) Components.

Finally, the **3D environment** that allows real information representation collected from the simulated environment is composed of an editor and a viewer. An organization is surrounded by an environment that is constantly changing. In this case the model of the environment is determined by the design phase of the building in which the simulation is developed. This process is carried out through a tool based on Unity 3D [50].

The **3D editing environment** (Figure 3) allows the modeling of virtual environments for urban areas (buildings, flats, apartments, gardens, parks, etc.) in three dimensions with a high level of realism and quality. This tool is able to generate 3D virtual environments in scale and define elements such as floors or walls. It also includes a set of default furniture models that can help to reduce the virtual environment modeling time. This editor uses different services from a *cloud computing* platform [51], being the main ones (i) a repository where the different 3D objects are stored, (ii) a repository of the scenarios created by the users of the platform and a (iii) user management service. Thanks to the possibility of remotely storing of the different Unity 3D packages that contain the 3D objects and textures used in the editing tool, the editor is able to obtain these models on demand. This allows the tool to be lighter, since it is not necessary that all the models offered by the platform are previously stored in the user's computer. There is a total amount of 14 different categories of 3D resources in the platform, but it is possible to include new categories if it is required. Besides, in order to facilitate the handling of different types of objects, a classification has been established according to their placement and behavior within the 3D editor:

- TYPE 0: Objects whose pivot is in the lower part of its mesh, that is, correspond to the objects that are placed at ground level.
- TYPE 1: Objects whose pivot is at the top of it, such as a lamp or any other object that hangs from the ceiling.
- TYPE 2: The pivot is on the side of the mesh to facilitate its placement on a wall.
- TYPE 3: The pivot is in its lower part, as in the case of type 0, including the possibility of placing these objects over other objects.
- TYPE 4: Objects of type 0 including an accessibility property of to them. That is, these objects will be used to perform the different interactions between avatars and furniture.
- TYPE 5: Objects of type 2 including the accessibility property.
- TYPE 6: Objects of type 3 including the accessibility property.

On the other hand, after completing the process of generating the virtual environments through the 3D editing tool, a json file is generated, in which all the environment characteristics are stored. By using these files, it is possible to generate the same scene in Unity, but without the need of store the unity project, (which is usually heavy in terms of storage). This lightweight file is sent to the *cloud computing* platform, so that, if the same scene has to be generated again, it is only necessary to download this file through the 3D editing tool or the 3D viewer, and 3D model is generated again using the 3D models that are provided by the platform, thus favoring the possible reuse of all resources by different users. The platform users can manage if the generated scenarios are public, private, or shared with certain users. To do this, thanks to the user management system, users of the platform can log into the system, accessing the list of projects for which they have access. Thanks to the use of the functionality offered by the *cloud computing* platform, several benefits are obtained such as a reduction in the amount of data to be stored by users, an improvement of the elasticity and scalability of the use of resources, or the dynamic adjustment to the load of the system and the needs on the part of the users of the platform. In addition, more effective use of computer resources can be made, since they can be shared among several entities.

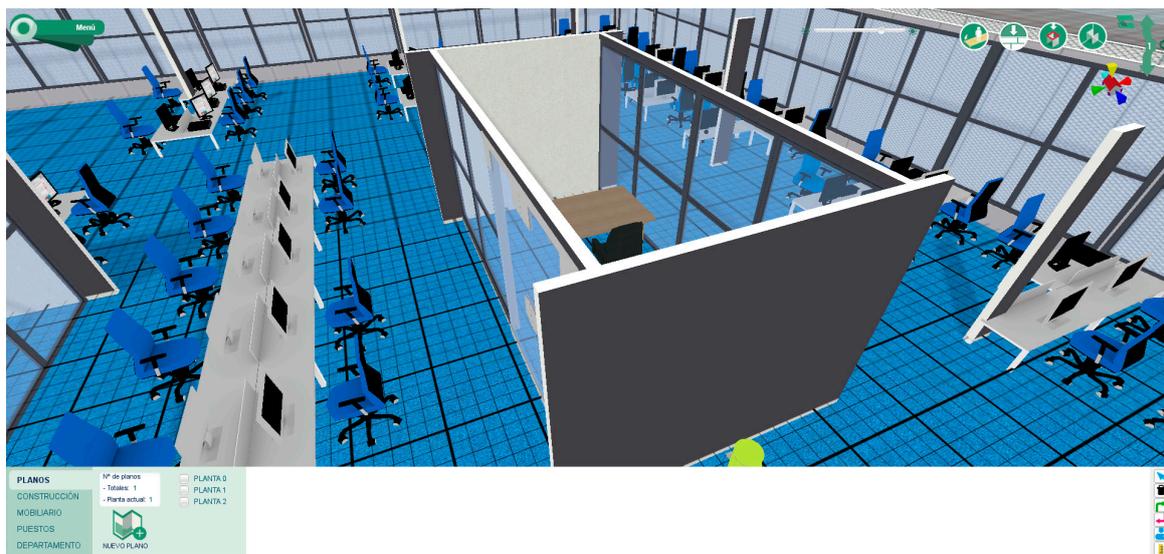


Figure 3. The 3D edition environment.

The **3D viewer**, which supports the representation and rendering of the environment and the features that were previously built with the 3D editor. It is also responsible for showing the different agents that are part of the system, making it possible to visualize the performance of the different activities of each agent. One of the main problems that must be faced in 3D scenes is to move one model from one point to another, so it is necessary to use an algorithm that calculates the route. In this case CritterAI [52] was used. It is a navigation system for Unity, which offers functionality related to the generation of navigation meshes, road search and movement management. This tool provides different options of the route calculation algorithm to be used: Dijkstra and A\*. These algorithms are used to calculate the lowest cost path between an origin and a destination node. The nodes on which these algorithms will search for the minimum cost path will be the ones generated in the navigation mesh.

It has been necessary to add synchronization mechanisms between the 3D viewer and the agent platform, so that there is a correspondence between what happens on the agent platform and what is displayed in this tool. In addition, this platform is the main way of interaction between humans and machines, allowing experts in HR to modify the simulations based on their objectives. The operations that can be carried out are the creation and elimination of agents and assignment of tasks. As discussed in the previous chapter, all the operations requested from the 3D environment are communicated to the agent platform, which will attend these requests and confirm or not its completion in order to update the visualization status accordingly. Figure 4 shows the set of modules that make up the 3D editing environment and 3D viewer, previously explained.

