Published in: Journal of Computer and System Sciences, Elsevier, 81, pp. 585-598, 2015.

A model for managing interactions between tangible and virtual agents on an RFID interactive tabletop: case study in traffic simulation

Yoann Lebrun[b,a,*], Emmanuel Adam[a], René Mandiau[a], Christophe Kolski[a]

[a] Univ Lille Nord de France, F-59000 Lille, France,
UVHC, LAMIH, F-59313 Valenciennes, France,
CNRS, UMR 8201, F-59313 Valenciennes, France
[b] Play Research Lab - CCI-Grand Hainaut,
3 avenue Sénateur Girard,
BP 80577, F-59308 Valenciennes Cedex, France

Abstract

Interactive tables are more frequently being suggested to support collaborative and co-localized applications. They can be tactile, tangible or mixed. New sensory entries enable users to manipulate different types of tangible objects. The purpose of this article is to propose a global interaction model that associates multiagent system concepts with interactive table entities. The model incorporates simultaneously usable entities: virtual and tangible entities. The virtual entities can be seen through video projection or the use of a screen integrated into the interactive table. The tangible entities are physical objects that can be manipulated by one or more users around the table. These objects are detected through dedicated sensors. A case study illustrates the proposal. The study incorporates the management of a road traffic simulator using the *TangiSense* interactive table; this table is equipped with RFID technology. The illustrations mainly show communication between the different agents involved in the simulation.

Keywords: Multiagent System, Situated Agents, RFID, Interactive table, Tangible object, Virtual object, TangiSense

1. Introduction

Simulation is a tool used for analyzing and designing in many diverse areas [5]. For many years, simulators have been proposed in numerous application fields and work situations, such as transportation [49], medicine [44] and ecology [39]. Most simulations make use of a standard software and hardware architecture. This architecture chiefly uses standard sensory tools: <screen, keyboard and mouse > [15, 56]. However, many simulators require experts for correct and efficient use. To improve the global functioning of a system, such simulation tools must allow collaboration to take place between various users with different profiles and work strategies. These users may need to communicate about a common objective and work together to make mutual decisions after studying different situations and alternatives. For example, architects, electrical engineers and security specialists may need to envisage different solutions to fit-out a building; however, these professionals are most likely not experts in complex simulator functioning. Consequently, it is necessary to propose more user-friendly and interactive approaches that allow people from different domains to engage in intuitive manipulation. With the spectacular evolution in technology, computers have become smaller and smaller [54]. This has led to the appearance of new interaction platforms, such as PDAs, tablets, smartphones

*Corresponding author. Phone: +33 3 2751 1496.
Email address: yoann.lebrun@univ-valenciennes.fr,y.lebrun@grandhainaut.cci.fr (Yoann Lebrun)

and interactive tables. Human body movements (mainly using the fingers) can substitute for a device to enable a user to interact with a machine. There are also new platform types that take into consideration the manipulation of everyday, generally manipulable objects. Interactive tables offer new possibilities by detecting tangible objects as well as interactions between these objects and users [26, 38].

Such interactive approaches are very different from other approaches to computer use. Although users interact with computers on an individual basis, several communicating users can exploit interactive tables. The technology used can be intuitive, i.e., use more natural interactions [16], enabling unfamiliar users to manipulate objects as well. Interactive table use is being revolutionized by new forms of human-machine interaction, which imply different approaches to designing applications. The reasons for doing this research were to manage virtual and tangible entities evolving on an interactive table with agent. This concept allows entities to interact and be autonomous in decision-making, the actions they take and the knowledge they acquire. The aim of this paper is to build new agent-based applications with natural interactions on interactive tables where users are not just passive observers of the application in use. Users can participate and play a role in such applications by manipulating tangible objects. Users can simultaneously interact with the applications through the table; in other words, users can collaborate around the table.

This paper expands on [32] by more precisely describing the principles of interaction between tangible and virtual agents on interactive tables. It provides a detailed explanation of interactions between agents after a user performs an action on the table with a tangible object. A case study highlights these principles of communication. This paper is organized as follows:

- The second section, called Related Work, explains interactive table technologies, and mainly tactile or tangible interactions. We focus on tangible interaction. Furthermore, this section covers the technical and applicative characteristics of these interactive media. We intend to make the applications more intuitive and realistic using behavior for the entities evolving in these interactive media. To do this, we analyze the different types of recognition and processing systems that would help adapt an application to the different characteristics of the context in which the objects are manipulated.

- In the third section, we propose a model that enables interaction and communication to take place between (a) the various entities comprising the application and (b) the users. This description identifies the sensory inputs of the interactive medium and then adapts the behavior of the entities/agents of a multiagent system to the environment in question.

- In the fourth section, a case study is presented. It illustrates how objects react to environmental modifications and how objects can interact with each other. Our application is a road traffic simulation. This application is instantiated on an interactive table called *TangiSense*. This table exploits RFID technology. This technology enables road signs provided with "RFID" tags to be detected. An agent's association with the entities enables the objects to interact and communicate with each other as well as with the simulated vehicles displayed on the table's surface.

- Finally the paper ends with a conclusion as well as suggestions for future research on the evolution of multiagent systems applied to these media.

2. Related work

2.1. Interactive tables as new interaction platforms

Ambient computing focuses on issues of establishing an "Intelligent" daily space that provides access to information or services in a physical environment. Ambient computing is present in our daily life and offers different services while hiding the complexity of the technology required for the user [1, 9, 54] (for example, in a car with Emergency Brake assist or automatically activated headlights). In everyday life, a user can interact anytime, anywhere, using several platforms connected to the rest of the world. This connection is mainly ensured via Internet and wireless technologies [25, 37, 52]. Since these technologies are constantly evolving, new interaction platforms are used with increasing frequency. In this article, we focus on interactive tables as an interaction platform. Such tables are considered innovative interaction medium. They do not require a mouse or a keyboard to ensure Human-system Post-WIMP (Windows, Icons, Menus, Pointer) interaction [12].

Interaction through "Multi-touch" devices enables users to interact with material entities using several contact points. For instance, interaction can be based on the use of several fingers (the "Multi-touch" concept) or a single finger (the "One-touch" concept: a single contact point that can be assimilated with the use of a mouse). In addition, interaction can be based on sound capturing devices [42]. The user interface can be projected onto the interactive table and then exploited by several users simultaneously through digital content. Compared with traditional interaction platforms, the main advantage of an interactive table is the social interaction it provides [43].

Using applications on interactive tabletops occurs in increasing numbers of domains, including gaming [34, 47], commerce [11, 53] (mainly catering), simulation [17, 24] and education [23, 28, 33].

One of the most widely used and instanced domains is music. As an example, a visual and auditory demonstrator is described in [18]; it is rapidly and easily handled by a set of users. This application is designed to provide users with a simple user interface for creating music compositions by manipulating important (and often considered difficult to master) concepts, such as rhythm and instruments.

2.2. Tangible interaction

Ishiii and Ullmer [16] present tangible user interfaces as user interfaces that enable real (tangible) objects integrated into a virtual environment to be manipulated. For Blackwell et al. [8], "Tangible User Interfaces (TUIs) are those in which physical objects are used to represent and control computational abstractions".

Combining an interactive table and a tangible user interface, which makes it possible to overlay our perception of reality with a virtual model, generates new interactive tables with tangible surfaces [35]:

- ReacTable [18]: This is an electroacoustic musical instrument. It enables users to study musical performances as they are taking place. ReacTable combines a tangible user interface with techniques such as modular synthesis, visual programming and visual feedback (thanks to the camera).

- SMART Table [1] is an interactive learning system. It allows groups of users to work simultaneously on the same surface. Information is captured using a camera located under the table's surface. The table is intuitive: young children can use it without receiving any instructions. It is possible to create numerous learning activities using toolboxes, which are used to personalize and create new activities. Activities can be redefined according to the learning capacities of the children.

- Table-robots [6] is used to study interactions between mobile robots and models for marking / reading / processing in the environment. Real robots evaluate in an active and graphical environment that displays information and interacts. The instanced applications are inspired by ant algorithms dedicated to seeking optimization in a graph. In this case, robots exchange information with the environment by leaving virtual pheromones.

These tables enable physical objects on their surface to be manipulated. These objects can take various forms. Their positions are detected according to the technology used, which includes a camera [19, 21], magnetic following [55], weight detection [46] and radio frequencies [36]. Objects can be identified using different technologies, such as bar codes, images [18], specific shapes and RFID chips [30, 45].

2.3. Recognition systems for interactive tables

User actions can be explicit through the direct manipulation of an object or implicit when the manipulation is not performed for the purposes of interacting with the software system. The difficulty lies in understanding when the user wishes to act on the system. To determine this, interactive media must integrate recognition and processing systems.

[1] Smart Technologies. Smart Table. Website, http://smarttech.com/us/Solutions/Education+Solutions. 2009. (Reachable in 2014).

The recognition systems (e.g. TUIO [22] used by IntuiFace and ReacTIVision [21] used by ReacTable) identify and analyze movements on the tactile surface and/or tangible table. This analysis must consider user movements if possible, and these movements must be recognized by the support. In addition, a processing system can use received/successful data from the recognition system. The temporal evolution of the interaction (e.g., when a new user appears) is transmitted through a message between a server and a client (application). During the analysis of the interactions, a list is updated. If a list contains a change, the event is transmitted through a specific communication protocol (e.g., location description, event dating, additions/deletions).

The processing system uses the data perceived/received by the recognition system to translate the data into a more complete representation. The objective of the system is to interpret the movements of the users through a recognition engine. For example, the recognition of simple movements (such as clicking with one finger, dropping an element, double clicking with two fingers) and complex movements enables shapes (triangles, circles, squares), characters, numbers and punctuation to be drawn.

These different formulation methods enable applications to be developed and particular actions to be associated with a given formulation. If the formulation involves drawing a circle with a finger or turning an object in a particular direction, this can have several meanings to the application. In the case of an application defining a map, the formulation can enable the user to zoom in or out on an area of the map, while for musical applications, this action can allow the user to increase or decrease the frequency with which notes will be played. Consequently, each formulation can be interpreted differently depending on the applicative context [10, 40].

2.4. Taking the context into account to enrich the system's intelligence

The context describes how applications can react to changes in the physical environment or from the user. For example, Ward et al. [51] interpreted context with user location and environmental state. They also considered object location. Pascoe et al. [41] defined the context-awareness concept, in which context is defined according to four generic contextual capacities: perception, adaptation, resource discovery, and contextual augmentation. They defined context as a set of informational elements that make it possible to characterize the situation of an entity, i.e., a person, place or object that can intervene in the interaction between a user and an application.

Generally, the actions resulting from a recognition system are produced following a state change. For example, when a pressure sensor detects a presence, it switches on a light. This change does not integrate external factors that alter the perception of the object (the pressure sensor detects a presence, but not if there is already enough light in the room, in which case it does not switch on the light). The systems used by interactive tables do not integrate such a concept. The associated actions are related to an actuator ("I place the object - I perform an action").

To handle the problems related to the majority of interactive surfaces, several systems have integrated artificial intelligence concepts to endow tables with intelligent behavior. This behavior is ensured through the manipulated entities. In most cases, the reasoning process is exploited using software agents. These agents are endowed with behaviors, the ability to process and the ability to communicate *Intelligent Table* [2], *the Table-Robots* [6], or *Blip-Tronic 3000* table[3]. The Blip-Tronic is a musical instrument that plays sounds that vary with the type of object detected. The tone varies with the distance between the objects.

Although many solutions exist, they are generally related to an instantiated application (i.e., for certain interactive tables). The evolving entities do not integrate (or do not sufficiently integrate) behavior that can be interpreted to enrich the system's intelligence. Furthermore, the entities only interact with the medium. We rarely find cases where entities work together to exchange environmental data. In order to overcome such limitations, we propose a global interaction model in the next section. This model can be exploited simultaneously by various virtual and tangible entities evolving on the interactive table surface. This approach allows entities to interact by exchanging messages.

[2] Panasonic. Intelligent Table. Website, http://en.akihabaranews.com/7007/household/hdtv-the-interactive-and-intelligent-table-by-panasonic. 2006. (Reachable in 2014).

[3] Peter Benett, Sean Toru, and Lisa Tutte-Scali. Blip-Tronic 3000. Website, http://www.petecube.com/bliptronic3000/. (Reachable 2014).

3. Proposition: Global interaction model for interactive media by agent type

This section presents our proposal for managing virtual and/or tangible entities. These entities evolve on an interactive table surface using agent concepts. These concepts are used to design and implement applications for interactive tables. They take into account an environment's dynamic characteristics. In addition, the proposed model considers the heterogeneousness of the entities, the distribution of information and the complexity of the system. Then, we list the various agents involved as well as their interactions with the interactive medium's environment.

The proposed interaction model is based on a multiagent architecture. This architecture is illustrated by the concept of instantiated agents using an UML class diagram. This UML class diagram defines the agents that are involved in receiving sensory inputs from interactive tables. Agents are characterized by their autonomy in decision making, their ability to act and their knowledge. Their knowledge includes various pieces of information related to the behavior of an agent, the management of interactions with other agents and the environment in which an agent evolves. A behavior for the agents is represented by a set of roles with the aim of achieving a goal. These roles apply rules, aims and constraints. Therefore, we use the formalism of agents and roles detailled in [27, 31] to focus our proposition on different interactions for interactive media. We define the agent typology and the possible interactions between the agents and the users.

3.1. Agent typology

The type of agent is defined according to its interactions. We can define the agent type as a group member based on the Agent Group Role (AGR) model [13]. In this model, a group is a set of agents with common characteristics. The membership of an agent in a group means that it only plays the roles defined for its group [48]. The choice of agent type is determined mainly by the different interactions that occur during the simulation on interactive media. These interactions occur through tactile surfaces, either by using virtual objects or tangible objects [32].

In our context, we distinguish three types of agents: situated tangible agents, situated virtual agents and virtual agents (see descriptions below). These three types form specific groups. For example, the situated agent group includes situated tangible agents and situated virtual agents. The non situated agent group corresponds to all virtual agent types that ensure the consistency of the multiagent system (for example, a virtual agent with a role of manager or observer).

Situated tangible agents. A situated tangible agent is connected to a tangible object (i.e., an object that can be manipulated by one or more users). An agent's association to a tangible object is created to attribute roles to objects through agents. The situated tangible agents are agents situated in a plan corresponding to the interactive medium's detection surface. The situated tangible agents react to the environmental modifications generated by the movement of objects.

Situated virtual agents. A situated virtual agent is associated with virtual objects that are endowed with a graphical representation on the interactive medium. Unlike tangible objects, virtual objects are able to change shape, color, and size, and to disappear within the application. Every situated virtual agent can reason and have a set of roles that enable it to evolve in an environment. Each agent also has a view limited to a perimeter. This perimeter can vary according the agent's roles and the instantiated application. In the case of tactile media, a user can directly move virtual objects.

Virtual agents. Virtual agents handle the administration of the interactive medium and application progress. Virtual agents with a role of manager ensure optimal operational progress between various agents (tangible or virtual). Their role consists of sending information received from other interactive medium layers (hardware, middleware or software). Virtual agents also manage the dynamics of agent roles and check the consistency of these roles. In addition, they contribute to embedding new agents into the global environment.

3.2. Interaction between agents

Interaction between users and the surface (i.e., the table) happens through the manipulation of tangible objects and/or virtual objects if the support offers this possibility (*e.g.*, if the virtual surface is tactile).

Figure 1 describes an architecture model that focuses on interactions between agents. This figure demonstrates the principles behind the interactions between the various groups of agents as well as the interactions between users. The interactive surface can be exploited by one (U_1) or more users (U_1 to U_n). A set of multiple users can be represented by a group (for instance, group U_1, U_2 and U_3). The interactions between users (U1 to Un) are not discussed in this paper: they can be verbal and are currently only mediated through common representation on the interactive table and the associated tangible and virtual objects: each action performed on an object by a user is visible by the other users (as mentioned in [35]).