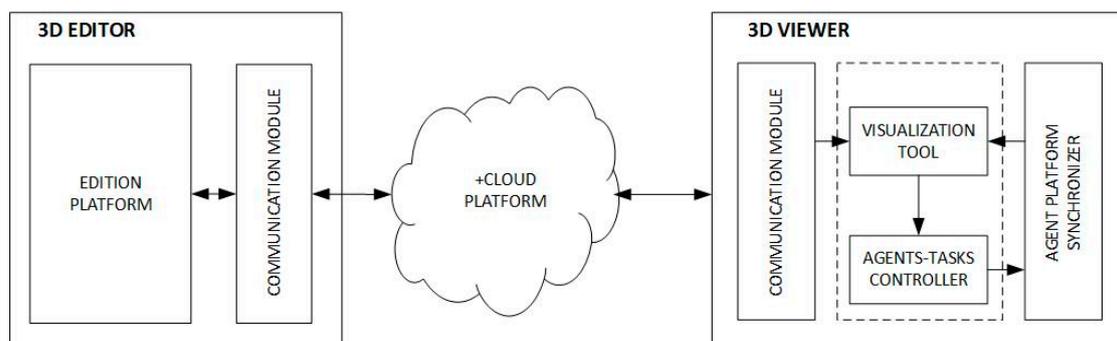


Figure 4. The 3D environment modules.

### 3.2. Implementation of the Organizational Model

After the analysis of the current state of the art in agent specification languages, platforms and specifications for the development of agents, we consider that the standard promoted by FIPA is flexible enough to define with guarantees any type of interaction between agents, so it adapts perfectly to the requirements that were initially marked to develop the platform. Since JADE [31] is a free open source system, it is well documented, and it is the most widely used solution within the development of multi-agent systems based on the FIPA standard, it has been taken as the basis for the implementation of the MAS, adapting it to support the management of VO. Thus, for the management of the existing agents in the VO, a specially designed database is used, which allows to store the agents' characteristics: the roles which have been assigned to them, the services they offer, as well as additional features that may be presented by the agents.

The control of the information stored in the database, and therefore of the organizational models used in each simulation, is encapsulated in a JADE agent called the Database Agent (DBA). This agent offers a set of services that facilitate all forms of access to the database, preventing inappropriate access by agents who should not be able to consult or modify the content of the database. Among the services offered by this agent, is to add agents, roles, services, etc., as well as query, modify or delete information. To access the different services, a module to check permissions is implemented, making the services accessible only to the agents that are specified.

Next, the structure of the database that will be used to store the structure of the VO is shown in Figure 5. Through a relational model, it is possible to define the different roles associated with the agents, as well as the tasks that can be carried out by an agent with a specific role. Additionally, three specific tables have been incorporated for the case study in which the proposed model will be evaluated, thus being able to define which agents intervene in each project, if they present some type of disability and associate roles with the different departments of a company.

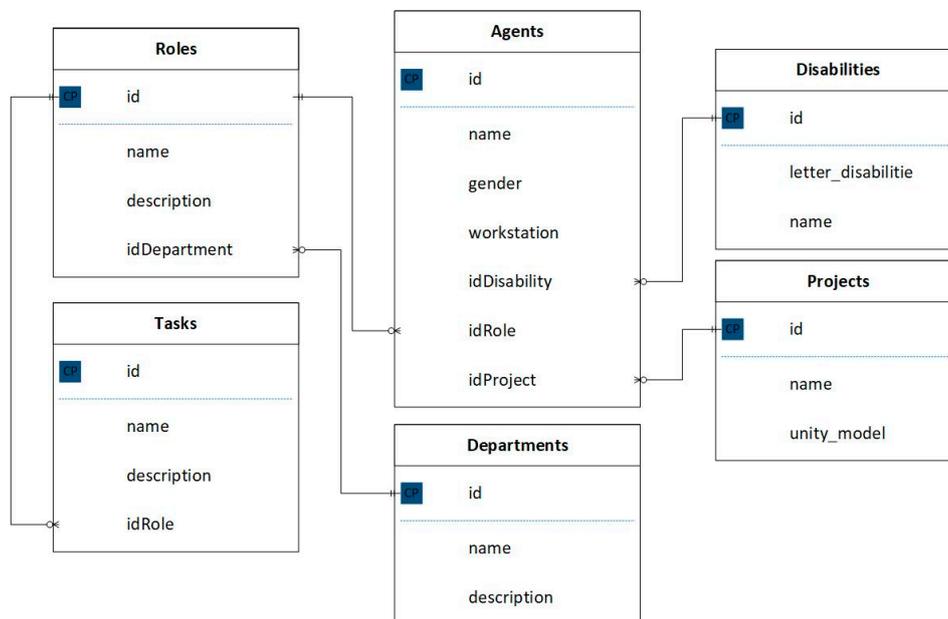


Figure 5. Virtual Organization database schema.

### 3.3. Structure of the Agents

Figure 6 shows in detail the structure of each of the agents, which consists of three large blocks: (i) a communication module, which will allow the agent to communicate with the other agents of the platform; (ii) a reasoning module, which is detailed below; and (iii) an agent communication module with the 3D environment, formed by a sensor and an effector.

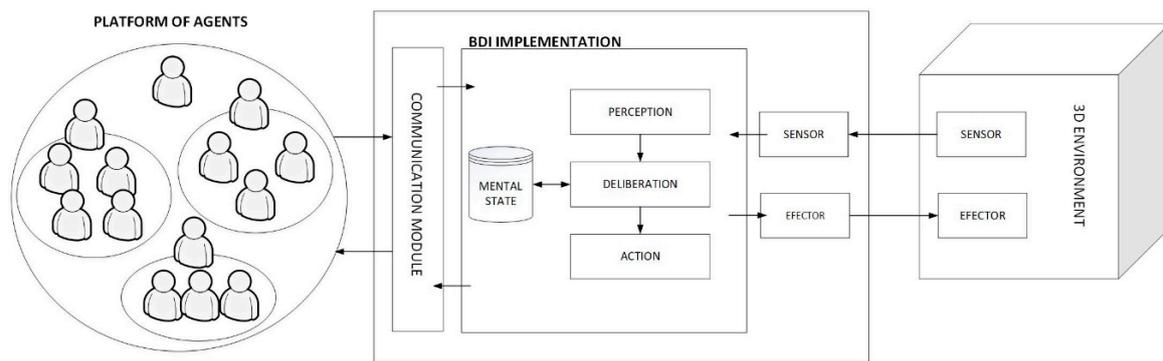


Figure 6. Agent structure.

For the definition and implementation of the reasoning module of agents within the platform, the belief–desire–intention (BDI) model [26] has been selected. The behavior of the system is therefore determined by the mental attitudes of the agents. For this reason, it is fundamental to achieve a correct operation of the system when the deliberation processes are limited by the resources.

In each iteration, the interpreter first reads an event queue and generates a list of options (possible goals to be achieved). Then it selects the best of these actions and adds them to the structure of intentions. If there is any intention to perform an atomic action, the agent executes it. Events that may occur during the interpreter’s execution cycle are added to the queue. Finally, the agent modifies the structures of intentions and desires by eliminating both the intentions and desires that have been successfully carried out and those that are not possible. With this model, deliberative systems are more appropriate for dynamic environments and are also more suitable for applications that work in real time. In this way, this model perfectly adapts to the requirements of the platform that is presented.

In addition to the reasoning module, it is necessary to introduce two additional modules that allow the agent to obtain information about the environment (**sensor**), as well as to be able to modify it (**effector**). It is necessary that these modules are distributed, having one component present in the agent’s own implementation within the agent platform, and another component within the 3D environment. In this way, the agent is able to be aware of the state of the three-dimensional environment that has been rendered for the simulation in progress, monitoring the changes that are produced in the virtual world.

The agent model presented above is designed to represent different actors involved in the simulation in representation of elements such as human beings, or similar. However, it is also necessary to introduce another type of agents: **environment agents**. These agents represent elements of the environment with which the other agents can interact directly. Examples of these agents can be telephones, elevators, photocopiers, etc. These agents are not aware of the environment around them, they will only be aware of their internal state, so they will lack of sensors. Any interaction with other agents cannot be initiated by an environment agent. When an agent requests an interaction with them, it will consult their internal state, and will communicate the pertinent information to the agent who finally, will determine according to this information if they can carry out the action that has previously been assigned to them.

The communication between agents is important in order to achieve the objectives of the system, where the objectives of the individual agents coexist. There is a set of stages in which the need for the existence of communication between agents is obvious. For example, when defining the problem to be solved, as well as its decomposition and distribution among the agents that make up the system. To carry out the communication between the agents, they are equipped with a FIPA ACL **communication module**. Thanks to the use of JADE it has not been necessary to redefine additional functionality, since this platform facilitates a complete implementation of these interaction protocols between agents.

### 3.4. Communication of the Agent Platform with the Visualization Tool

One of the essential tasks within the platform is to establish a communication between JADE and the 3D environment, so that there is a real correspondence between what happens in the agent platform and the visualization that will take place on the 3D platform. Within the agent platform, there is an agent in charge of managing communications in the JADE side.

For this purpose, the communication between the agent platform and the 3D environment is done through Transmission Control Protocol (TCP) sockets. In the Unity side, a module dedicated to the management of the requests made by the agent platform and vice versa is included. The tasks of creation, elimination and interaction between agents executed on the platform must be updated in the 3D environment, and the tasks of creation, elimination and interaction between agents executed from the 3D interface, must be carried out first in the agent platform for being later updated in Unity. For the implementation of the communication module hosted in Unity, IKVM.NET is used. It allows the execution of compiled Java code in MICROSOFT.NET. The communication will be divided into 2 blocks:

1. Sockets dedicated to carrying out the tasks in JADE and their corresponding update in Unity.
2. Sockets dedicated to the performance of the tasks initialized from the Unity interface, which must be done first in JADE and then updated in Unity.

For the simulation of tasks in JADE that will be updated later in Unity, three elements are used: two sockets, one for sending tasks and another for the confirmation of the performed tasks, and a table with which will keep a record of these tasks (Figure 7).

- **UnityTasks**, stores tasks that have not yet been updated.
- **JadeProducer**, it allows to send tasks to Unity through a TCP socket. In Unity, there will be a resource (**UnityConsumer**) which is listening to the requests that are being produced.
- **JadeConsumer**, opens a TCP socket to receive information regarding the performed tasks. In Unity, **UnityProducer** is in charge of sending the confirmation of the performed tasks.

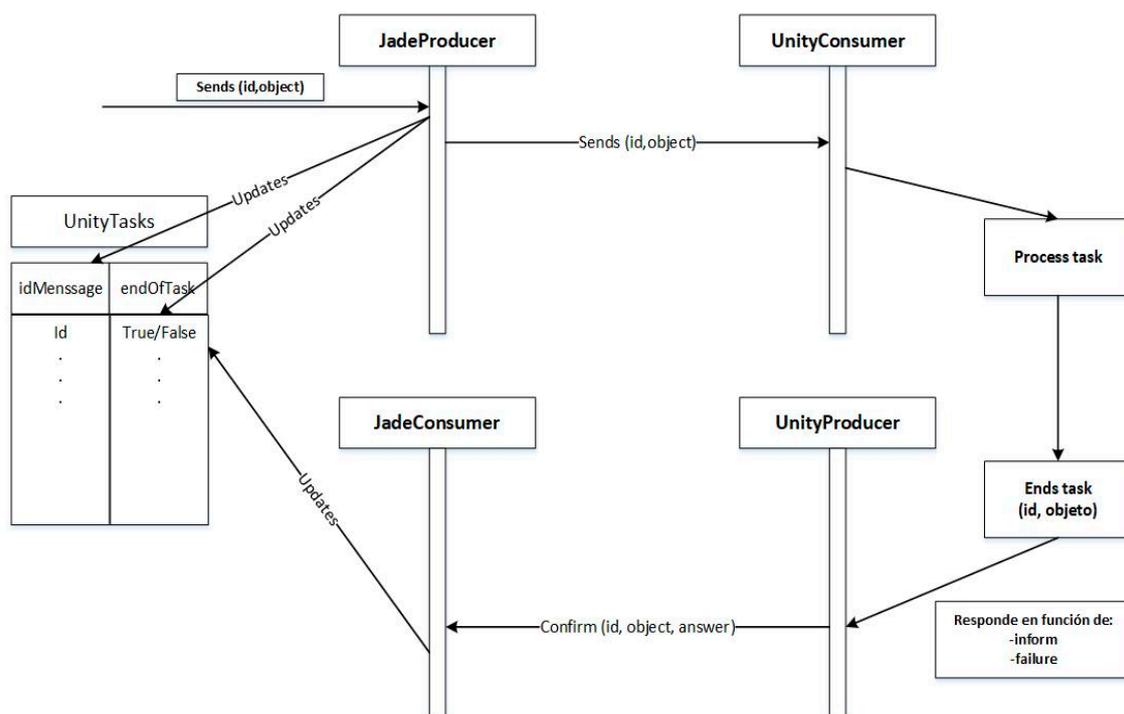


Figure 7. Java Agent Development Environment (JADE)-Unity communication.

Regarding the tasks that are performed in Unity and later updated in JADE, UnityTasks resides on the Unity side, since it is in charge of making requests and must verify that everything that is done in JADE is updated later in Unity. Figure 8 shows the scheme that follows this communication, where a TCP socket will be used. JADE is the server which will listen to the requests performed from Unity. Each request in Unity is a new client that requests a connection to the server, so throughout the execution several clients can coexist. The requests made from Unity can be of three different types: creation of agents, elimination of agents or assignment of tasks to agents.