We distinguish several types of interaction in the system:

- Observational interactions between a non situated agent group and a situated agent group: such interactions enable, for example, a situated virtual or tangible agent to obtain a new role (e.g., the driver role in a road traffic simulator). A virtual agent with a role of manager must have access to a database to associate a role according to the requirements of a situated agent group. In the database, roles are selected using different criteria incorporated into a request. These criteria highlight situations of conflict between roles that can be opposite, similar, or inaccessible to the characteristics of the agent (e.g., the capacity of the agent to perform a set of roles).
- Exchange interactions: these correspond to the messages sent between different types of agents.
- Hardware interactions, which are used to receive the signals emitted by the physical environment and to emit notifications in the device (for example, LED activation and sound activation if the hardware permits).
- Middleware interactions: these correspond to the management of interactions between the hardware and the agents. In this situation, agents receive information about their position on the virtual and tangible surface. For example, situated virtual agent SVA_1 associated with virtual object VO_1 receives virtual coordinates (xv, yv) on its position from the middleware.
- User interactions, which result from object manipulation. The user can move tangible and virtual objects (virtual objects are movable, either indirectly through a tangible object or directly on the table if it has sensors (e.g., like for a touch screen).
- Mixed interactions, which result from the subordination of the position of an element with respect to the position of another one. A mixed interaction builds a relationship between the real or tangible world of the user and the virtual world in a coherent manner. In this case, we identify interactions between the agents (STA: Situated Tangible Agent or SVA: Situated Virtual agent) and between users (U) and agents (STA or SVA) as illustrated in Figure 2. Then, we show, for example, that a tangible agent can act on the position of virtual agents, but not vice-versa.

In this situation, when a user moves a tangible object, that user visualizes and modifies the position of a virtual object.

Therefore, we distinguish two modes of interaction [20]:

- **Direct interaction** is a means of communication that allows agents to exchange information directly with each other. This interaction happens through the exchange of messages between agents according to a predefined communication protocol. Communication protocols can facilitate and specify a type of interaction between agents. According to the protocol, one agent can send a message to another agent or to a group of agents.
- **Indirect interaction** comes from signals emitted by environment. In this approach, agents do not send messages; rather, they perceive the environment. For example, when a user moves a tangible object, signals are emitted from the environment. These interactions are generally characteristic of reactive agents responding to environmental changes.

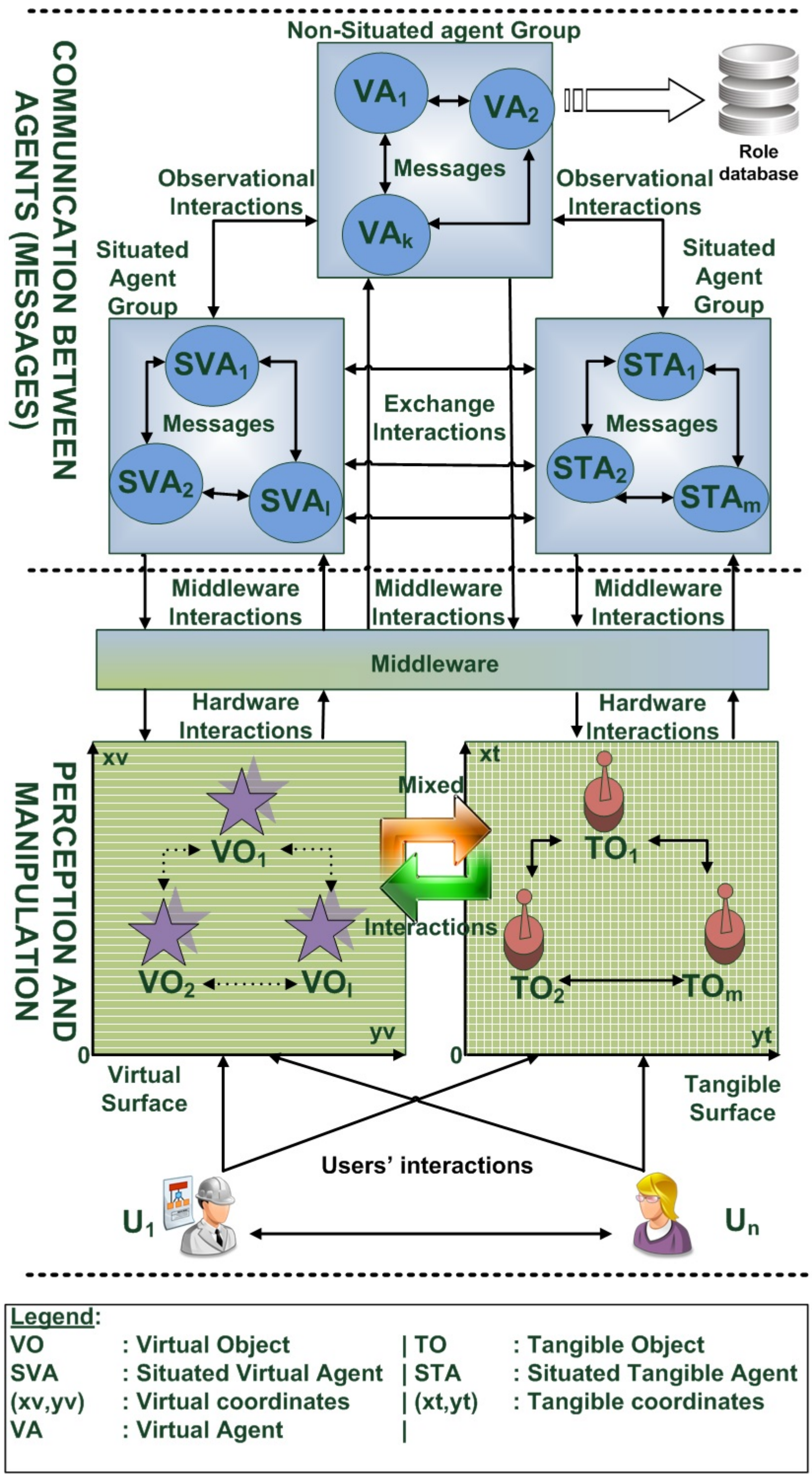


Figure 1: Different interactions on interactive surface

Objects/agents may arise out of these interactions. In other words, when assembled, they can perform

$$
\begin{array}{rcl}
SVA & \Longrightarrow & SVA \\
STA & \Longrightarrow & SVA \\
STA & \Longrightarrow & STA \\
U & \Longrightarrow & STA \\
U & \Longrightarrow & SVA
\end{array}
$$

Legend
$\Longrightarrow$: *act on the agent location*

Figure 2: Interaction between agents and users [32]

new functions they had not had independently of each other. These functions represent a way to complete the objectives of an agent in an open system [2].

3.3. Communication between agents

Interactions between agents rely on FIPA-ACL (Foundation for Intelligent Physical Agents-Agent Communication Language) [7] to exchange information and data. This language is considered an extension of KQML (Knowledge Query and Manipulation Language) [14] and is based on the speech act theory. FIPA-ACL enables agents to detail their communication with performatives (like 'inform', 'propose' and 'query'). A message communicated using FIPA-ACL is mainly characterized by the sender, the receiver(s) of the message, the content and the performative.

An agent can send a message to an agent group (like a group of situated agents), to an agent type (like a situated tangible agent type), or directly to a specific agent like the virtual agent (VA_1). Agents can register themselves within service directories (with a directory facilitator agent for example [4] under a 'virtual' or 'tangible' service, for example. An agent can query the directory facilitator agent to retrieve the list of all agents registered with a particular service. When an agent receives a message, it updates its knowledge and adapts its behavior.