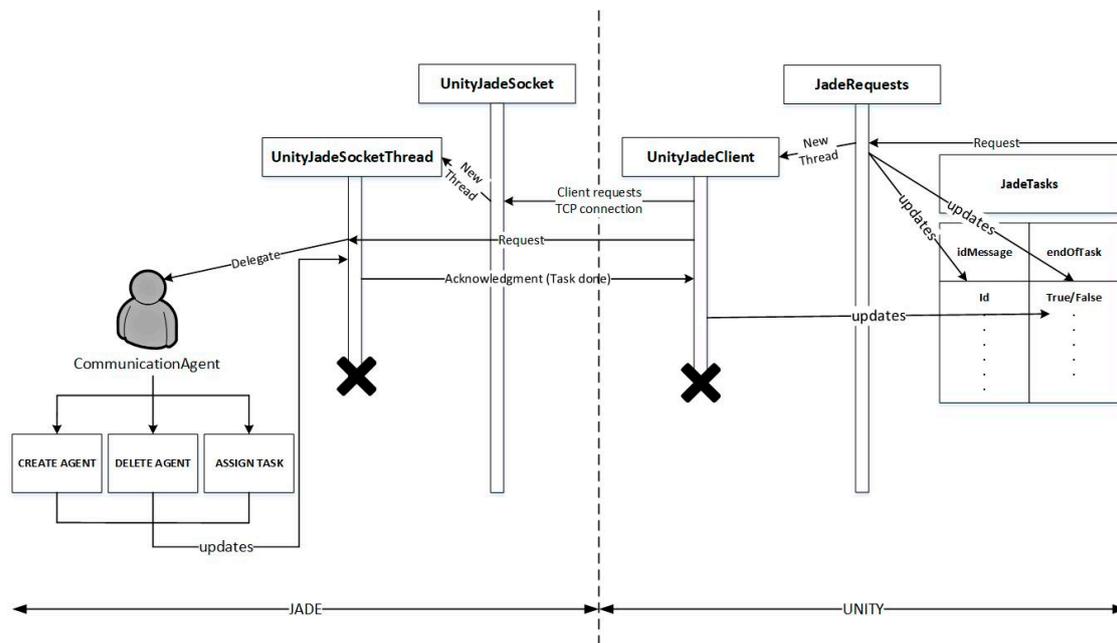


Figure 8. Unity-JADE communication.

The **Communication Agent (CA)** is in charge of attending the requests that arrive from Unity, carrying out the tasks that correspond according to the frame exchanged. The frame that is sent in each request contains the components shown in Figure 9.

Frame type	Id task	Agent name	Extra 1	Extra 2
------------	---------	------------	---------	---------

**Frame type:**

- “C” for creation frames
- “D” for deletion frames
- “T” for task assignment frames

**Id task:** identification number

**Agent name:** name of the agent which performs the request

**Extra 1:** this field meaning may vary depending on the frame type:

- “C” frames: kind of agent
- “D” frames: null value
- “T” frames: kind of task to be implemented

**Extra 2:** this field meaning may vary depending on the frame type:

- “C” frames: extra information attending to the case of study
- “D” frames: null value
- “T” frames: true/false, indicates if the task performance has been finished in Unity

Figure 9. Communication frame.

#### 4. Case Study

This section presents the case study which was conducted in a real environment to assess the performance of the developed system. The scope of the case study is the analysis of the accessibility of the different jobs in a real company: the offices of Indra Sistemas S.A. in Salamanca. By this way, through the performance of a set of simulations and the supervision of experts in HR and accessibility, knowledge about the different problems that disabled workers of the company may encounter can be extracted.

For the performance of the simulations, it was necessary to define the characteristics of the VO, in such a way that it would be able to modeling the processes carried out in the company as closely as possible to reality [53]. To do this, the different roles that agents can acquire, the services necessary for the proper functioning of the organization, the rules that will govern the society and the agents' own messages and iterations must be defined.

In addition, the environment in which the simulation will be carried out must be defined, in this case the building of the company under study. For this purpose, the 3D editing tool mentioned above was used. The 3D editing process begins from the plans of the building, so the environment in which the simulation takes place is as similar as possible to the real world. Furthermore, the disposition of the furniture inside the building must also be defined in order to evaluate, for each job, if each of the working places are accessible for people with different types of disabilities.

Afterwards, a series of simulations that show the behavior of the organization in different situations can be carried out, finally getting into the validation of the proposed model. First, we define the interaction model, analyzing the needs of the users who use the system and the way in which the exchange of information takes place. We define the following roles:

- **User:** represents the user or customer of the system. In this context, the user may be responsible for defining the distribution of the different elements in the building. Users can be the human resources responsible for assigning jobs to employees or the architect in charge of designing the building in which the professional activity will be carried out, in order to obtain information about the accessibility of the environment. User:
  - Is responsible for initiating the simulation process.
  - Will model the different agents that act as actors representing the workers of the company, as well as the tasks they will perform during the simulation.
  - Can access to the information that is generated after the processes that are simulated.
- **Manager:** this agent is responsible for carrying out the task planning that will be carried out by the whole organization. It may be present or not in the system, depending on the organizational structure of the company to be represented. Manager:
  - Generates the tasks that represent the objectives to be fulfilled by the whole organization.
  - Delegates the task distribution to the different area managers according to the nature of each task.
- **Department/Area managers:** these agents are in charge of planning the distribution of tasks assigned to the department or area that manages. Department managers:
  - Receive the tasks from the manager.
  - Plan the task distribution among the agents that make up the department based on their availability (agents with a lower number of tasks will have a higher priority) and capacities (not all agents offer the same services).
- **Workers:** agents who represent each of the workers involved in the simulation.

- The agents will have a stack where they store all the tasks they must perform, which have been previously assigned by the area manager.
- They will send information regarding the degree of success with which they have carried out each of them.
- All these agents have a set of elements that define their common behaviors and cognitive abilities. In addition to these common characteristics, the agents will have their own characteristics, defined according to their role within the organization and their disability.
- The assigned roles will determine the tasks they can perform and the specific behaviors for each one of them, so different executions can be modeled for the same task according to the type of disability of the worker.
- **Environment agents:** these agents represent those elements of the building with which the worker agents can interact directly. Some examples of this type of agents can be found in different elements of the environment such as telephones, elevators or photocopiers.
  - These agents are not aware of the environment around them, they are only aware of their internal state.
  - Any interaction with other agents will be initiated in any case by a worker agent.
  - When an agent requests its use, it will consult its internal status, checking if it is free and if it has the necessary resources so that the requested task can be carried out. It will communicate this information to the working agent, determining if it can complete the requested task.
- **Database Agent:** this agent offers a set of services that facilitate all forms of access that are held on the data base that stores the information regarding the structure of the VO. Database Agent:
  - Will take care of inadequate accesses by agents that should not be able to query or modify the contents of the database. These services will allow adding agents, roles, services, etc. to the database, as well as query, modify or delete information.
  - Includes a small module which is in charge of checking will require to check permissions, making the services accessible only to the specified agents.
- **Human resources agent:** The proposed VO can be classified within the so-called semi-open organization, since it will have a mechanism to control the admission of agents. In this way, it is necessary to make a request to evaluate the inclusion or not of an agent to society. The human resources agent is in charge of carrying out this task. The human resources agent:
  - Evaluates the candidatures of those agents which request access to the organization, approving the different proposals based on the vacancies that exist in the department to which the agent intends to access.
  - The number of spaces in each department will be established by the working places that are defined in the work environment. Each of the working place will be associated with a single department and an agent if they are occupied.
- **Communication agent:** it is responsible for communicating the multi-agent system with the visualization tool, in order to facilitate the message-passing between the two modules of the architecture.
  - It guarantees that there is a real correspondence between what happens in the agent platform and the visualization that takes place in the 3D viewer.
- **Supervising agent:** analyzes the different behaviors and evolutions of the agents involved in the simulation to carry out an analysis of the processes that take place in the organization.

- It will communicate with the other agents after the completion of a task, collecting whether they could carry it out or not.
- If a task has been carried out satisfactorily, it will be communicated which task was performed, which agent intervened, and the time and resources that have been used to carry it out. Otherwise, the agent involved shall indicate to the supervising agent the reasons why it could not be carried out.

Within the processes that are carried out in the company under study, three departments have been identified, according to the tasks that workers can perform: reception, administration and maintenance. The worker-type agents will have associated a series of tasks that they can carry out depending on the department to which they belong. We define the following tasks as well as well as possible additional environmental agents that may be necessary for each of them (see in Table 1. Relations between tasks, departments and environment agents).

**Table 1.** Relations between tasks, departments and environment agents.

Task Type	Department	Environment Agents
Make photocopies	Reception, Administration, Maintenance	Photocopier
Go to the bathroom	ALL	Toilet
Fire alarm	ALL	Siren
Answer Telephone	Reception, Administration	Telephone
Carry letters	Reception	-
Drink coffee	ALL	Coffee maker
Eat	ALL	Microwave
Throw Trash	Maintenance	Trash can
Collect documentation	Reception, Administration	-
PC	Reception, Administration	-
Check elevator	Maintenance	Elevator
Watering plants	Maintenance	-
Sweep	Maintenance	-
Collect garbage	Maintenance	Trash can, container

Regarding the relationship of a person with the physical environment, the activities that are carried out during its daily life are widely varied, but the accessibility difficulties that arise when doing them are repeated. Therefore, the activities are analyzed from the perspective of accessibility to detect what difficulties they generate in order to find an alternative. To that end, two components have been established for all the activities that workers can carry out: displacement and use. **Displacement** is the transfer from a point to the ideal place where to perform an action, which implies being able to move around the environment without the presence of limitations or obstacles. Two types of displacement can be differentiated: (i) *horizontal* (displacements made in rooms, corridors, etc.) and (ii) *vertical* (displacements that imply a change in height, such as steps, stairs or ramps). The **use** is the performance of the action itself. Two types of use are differentiated: (i) *preparation*: situation process or approach to the object to be used and (ii) *execution*: the performance of the desired activity; it is the final objective of the process. If the person has any limitation that hinders the performance of their activity with respect to an individual without disabilities, we can use these as a reference to define the specific needs of these individuals in order to project the buildings. Difficulties of different types may appear:

Attending the displacement:

- **Maneuver:** limitation of the ability to access to spaces and move within them.
- **Change of level:** difficulties that arise when overcoming unevenness.

Considering the use of spaces, the following difficulties may appear:

- **Reach:** when there are limitations in the possibilities of reaching objects and perceiving sensations.

- **Control:** difficulties that arise as a direct consequence of the loss of ability to perform actions or precise movements with the extremities.

In order to guarantee the movement within a building, it is necessary to define accessible routes that link the different spaces. These itineraries must meet three basic functional criteria:

- That they are flat or with a gentle slope.
- That they have a passage area free of obstacles: they must be wide enough to fit wheelchairs, and also be free of obstacles of medium height, which are dangerous for the blind people.
- That they have safe support and guidance elements: that is non-slip pavements, handrails that serve as support, differentiated textures for the blind, etc.

For each action, four functional objectives have been distinguished:

- That it can be carried out, also, by a seated person. Thus, it can be performed by wheelchair users, elderly people with difficulties, etc.
- Without leaving the site while it lasts. Most people with limitations use their upper limbs to help themselves with movement; therefore, they can hardly move and act at the same time.
- Disregarding fine joints. The faucets, switches, door handles, control elements, etc., must have a design that allows their manipulation by those who have lost strength and dexterity in the hands.

Once the building model and the interaction model have been defined, we can proceed to perform the simulation. To do this, the visualization tool must be deployed, which at the beginning of the process, has perform different operations: (i) loading the building model; (ii) generating of the navigation mesh; (iii) synchronizing with the agent platform. Figure 10 shows a screenshot of the simulation tool.



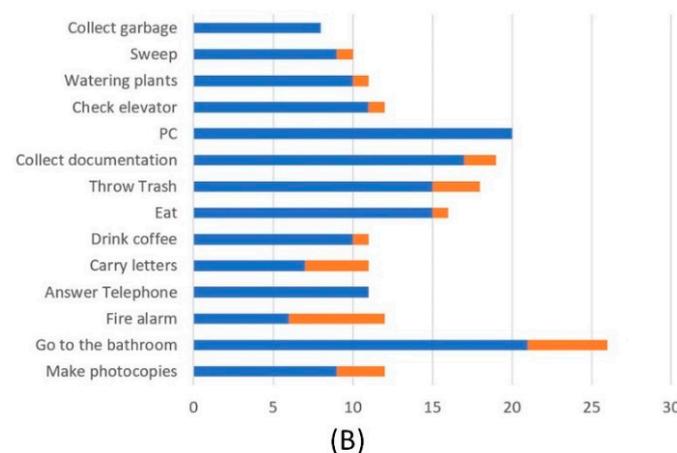
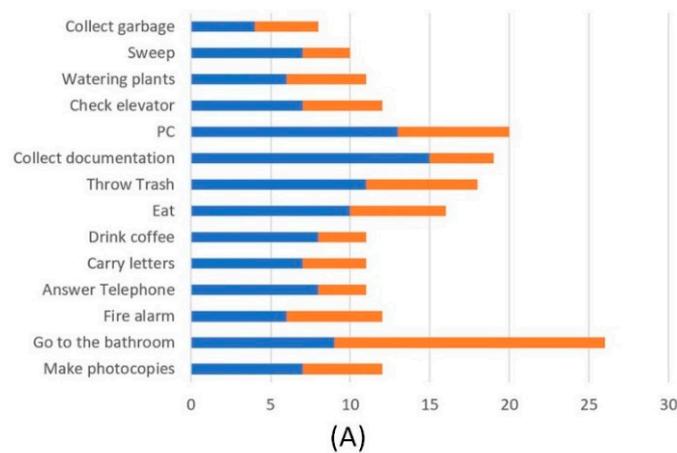
Figure 10. Simulation tool.