The interaction mechanism used by agents is illustrated by an UML sequence diagram in Figure 3. This UML sequence diagram shows several detections of two tangible objects. When a user (U1) puts tangible objects on the interactive surface (messages 1.1 and 2.1), the middleware analyzes the hardware interactions (e.g., signals emitted by the hardware) to detect the position of tangible objects. If a tangible object is detected for the first time by the middleware and is not associated with a situated tangible agent, the middleware uses the agent mechanisms to communicate this information to the virtual agent $va1$ (messages 1.2 and 2.2). Messages received by any type of agent are put in the line. Next, messages are processed in the order in which they are received and are used to activate abilities or perform specific actions. $va1$ processes message 1.2 regarding the detection of a tangible object $to1$ and creates a situated tangible agent $sta1$ (message 1.3). Next, $va1$ informs the middleware regarding the identification of the agent $sta1$ (message 1.4). When the middleware receives performative 'INFORM-REF', a link between the tangible object ($to1$) and the situated tangible agent ($sta1$) is realized (message 2.5). Thus, this middleware is able to directly interact with the situated tangible agent when its unique related object is moved. The processing of the second message received by the agent $va1$ (message 2.2) similar to message 2.1 starts at the end of the message 1.4 (e.g., when agent $va1$ has completed specific actions relative to message 1.2). Next, when a tangible object is moved on the interactive surface (message 3.1 and 4.1), the middleware sends a message to agents $sta1$ (message 3.2) and $sta2$ (message 4.2) for each new position detected.

Based on modeling, in the following section we suggest applying the various agent typologies and their interactions through a road traffic simulation application. This application is used on an interactive *TangiSense* table. It enables tangible objects to be manipulated using RFID technology.

[4] The directory facilitator is the agent who provides the default yellow page service in the platform. This agent is automatically included and activated in the FIPA platform (i.e., in the JADE platform).

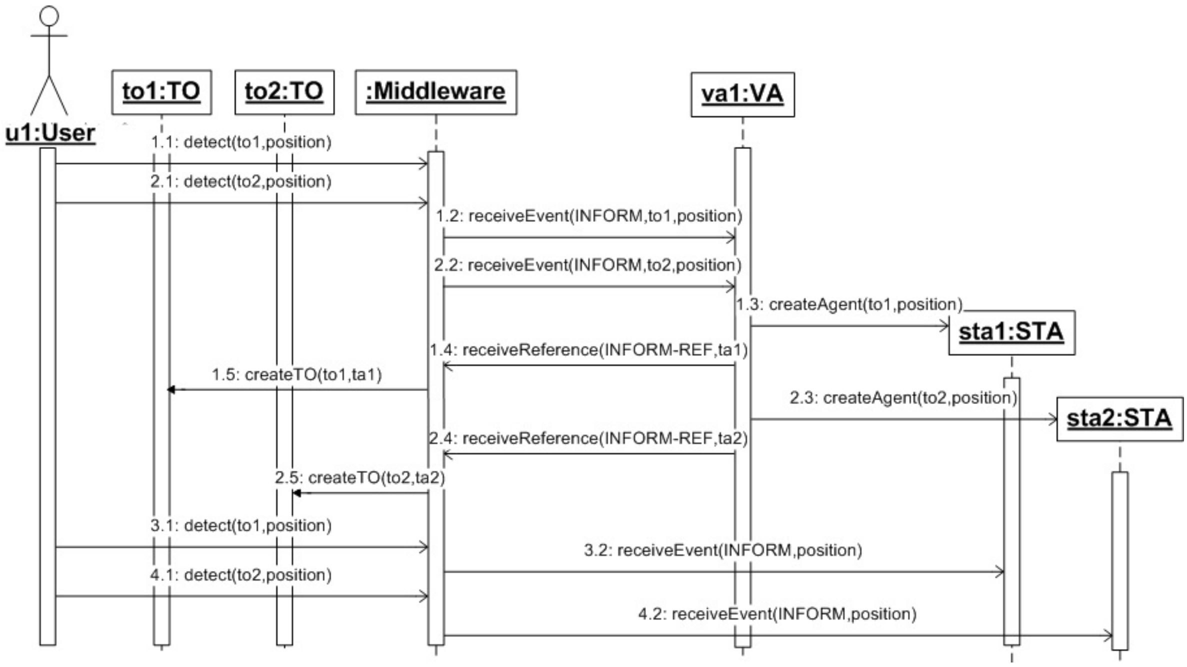


Figure 3: Exchange of messages between agents during detection of a tangible objects

4. Case study: agent-based simulation on an interactive table

In this section, we propose to implement interactions between the agents presented in the previous section on the interactive *TangiSense* table, which is equipped with RFID as capture technology. *TangiSense* was designed by the RFIdees company [5]. The table used in this scenario is the second version (produced in 2013), which has an HD screen.

Based on existing simulators and their related issues, such as use difficulty and a lack of collaboration between users (e.g., experts, decision-makers), as described in section 1, we demonstrate the benefits of a simulator dedicated to managing a transportation network.

The proposed road traffic simulator is a network comprised of links (e.g., roads and highways) and nodes (intersections). The simulator is intended for use by experts in safety, architecture and transportation, as well as by non experts like local elected officials, from whom approval must be obtained on road or infrastructure modifications.

Figure 4 illustrates a road traffic simulation managed by a multiagent system. The network is displayed on the *TangiSense* screen. The simulator aims to test hypotheses for reducing waiting time in an intersection, crisis management (e.g., in the event of an accident) and infrastructure modifications. Other objectives consider user actions on the application while validating the principle of associating situated virtual agents and situated tangible agents to obtain mixed interactions.

4.1. Tangible object validation

Tangible objects are used so that the simulator can interact with road signs (e.g., traffic lights, do not enter signs, one-way signs) and modify them. These tangible objects are manipulated by users and interact with other objects (e.g., objects from the situated agent group). These objects are equipped with RFID tags so that the network structure can be modified by changing intersection type or by reducing speed limit, for example.

Other interaction objects may be used to move the map (the idea is to put the area in the middle of the table by translating the map that the user wants to observe), to show street names and speed limits (these objects

[5] http://www.rfidees.fr/. (Reachable in 2014).

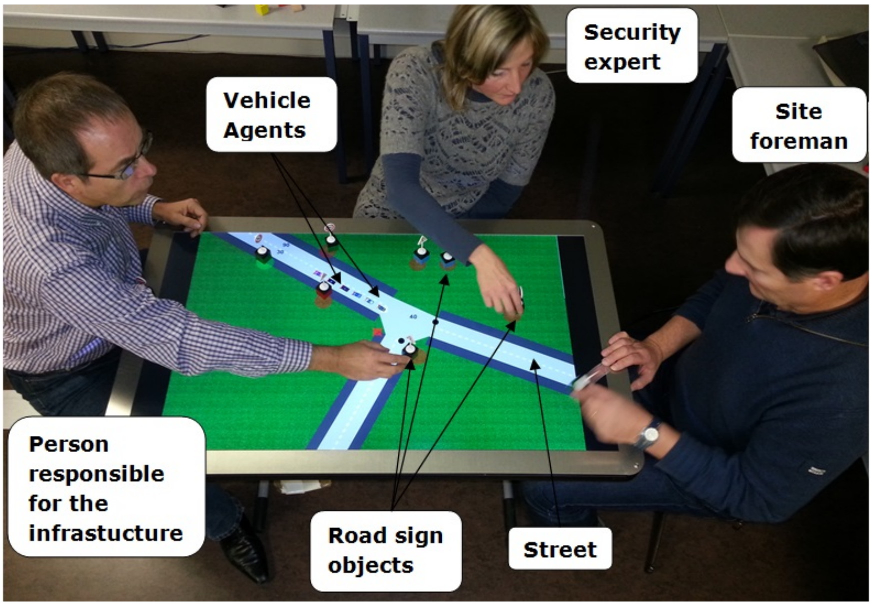


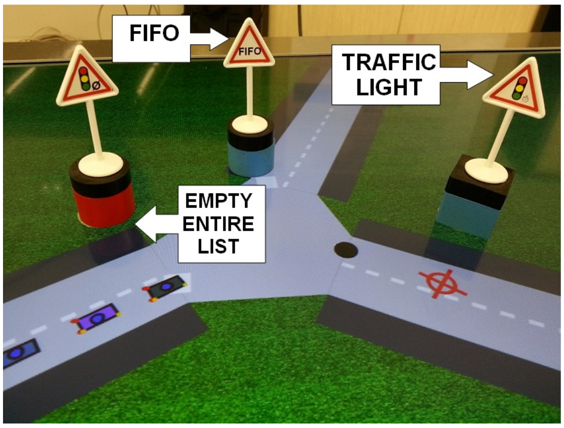
Figure 4: A road traffic simulation on the *TangiSense* interactive table (new version with integrated HD screen)

are used to obtain information on the entire road infrastructure) and to modify the map scale (the rotation direction is determined by the relationship between the old and the new radian value). Modifying the scale changes the precision of the map display. Users can employ this precision at their discretion to make it easier or more difficult to place tangible objects and change the specific road traffic area. Each of these objects is associated with situated tangible agents and is used to facilitate interaction between the users and the table (user interactions). The actions are more intuitive and can be multiple (depending on the number of people and the number of simultaneously used objects).