The execution of the simulation will depend on the set of tasks that the MAS has to perform, which may be specified by the user, or generated by the management agent, as well as the process of the tasks assignment. Overall, the simulation will be defined by the definition of:

- The set of agents responsible for each department:  $R = \{R_0, \dots, R_j\}$
- The set of worker agents:  $A = \{A_0, \dots, A_j\}$
- The set of tasks to be performed by the worker agents:  $T = \{T_0, \dots, T_k\}$
- The type of resources:  $l \in \{1, \dots, L\}$
- The set of resources (environment agents):  $E = \{E_0, \dots, E_m\}$
- The amount of available resources of n type at a certain t time:  $L_{n,t}$

The result space after the performance of a certain task by a worker agent has only two possible output states (success and failure):  $O \in \{0,1\}$ , where 1 represents the success of the performance of a task, and 0 represents its failure.

First, a simulation of 15 min is planned. For the initial set-up of the different agents that represent actual workers within the organization, a Human Resources manager established a simulation of a total of 8 agents, whose department was assigned proportionally to the amount of workers of each department that were working at the company –1 receptionist, 2 maintenances and 5 administrative. Three of them have reduced mobility problems, three are visually impaired and one is deaf. The assignment of the work places for each of the agents was made just according to the productive processes that take place in the organization, without taking the possible architectural barriers that they might find into account. Once the environment and the agents’ distribution and characteristics were set up, the simulation can take place in the 3D viewer. After the simulation, the tool generates a detailed report with the accessibility problems that the avatars have encountered, which have impeded the achievement of the tasks assigned to them. 118 were successful and 79 failed (Figure 11A). Some of the deficiencies found were: (i) The lack of adapted bathrooms in the building; (ii) working places not adapted for people with visual or hearing disabilities: impossibility to perform tasks such as answering the telephone, or being aware that the fire alarm is active; (iii) lack of ramps that enable the access to higher floors as an alternative to the elevator, which in case of breakdown would obstruct the access or exit to the building; (iv) limited access to certain work stations–spaces not large enough for access with wheelchairs or; (v) not accessible furniture elements–file cabinets, photocopiers.



**Figure 11.** Successful and failed tasks after simulations. (A) results after first simulation; (B) results after second simulation.

After the detection of these accessibility problems, all the information obtained from the tool was used by the Human Resources manager in order to perform several changes (both in the distribution of environment elements and workers) that could emerge into an improvement of the accessibility of the working place. First, the 3D editor was used to perform a redistribution of the elements of the furniture susceptible to improvement and the adaptation of the work places for users with auditory and visual deficiencies. Secondly, a relocation of the work places of those users of wheelchairs that could not perform their designed activities due to several displacement criteria such as maneuver, change of level, reach or control was made using the 3D viewer. After doing this, a new simulation with the same agents and task-assignment was made, obtaining as a result a total of 169 tasks performed satisfactorily and 28 failed (Figure 11B), going through a reduction of 40.1% of failed tasks in the initial simulation to 14.2% in the second simulation. Obviously, there are elements of the environment that will continue to limit their accessibility, such as the lack of access ramps or adapted toilets, but the possibility of improving the accessibility of the environment through the restructuring of employees' jobs or furniture in the critical points detected through the simulation. Since there are infinite possibilities of furniture distribution, it is difficult to establish an optimal solution regarding the accessibility levels of work environments. However, it would be possible, given a fixed distribution of the elements of the environment, to make all possible assignments of types of employees, jobs and tasks, thus being able to know what the optimal distribution of employees under a certain structure of the elements of the environment is.

## 5. Conclusions and Future Lines of Research

Once the simulation platform has been presented and evaluated, it can be affirmed that it allows the performance of simulations oriented to the labor integration of the disabled, allowing not only to analyze the tasks that are carried out in a work environment, but also the physical spaces where these tasks take place.

From an academic point of view, it has been possible to model a real organization through an VO of agents with self-adaptive capacities. This system has been implemented using JADE, but adapting this platform to specific needs. In the same way, a 3D environment has been created, and it is capable of modeling and enabling simulations in work environments. In the same way, a system that allows communication between both platforms, which allows sharing relevant information about communications, has been developed. In short, the implementation of MOVICLOUD allows to simulate different processes in a work environment, and it has been proven to be appropriated in order to establish the accessibility problems that may occur in it. In this sense, the representation of the employees working activity is performed, such and as presented in the case study. Thanks to the flexibility provided by the editing tool, which allows to quickly model virtually any office environment using specifically designed models for rapid deployment, MOVICLOUD is valid for simulations carried out in office environments. More specifically, thanks to the case study, the tool has been able to define accessibility problems and architectural constraints following specific criteria of displacement. This tool has been evaluated in a work office, proving to be a useful tool when it comes to identifying accessibility problems. The definition of the models to be used and the problems to be detected in each task based on defined criteria facilitates the inclusion of the study of different new tasks. That is why this tool allows to model and include new actions to be developed by the actors with ease. However, it should be noted that MOVICLOUD is aimed at simulating processes within office environments, so that its use is currently limited to this area. In the event that the tool is intended for simulation in an open environment and with non-human actors, it would be necessary to include additional functionality.

Based on the obtained results, we can conclude that the objectives set at the beginning of the project have been met. With the successful achievement of this work, it is possible to obtain useful information to facilitate the accessibility of work environments to disabled people. This has been possible through the use of agent-based social simulation techniques, using organizational concepts

to model the interactions between the actors and a three-dimensional environment. Although this tool initially was conceived for the detection of accessibility problems in office environments, thanks to the generic modeling of the platform and the use VO of agents, it is considered a valid tool for the simulation of any agent production process representing human beings in the field of office environments. Once the platform has been validated, it can be considered that within the scope of accessibility problems, it can be very useful at the time of: (i) early detection of failures in the design time of the buildings, facilitating new designs of the distribution of building elements, such as rooms, corridors, ramps or elevators to avoid in advance future problems related to the displacement of people with disabilities; (ii) prevent people with disabilities from facing accessibility problems before starting work at their place of work, being able to adapt their position and environment in a more appropriate way according to their capabilities (iii) improve the accessibility of people who are already working in a certain place.

Below, the different lines of research and future work that have been raised in the context of this work are shown:

- Include the possibility of making an abstraction of the building model through ontologies, so that the information of the environment does not only reside in the three-dimensional environment, but the agent platform also has knowledge of it.
- Allow the development of different three-dimensional environments, not only limited to representation in buildings.
- Improve the extraction of knowledge by improving the algorithms that intervene in the accessibility recommendations, offering real-time statistics and a report generation services.
- Include additional functionality to the visualization, so that there are alternatives when extracting information about the processes, such as graphs that show how the relationships between those agents interacting with each other or the generation of heat maps that show which are the routes taken by each worker in the three-dimensional environment.
- Perform stress tests to check the limitations of the platform in relation to communication processes and the size of the 3D environments. In this way, it will be possible to estimate the maximum number of agents and the characteristics of the 3D model supported by the platform over different hardware infrastructures.

**Acknowledgments:** This work has been supported by Cátedras de Tecnologías Accesibles of INDRA and Fundación Adecco. The research of Alberto L. Barriuso and Daniel H. de la Iglesia has been co-financed by the European Social Fund (Operational Programme 2014–2020 for Castilla y León, EDU/128/2015 BOCYL). Álvaro Lozano is supported by the pre-doctoral fellowship from the University of Salamanca and Banco Santander.

**Author Contributions:** Alberto L. Barriuso and Fernando De la Prieta conceived and designed the multi agent system, while Daniel H. De La Iglesia and Álvaro Lozano have contributed with the development and implementation of the system; Gabriel Villarrubia has designed and coordinated the case study. All authors have participated in the writing process of the article.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. World Health Organization. *World Report on Disability*; WHO: Geneva, Switzerland, 2011.
2. Parmenter, T. *Promoting Training and Employment Opportunities for People with Intellectual Disabilities: International Experience*; International Labour Organization: Geneva, Switzerland, 2011.
3. *Standard Rules on the Equalization of Opportunities for Persons with Disabilities*; Technical Report; United Nations: New York, NY, USA, 1994.
4. European Commission. *Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of the Regions*; European Disability Strategy 2010–2020: A Renewed Commitment to a Barrier-Free Europe (COM/2010/0636); European Commission: Brussels, Belgium, 2010.

5. *Americans with Disabilities Act of 1990*; S.933—101st Congress; The United States Congress: Washington, DC, USA, 1990.
6. International Labour Organization. *The ILO Global Business & Disability Network Charter*; International Labour Organization: Geneva, Switzerland, 2015.
7. Wang, F.-Y.; Carley, K.M.; Zeng, D.; Mao, W. Social computing: From social informatics to social intelligence. *IEEE Intell. Syst.* **2007**, *22*, 2. [[CrossRef](#)]
8. Erickson, T.; Kellogg, W.A. Social translucence: An approach to designing systems that support social processes. *ACM Trans. Comput. Interact.* **2000**, *7*, 59–83. [[CrossRef](#)]
9. Robertson, D.; Giunchiglia, F. Programming the social computer. *Philos. Trans. R. Soc. A* **2013**, *371*, 20120379. [[CrossRef](#)] [[PubMed](#)]
10. Barthélemy, J.; Carletti, T. An adaptive agent-based approach to traffic simulation. *Transp. Res. Procedia* **2017**, *25*, 1238–1248. [[CrossRef](#)]
11. HÖrl, S. Agent-based simulation of autonomous taxi services with dynamic demand responses. *Procedia Comput. Sci.* **2017**, *109*, 899–904. [[CrossRef](#)]
12. Aliabadi, D.E.; Kaya, M.; Sahin, G. Competition, risk and learning in electricity markets: An agent-based simulation study. *Appl. Energy* **2017**, *195*, 1000–1011. [[CrossRef](#)]
13. Immonen, E. Simple agent-based dynamical system models for efficient financial markets: Theory and examples. *J. Math. Econ.* **2017**, *69*, 38–53. [[CrossRef](#)]
14. Kamara-Esteban, O.; Azkune, G.; Pijoan, A.; Borges, C.E.; Alonso-Vicario, A.; López-de-Ipiña, D. MASSHA: An agent-based approach for human activity simulation in intelligent environments. *Pervasive Mob. Comput.* **2017**, *40*, 279–300. [[CrossRef](#)]
15. Terán, J.; Aguilar, J.; Cerrada, M. Integration in industrial automation based on multi-agent systems using cultural algorithms for optimizing the coordination mechanisms. *Comput. Ind.* **2017**, *91*, 11–23. [[CrossRef](#)]
16. Anderson, J.H.; Downs, J.A.; Loraamm, R.; Reader, S. Agent-based simulation of Muscovy duck movements using observed habitat transition and distance frequencies. *Comput. Environ. Urban Syst.* **2017**, *61*, 49–55. [[CrossRef](#)]
17. Montagna, S.; Omicini, A. Agent-based modeling for the self-management of chronic diseases: An exploratory study. *Simulation* **2017**, *93*, 781–793. [[CrossRef](#)]
18. Davidsson, P. *Multi Agent Based Simulation: Beyond Social Simulation*; Springer: Berlin/Heidelberg, Germany, 2000; pp. 97–107.
19. Davidsson, P. Agent Based Social Simulation: A Computer Science View. *J. Artif. Soc. Soc. Simul.* **2002**, *5*, 1–7.
20. Borshchev, A.; Filippov, A. From system dynamics and discrete event to practical agent based modeling: Reasons, techniques, Tools. In Proceedings of the 22nd International Conference of the System Dynamics Society, Oxford, UK, 25–29 July 2004; p. 45.
21. Venkatramanan, S.; Lewis, B.; Chen, J.; Higdon, D.; Vullikanti, A.; Marathe, M. Using data-driven agent-based models for forecasting emerging infectious diseases. *Epidemics* **2017**. [[CrossRef](#)] [[PubMed](#)]
22. Haman, I.T.; Kamla, V.C.; Galland, S.; Kamgang, J.C. Towards an Multilevel Agent-based Model for Traffic Simulation. *Procedia Comput. Sci.* **2017**, *109*, 887–892. [[CrossRef](#)]
23. Han, T.; Zhang, C.; Sun, Y.; Hu, X. Study on environment-economy-society relationship model of Liaohe River Basin based on multi-agent simulation. *Ecol. Model.* **2017**, *359*, 135–145. [[CrossRef](#)]
24. Wagner, N.; Agrawal, V. An agent-based simulation system for concert venue crowd evacuation modeling in the presence of a fire disaster. *Expert Syst. Appl.* **2014**, *41*, 2807–2815. [[CrossRef](#)]
25. Devia, N.; Weber, R. Generating crime data using agent-based simulation. *Comput. Environ. Urban Syst.* **2013**, *42*, 26–41. [[CrossRef](#)]
26. De Almeida, S.J.; Ferreira, R.P.M.; Eiras, Á.E.; Obermayr, R.P.; Geier, M. Multi-agent modeling and simulation of an *Aedes aegypti* mosquito population. *Environ. Model. Softw.* **2010**, *25*, 1490–1507. [[CrossRef](#)]
27. Bergenti, F.; Poggi, A. *LEAP: A FIPA Platform for Handheld and Mobile Devices*; Springer: Berlin/Heidelberg, Germany, 2002; pp. 436–446.
28. Poslad, S.; Buckle, P.; Hadingham, R. The FIPA-OS agent platform: Open source for open standards. In Proceedings of the 5th International Conference and Exhibition on the Practical Application of Intelligent Agents and Multi-Agents, Salamanca, Spain, 4–7 July 2000; Volume 355, p. 368.