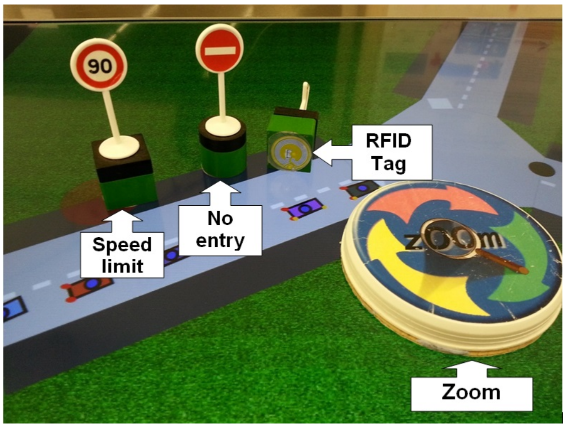
Every intersection is associated with a situated virtual agent, who manages all inputs with roles. To modify an intersection, a user must put a tangible object on the road. This object must be placed on the right side of the street and close to the intersection. If this tangible object is detected for the first time, the middleware sends the object position to a virtual agent (e.g., agent from the non-situated agent group). This virtual agent creates a situated tangible agent. Next, this situated tangible agent is associated to a tangible object by observational interactions. In this road traffic simulator, we distinguish three types of traffic lights (each corresponding to a tangible object, Figure 5(a)):

- *TRAFFIC LIGHT* tangible object: this corresponds to a block in one of three states (red, green and orange) that changes at regular time intervals, enabling vehicles to be successively moved by one of the inputs.
- *FIFO* tangible object: the intersection is a line of vehicles that move according to the first-in first-out principle. The inputs are symbolized by lights that are ordered such that the first vehicle from the list can exit. The intersection is based on a reservation algorithm proposed by Vassirani and Ossowski [50]. When driver agents approach the intersection, they send a message to the situated virtual agent responsible for the intersection to indicate the time of arrival.
- *EMPTY ENTIRE LIST* tangible object: this kind of intersection allows all the vehicles of an input to pass before changing the status of the input that contains the most vehicles. In this situation, the time interval between changes in the status of the light are irregular and the choice of inputs is not necessarily sequential.

These tangible objects are represented in figure 5(a) by traffic lights. However, in figure 5(b), they can also be characterized by a zoom, a speed limit, a do not enter (other types of road signs like a stop sign or a



(a) Traffic lights tangible objects



(b) Road signs and interaction tangible objects

Figure 5: Tangible objects equipped with RFID tag(s)

yield sign may be studied and added to the simulator). To be considered in the simulation, the speed limit and the not entry objects should be placed on a sidewalk for each input while the zoom can be placed anywhere. When any type of tangible object is detected (through its RFID tag) by the support, LEDs light up beneath it to indicate to users that the object is being taken into account by the system.

When an object is taken into account by the simulator, there is an option to remove it from the support. This feature is designed so that numerous identical objects are not necessary. It was also designed to be used in combination with user actions when the user changes the map (e.g., zooming in on the map, moving the map). When the map is changed while tangible objects are on the interactive table, objects will be taken into consideration by the simulator only when the user moves them.

The action plan is to reduce traffic jams at an intersection with tangible road sign objects. Figure 6(a) illustrates that roads and vehicles are initialized and shown when launching the application. Roads are generated from OpenStreetMap [6] cartography (a free map that enables the traffic of any city to be simulated). Vehicles are associated with virtual agents and generated from an XML file. This file contains their starting point. The XML file also determines if vehicles move randomly or toward a set of objectives to be achieved.

The agent in charge of the intersection validates the user action when entries are affected (Figure 6(b)) and executes the entry behavior by switching the type of traffic light. In the event of an intersection with traffic lights, vehicles move according to the status of the light. This information is perceived by agents through the environment in which they operate.

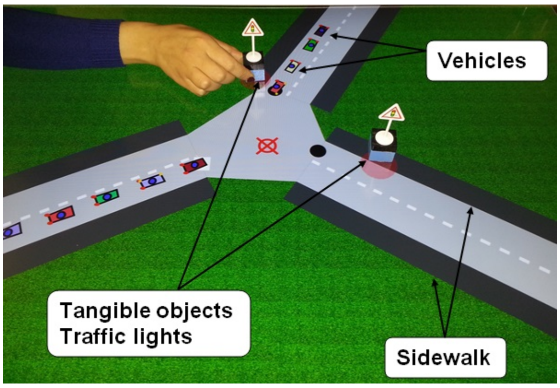
Figure 7 represents the immediate speed curve in km/h of a vehicle for a duration of four minutes.

During these four minutes, we measured and saved the immediate speed of a vehicle 8000 times, at regular time intervals, to establish this curve, which corresponds to vehicle movements.

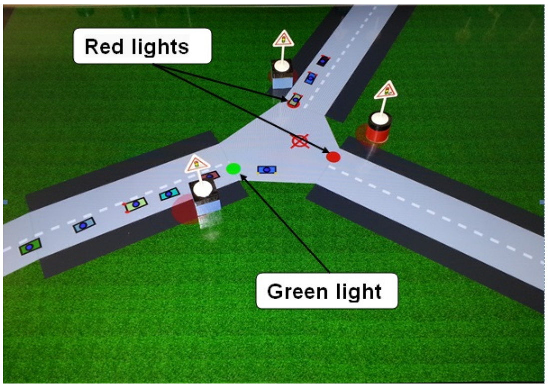
A vehicle starts at the initialization point on the map and accelerates gradually up to 30 km/h (launch area) before arriving at the road with a speed limit of 50 km/h. During the simulation, virtual vehicles/agents communicate with tangible objects/agents placed on the table by one of the users. This feature allows agent interactions to be tested (exchange interactions) and the new traffic light to be implemented (e.g., FIFO, EMPTY ENTIRE LIST). We can also evaluate mechanisms for sending a message to allow emergency vehicles to remotely change the state of a traffic light. This situation reduces the risk of accidents and clears the input at the intersection being approached by the emergency vehicle

In this simulation, the user puts a traffic light tangible object at each intersection. The traffic lights are set as follows: green lasts 30 seconds and orange lasts 5 seconds. Therefore an intersection opens every 35 seconds.

[6]http://www.openstreetmap.org/. (Reachable in 2014).