29. Sycara, K.; Pannu, A.; Williamson, M.; Zeng, D.; Decker, K. Distributed intelligent agents. *IEEE Expert* **1996**, *11*, 36–46. [[CrossRef](#)]
30. Strasser, M.; Baumann, J.; Hohl, F. Mole: A Java based mobile agent system. In Proceedings of the 2nd ECOOP Workshop on Mobile Object Systems, Linz, Austria, 8–9 July 1997; pp. 28–35.
31. Bellifemine, F.; Poggi, A.; Rimassa, G. JADE—A FIPA-compliant agent framework. Proceedings of PAAM, London, UK, 19–21 April 1999; pp. 97–108.
32. Zambonelli, F. Abstractions and infrastructures for the design and development of mobile agent organizations. In Proceedings of the International Workshop on Agent-Oriented Software Engineering, Limerick, Ireland, 10 June 2001; pp. 245–262.
33. Esteva, M.; Rodríguez-Aguilar, J.A.; Sierra, C.; Garcia, P.; Arcos, J.L. On the formal specifications of electronic institutions. *AgentLink* **2001**, *1991*, 126–147.
34. Van Dyke Parunak, H.; Odell, J. Representing social structures in UML. In Proceedings of the Fifth International Conference on Autonomous Agents, Montreal, QC, Canada, 29 May 2001; pp. 100–101.
35. Annunziato, M.; Pierucci, P. The emergence of social learning in artificial societies. Proceedings of Workshops on Applications of Evolutionary Computation, Essex, UK, 14–16 April 2003; pp. 293–294.
36. Dignum, M.V. A Model for Organizational Interaction: Based on Agents, Founded in Logic. Ph.D. Thesis, Utrecht University, Utrecht, The Netherlands, 2004.
37. Foster, I.; Kesselman, C.; Tuecke, S. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *Int. J. High Perform. Comput. Appl.* **2001**, *15*, 200–222. [[CrossRef](#)]
38. Peiro, J.M.; Prieto, F.P.; Zornoza, A.M. Nuevas tecnologías telemáticas y trabajo grupal. Una perspectiva psicosocial. *Psicothema* **1993**, *5*, 287–305.
39. Zambonelli, F.; Jennings, N.R.; Wooldridge, M. Developing multiagent systems: The Gaia methodology. *ACM Trans. Softw. Eng. Methodol.* **2003**, *12*, 317–370. [[CrossRef](#)]
40. Rodríguez, S.; Julián, V.; Bajo, J.; Carrascosa, C.; Botti, V.; Corchado, J.M. Agent-based virtual organization architecture. *Eng. Appl. Artif. Intell.* **2011**, *24*, 895–910. [[CrossRef](#)]
41. Bajo, J.; Fraile, J.A.; Pérez-Lancho, B.; Corchado, J.M. The THOMAS architecture in Home Care scenarios: A case study. *Expert Syst. Appl.* **2010**, *37*, 3986–3999. [[CrossRef](#)]
42. Zato, C.; Rodríguez, S.; Tapia, D.I.; Corchado, J.M.; Bajo, J. Virtual Organizations of agents for monitoring elderly and disabled people in geriatric residences. In Proceedings of the 2013 16th International Conference on Information Fusion (FUSION), Istanbul, Turkey, 9–12 July 2013; pp. 327–333.
43. Davidsson, P.; Johansson, S. On the potential of norm-governed behavior in different categories of artificial societies. *Comput. Math. Organ. Theory* **2006**, *12*, 169–180. [[CrossRef](#)]
44. Luke, S.; Cioffi-Revilla, C.; Panait, L.; Sullivan, K. Mason: A new multi-agent simulation toolkit. In *Proceedings of the 2004 Swarmfest Workshop*; George Mason University: Fairfax, VA, USA, 2004; Volume 8, pp. 316–327.
45. North, M.J.; Howe, T.R.; Collier, N.T.; Vos, J.R. The repast symphony runtime system. In Proceedings of the Agent 2005 Conference On Generative Social Processes, Models, and Mechanisms, Argonne, IL, USA, 13–15 October 2005; Volume 10, pp. 13–15.
46. Zato, C.; Villarrubia, G.; Sánchez, A.; Barri, I.; Rubión, E.; Fernández, A.; Rebate, C.; Cabo, J.A.; Álamos, T.; Sanz, J.; et al. *PANGEA—Platform for Automatic coNstruction of orGanizations of intElligent Agents*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 229–239.
47. Gilbert, N.; Terna, P. How to build and use agent-based models in social science. *Mind Soc.* **2000**, *1*, 57–72. [[CrossRef](#)]
48. García, E.; Rodríguez, S.; Martín, B.; Zato, C.; Pérez, B. *MISIA: Middleware Infrastructure to Simulate Intelligent Agents*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 107–116.
49. Dignum, V. *Handbook of Research on Multi-Agent Systems: Semantics and Dynamics of Organizational Models*; Information Science Reference; IGI Global: Hershey, PA, USA, 2009.
50. Unity Official Web Page. Available online: <https://unity3d.com/es/> (accessed on 22 September 2017).
51. La Prieta, F.; Rodríguez, S.; Corchado, J.M.; Bajo, J. Infrastructure to simulate intelligent agents in cloud environments. *J. Intell. Fuzzy Syst.* **2015**, *28*, 29–41.

52. CritterAI Documentation. Available online: <http://www.critterai.org/projects/cainav/doc/> (accessed on 22 September 2017).
53. Corchado, J.M.; Corchado, E.S.; Aiken, J.; Fyfe, C.; Fernandez, F.; Gonzalez, M. Maximum Likelihood Hebbian Learning Based Retrieval Method for CBR Systems. In *Case-Based Reasoning Research and Development*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 107–121.



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).