(a) Establishment of road sign objects at an intersection



(b) Management of road sign by an agent

Figure 6: Changing the virtual agent environment using tangible objects

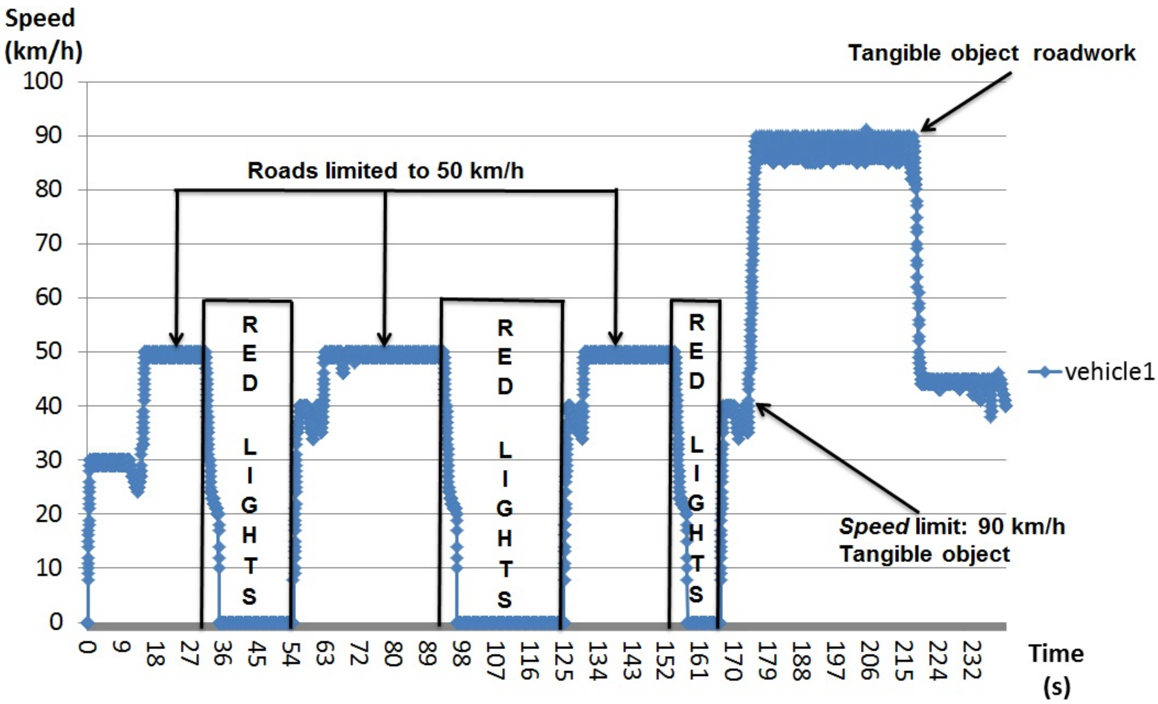


Figure 7: Instantaneous speed of a vehicle based on the road signs (case of a traffic light tangible object) [32].

When the vehicle arrives at an intersection, the associated agent analyzes the environment (e.g., entry into the intersection) and moves according to the status of the light. Figure 7 indicates that the vehicle has passed through three intersections with a red light. At the first intersection, the vehicle waited approximately 20 seconds, at the second intersection, 28 seconds and at the third, 8 seconds. Indeed, the waiting time varies according to a light's transition time.

During its travel and after passing through different intersection, the vehicle analyzes the environment to take into account the presence of a speed limit sign then a roadwork tangible object. The vehicle adjusts its speed according to the tangible object put by the user. Therefore, situated tangible agents will impact the simulation environment or its display. Situated virtual agents react by adapting their behavior based on these changes.

4.2. Virtual object validation

In the simulator, each vehicle is associated with a situated virtual "driver" agent as proposed in figure 1. This situated virtual agent has a set of behaviors and skills obtained by interactions with the non-situated agent group. Agents follow a set of rules that consist of following the direction of the road and keeping a safe distance from the vehicle ahead. Agents are also required to respect the road signs defined by the tangible braking objects on the table. The agent has environmental knowledge about the roads and road signs so that it can evolve in accordance with the various rules. The behavior of agents during the simulation may be normal (e.g., agent respects all the rules), dangerous (e.g., agent does not respect the speed limit) or aggressive (agent does not keep a safe distance). Depending on their behavior, these agents will variably modify the characteristics of the vehicle (e.g., position, acceleration, braking). Some drivers must carry out missions (e.g., go to work, go to the supermarket, go home). In this case, the behavior of the agent is to seek the shortest route for getting to each destination based on their understanding of the graph.

To illustrate our proposal, we selected an area of the city of Valenciennes (in North of France). This area is represented in the graph on which the vehicles move with normal behavior. It is composed of seven main intersections as well as one-way streets and two-way streets with speed limits of 50 to 70 km/h. We chose this area because it is difficult to move quickly on its streets. Unfortunately, there are no statistics about the numbers of vehicles or directions that these vehicles take at this location. Therefore, we varied experimental values for the following parameters: number of vehicles, departure of each vehicle, vehicle objectives and markings imposed by the intersections.

We decided to illustrate the school bus moving in Figure 8. This figure shows the possible starting points (S1, S2) for the buses and their different goals (*G1.1*, *G1.2*, *G2.1*, *G2.2*). In each simulation, we set a flow of 30 buses at the start of the simulation *S1* and set an initial time goal *G1.1* and then goal *G1.2*. Buses take the quickest routes (shown by dotted arrows) to get to each objective. When the final goal is achieved (goal *G1.2*), buses return to the first goal, and the process repeats (we reiterate that these data are defined in an XML file containing the description of each vehicle).

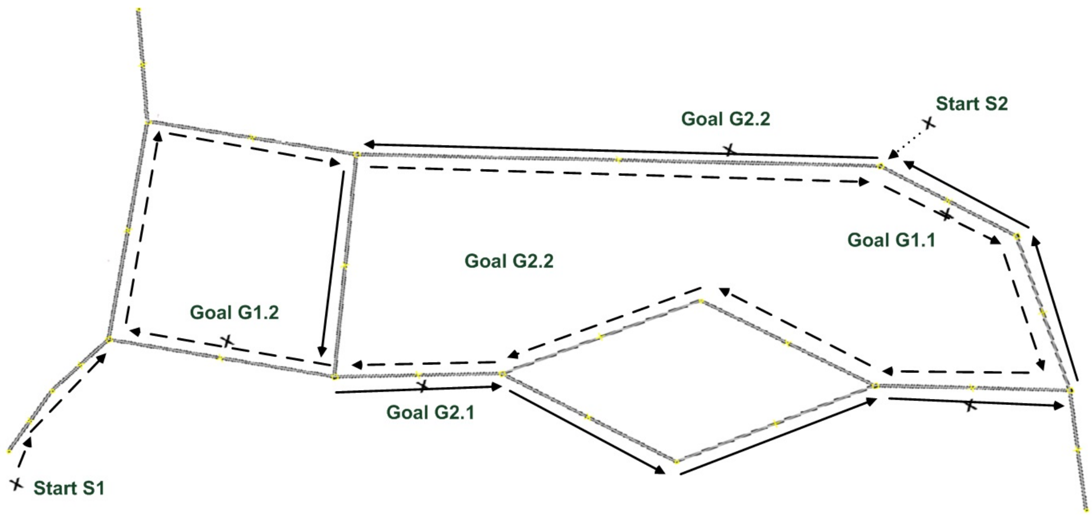


Figure 8: Ways of vehicles to achieve their goals

Next, we implemented a second stream of vehicles (named disruptive vehicles) that starts at *S2*. Disruptive vehicles can generate more or less heavy traffic depending on the time of day or event (e.g., end of the work day, traffic accident, soccer game). Therefore, such vehicles will disturb bus travel in an attempt to generate a realistic simulation. This flow can vary from 0 to 20 vehicles that either move randomly (without objectives) or with the following objectives: *G2.1* followed by *G2.2*. The routes for these vehicles are represented by the solid lines embedded with arrows.

Finally, we varied the signs at intersections by placing the three types of tangible objects (traffic lights, FIFO and EMPTY ENTIRE LIST). All intersections have the same tangible objects. For traffic lights, we set the duration of the *Green* state as 10 or 20 seconds with a 6-second margin of safety to let vehicles leave an intersection. This corresponds to the time between the *Orange* and *Red* states and the transition to the *Green* state of another light. For each simulation, the vehicles starting from *S2* were initially launched following buses from *S1*. This gave vehicles from *S2* time to disrupt the movement of buses from *S1*.

A flow of 30 buses starts from *S1*. We analyzed varying road signs and disruptive vehicles. We analyzed the time required for buses to achieve their goals in at least 10 buses routes. The simulation ended when the last vehicle had reproduced its goals 10 times. Achieving the same objectives 10 times provided a realistic average, especially for randomly moving disruptive vehicles.

The result of the simulation was the generation of a file containing information about the start and end times of the simulation as well as, the list of vehicles with their name and the times they entered the flow of traffic. For each vehicle, the transit times required between each of their objectives were recorded.

Figure 9 shows the results of an analysis of different simulations from different files. The data represent the calculations performed on different vehicle files to determine the average time taken by a bus to complete its route. This figure is a bar graph in which the x-axis represents the simulation parameters (e.g., 5d, 10d, 20d) for each of the signs at intersections (TIMING (10s), TIMING (20s), FIFO, and EMPTY ENTIRE LIST). The y-axis represents the average time (in seconds) needed for 30 buses to reach their goals. Columns depend on the parameters of the simulation and correspond to a number and type of disruptive vehicle (type is expressed either by the letter d for Dijkstra, which are vehicles with at least one goal (for instance: to go home), or the letter r for Random, which are randomly moving vehicles).

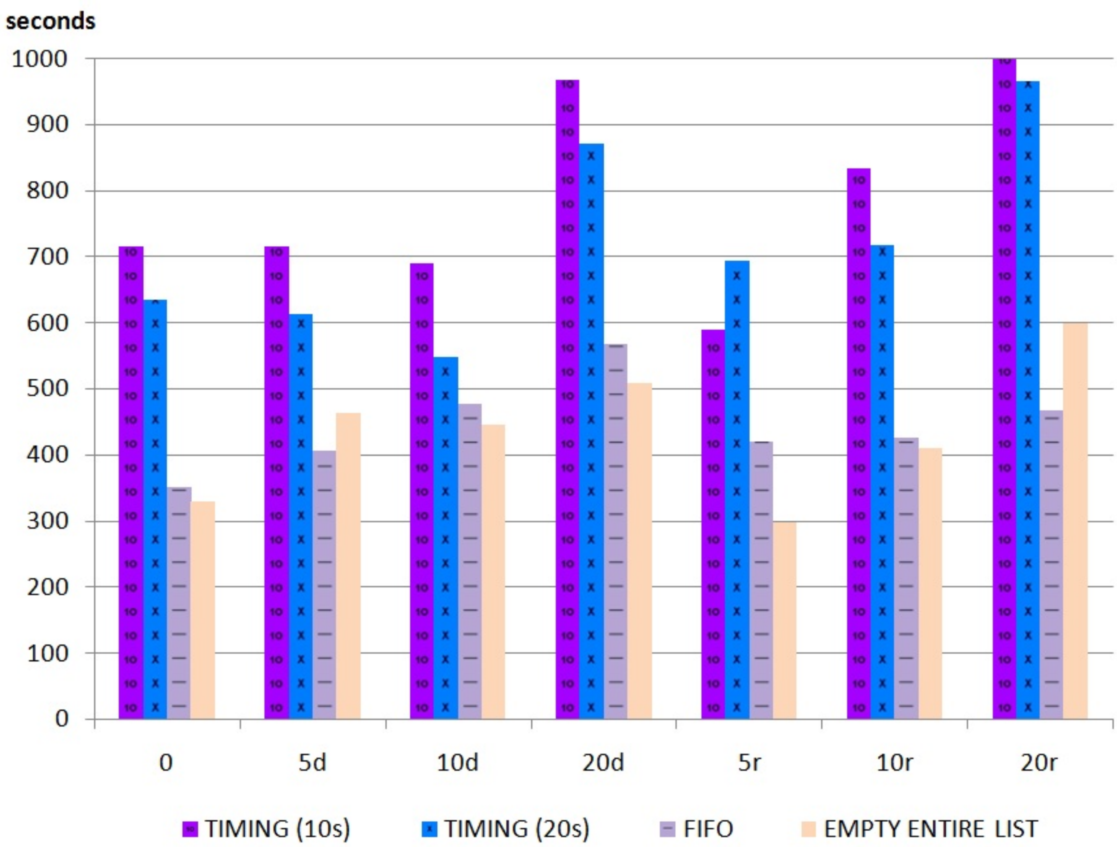


Figure 9: Results of various simulations on an area of the city of Valenciennes

Before analyzing the results, we specified that different simulations generate low dispersion data. The relative standard deviation is on average 1.27 % of all simulations, with a minimum of 0.37 % and a maximum of 3.77 %.

First, we observed that there were more disruptive vehicles when the average time is high (e.g., 966 seconds for 20r versus 693 seconds for 5r for TIMING (20s). The second road trend was a TIMING of 10s, which allows one to two vehicles to pass to reduce traffic compared with TIMING (20s), which allows three to four vehicles to pass. We inferred that lights with TIMING (10 and 20 seconds) lead to less traffic than FIFO lights, which in turn lead to less traffic than EMPTY ENTIRE LIST. This result does not hold true for all columns because the FIFO column indicates less traffic for 20r and 5d. The time difference used for a vehicle to achieve its goals in these columns (20r and 5d) with FIFO compared with ENTIRE LIST EMPTY is 131 seconds for the 20r parameter and 58 seconds for the 5d parameter. We can therefore conclude that the time saved by FIFO compared with the 5d EMPTY ENTIRE LIST parameter is too small to be able to draw conclusions. The randomly moving vehicles (20r) can determine whether one road sign is chosen relative to another. The general trend is to choose the EMPTY ENTIRE LIST road sign for a given area and set of objectives.

Accordingly, for this representative example, we validated the principles of agent communication with tangible and virtual entities on the surface of interactive tables. We showed that a combination of agent behaviors is needed to achieve a set of roles during the simulation and we discussed their interaction/communication during the simulation.

5. Conclusion

Interactive tables for manipulating tangible objects pave the way for new application design opportunities. In this article, we presented new concepts for managing interaction platforms (such as interactive tables that exploit agents). We proposed a solution for exploiting RFID technology that handles some existing problems using interactive tables. The complexity of these platforms is the result of multiple object detection and display technologies. It also affected by new forms of interaction, such as multi-user and tactile interaction.

We suggested modeling the interactions between various agents using the proposed typologies, which represent tangible and/or virtual entities. These entities evolve on an interactive table surface. The entities must be able to interact with each other as well as with users (and with different profiles). These agents can also be used to keep track of the performed actions so that the “life” of the object can be instantiated. Customizing these agents, whether virtual or physical, will allow them to use their intelligence to perform a set of actions and to adapt to different environments.

This kind of feature gives some insight as to what could be implemented in future applications in the Internet of Things context.

Then, we validated the approach using the *TangiSense* interactive table. Agents interact with both virtual and tangible objects. We highlighted all the elements and the concepts proposed throughout this article to enrich an object’s behavior by enabling it to communicate/interact with other objects or with human beings on interactive surfaces.

We intend to expand our work by using several interactive surface types (e.g., interactive tables, tablets, smartphones) simultaneously [29]. These surfaces can be remotely interconnected using the Internet [3] and data exchange.

Other possibilities involve expanding our work to include the field of serious games [4]. Serious games refers to a computer application that consistently combines both the utilitarian aspects (serious) and the fun aspects (game) of a video game. The aim is to move away from mere entertainment and design new applications on tangible interactive tables.

6. Acknowledgements

This research was partially financed by the French Ministry of Education, Research & Technology, the Nord/Pas-de-Calais Region, the National Center for Scientific Research (CNRS), the FEDER program, CISIT

(Plaiimob project) and the French National Research Agency (ANR TTT and IMAGIT projects, financial IMAGIT support: ANR-10-CORD-017). The authors would like to thank their partners on the TTT and IMAGIT projects: LIG, RFIdees, the CEA and Supertec. The authors thank warmly also the anonymous reviewers of the journal for their numerous constructive remarks.

References

[1] S. Abras, S. Ploix, S. Pesty, M. Jacomino, Advantages of MAS for the resolution of a power management problem in smart homes, in: 8th International Conference on Practical Applications of Agents and Multi-Agent Systems, PAAMS 2010, Springer Verlag, Salamanca, Spain, 2010.

[2] F.V. Aeken, Y. Demazeau, Minimal multi-agent systems, in: Proceedings of the 3rd International Conference on Multi Agent Systems, ICMAS '98, IEEE Computer Society, Washington, DC, USA, 1998, pp. 471–472.

[3] W. Ahmed, Y.W. Wu, A survey on reliability in distributed systems, Journal of Computer and System Sciences 79 (2013) 1243 – 1255.

[4] J. Alvarez, D. Djaouti, Serious games: an introduction, Questions théoriques, Valenciennes, France, 2012.

[5] J. Banks, Handbook of Simulation: Principles, Methodology, Advances, Applications and Practice, Engineering & Management Press, 1998.

[6] N. Beaufort, F. Charpillet, O. Rochel, O. Simonin, Interactive table to study interactions between swarm of robots and active environments, Technical Report, INRIA, 2011.

[7] F.L. Bellifemine, G. Caire, D. Greenwood, Developing Multi-Agent Systems with JADE, John Wiley & Sons, 2007.

[8] A.F. Blackwell, G. Fitzmaurice, L.E. Holmquist, H. Ishii, B. Ullmer, Tangible user interfaces in context and theory, in: CHI '07 Extended Abstracts on Human Factors in Computing Systems, ACM, New York, 2007, pp. 2817–2820.

[9] M. Buchmayr, W. Kurschl, J. Kung, A simulator for generating and visualizing sensor data for ambient intelligence environments, Procedia Computer Science 5 (2011) 90 – 97.

[10] O. Bucur, P. Beaune, O. Boissier, Définition et représentation du contexte pour des agents sensibles au contexte, in: Proceedings of the 2nd French-speaking conference on Mobility and ubiquity computing, UbiMob '05, ACM, NY, USA, 2005, pp. 13–16.

[11] T.H. Chen, H.H. Lin, Y.D. Yen, Mojo icuisine: the design and implementation of an interactive restaurant tabletop menu, in: Human-computer interaction: towards mobile and intelligent interaction environments - Volume Part III, HCII'11, Springer-Verlag, Berlin, Heidelberg, 2011, pp. 185–194.

[12] A.V. Dam, Post-wimp user interfaces, Commun. ACM 40 (1997) 63–67.

[13] J. Ferber, O. Gutknecht, F. Michel, Agent-oriented software engineering iv, in: P. Giorgini, J.P. Müller, J. Odell (Eds.), 4th International Workshop, Agent-Oriented Software Engineering (AOSE), volume 2935 of *Lecture Notes in Computer Science*, Springer Verlag, Melbourne, Australia, 2003, pp. 214–230.

[14] T. Finin, R. Fritzson, D. McKay, R. McEntire, Kqml as an agent communication language, in: Proceedings of the third international conference on Information and knowledge management, CIKM '94, ACM, NY, USA, 1994, pp. 456–463.

[15] A.K. Hartmann, A practical guide to computer simulation, World scientific Publishing, 2009.

[16] H. Ishii, B. Ullmer, Tangible bits: towards seamless interfaces between people, bits and atoms, in: Proceedings of the ACM SIGCHI, CHI '97, ACM, NY, USA, 1997, pp. 234–241.

[17] P. Jermann, G. Zufferey, B. Schneider, A. Lucci, S. Lépine, P. Dillenbourg, Physical space and division of labor around a tabletop tangible simulation, in: Proceedings of the 9th international conference on Computer supported collaborative learning, volume 1 of *CSCL'09*, International Society of the Learning Sciences, 2009, pp. 345–349.

[18] S. Jordà, The reactable: tangible and tabletop music performance, in: CHI EA, ACM, NY, USA, 2010, pp. 2989–2994.

[19] S. Jordà, G. Geiger, M. Alonso, M. Kaltenbrunner, The reactable: exploring the synergy between live music performance and tabletop tangible interfaces, in: TEI'07, ACM, NY, USA, 2007, pp. 139–146.

[20] H. Joumaa, Y. Demazeau, J.M. Vincent, Evaluation of multi-agent systems: The case of interaction, in: ICTTA'08, IEEE Computer Society, Damascus, Syria, 2008.

[21] M. Kaltenbrunner, R. Bencina, reactivision: a computer-vision framework for table-based tangible interaction, in: Proceedings of the 1st international conference on Tangible and embedded interaction, TEI '07, ACM, NY, USA, 2007, pp. 69–74.

[22] M. Kaltenbrunner, T. Bovermann, R. Bencina, E. Costanza, Tuio - a protocol for table based tangible user interfaces, in: Proc. 6th International Workshop on Gesture in Human-Computer Interaction and Simulation (GW 2005), Vannes, France.

[23] A. Kharrufa, D. Leat, P. Olivier, Digital mysteries: designing for learning at the tabletop, in: ACM International Conference on Interactive Tabletops and Surfaces, ITS '10, ACM, NY, USA, 2010, pp. 197–206.

[24] K. Kobayashi, T. Kakizaki, A. Narita, M. Hirano, I. Kase, Tangible user interface for supporting disaster education, in: ACM SIGGRAPH 2007, SIGGRAPH '07, ACM, NY, USA, 2007.

[25] A. Koyama, H. Suzuki, Real object-oriented communication method for ad hoc networks, Journal of Computer and System Sciences 79 (2013) 1101 – 1112.

[26] S. Kubicki, K. Borgiel, S. Lepreux, M. Wolff, C. Kolski, Réflexions autour des tables interactives : expérience utilisateur, utilisabilité, évaluation, in: Le travail humain, volume 75, pp. 229–252.

[27] S. Kubicki, Y. Lebrun, S. Lepreux, E. Adam, C. Kolski, R. Mandiau, Simulation in contexts involving an interactive table and tangible objects, Simulation Modelling Practice and Theory 31 (2013) 116–131.

[28] S. Kubicki, S. Lepreux, C. Kolski, Evaluation of an interactive table with tangible objects: Application with children in a classroom, in: 2nd Workshop on Child Computer Interaction UI Technologies and Educational Pedagogy, at CHI'2011, Vancouver, Canada.

[29] S. Kubicki, S. Lepreux, C. Kolski, Distributed ui on interactive tabletops: Issues and context model, in: M. Lozano, J. Gallud, R. Tesoriero, V. Penichet (Eds.), Distributed User Interfaces: Collaboration and Usability, Springer, 2013, pp. 27–38.

[30] S. Kubicki, S. Lepreux, Y. Lebrun, P. Dos Santos, C. Kolski, J. Caelen, New human-computer interactions using tangible objects: Application on a digital tabletop with rfid technology, in: J.A. Jacko (Ed.), HCI (3), volume 5612 of *Lecture Notes in Computer Science*, Springer, 2009, pp. 446–455.

[31] Y. Lebrun, E. Adam, S. Kubicki, R. Mandiau, A multi-agent system approach for interactive table using rfid, in: J.M.C..J.B.P. Y. Demazeau, F. Dignum (Ed.), 8th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 2010), volume 70 of *Advances in Intelligent and Soft-Computing, Advances in Practical Applications of Agents and Multiagent Systems*, Springer Berlin/Heidelberg, 2010, pp. 125–134.

[32] Y. Lebrun, E. Adam, R. Mandiau, C. Kolski, Interaction between tangible and virtual agents on interactive tables: Principles and case study, in: The 4th International Conference on Ambient Systems, Networks and Technologies (ANT 2013), volume 19, pp. 32–39.

[33] J. Lee, W. Moon, K. Yeom, D. Yoon, D.Y. Kim, J. Han, J.H. Park, Wave touch: educational game on interactive tabletop with water simulation, in: Proceedings of the 9th international conference on Entertainment computing, ICEC'10, Springer-Verlag, Berlin, Heidelberg, 2010, pp. 479–481.

[34] J. Leitner, M. Haller, K. Yun, W. Woo, M. Sugimoto, M. Inami, Incretable, a mixed reality tabletop game experience, in: Proceedings of the 2008 International Conference on Advances in Computer Entertainment Technology, ACE '08, ACM, NY, USA, 2008, pp. 9–16.

[35] A. Manches, C. OḾalley, S. Benford, Physical manipulation: evaluating the potential for tangible designs, in: Proceedings of the 3rd International Conference on Tangible and Embedded Interaction, TEI '09, ACM, NY, USA, 2009, pp. 77–84.

[36] E.S. Martinussen, T. Arnall, Designing with rfid, in: Proceedings of the 3rd International Conference on Tangible and Embedded Interaction, TEI '09, ACM, NY, USA, 2009, pp. 343–350.

[37] R. Mhemed, N. Aslam, W. Phillips, F. Comeau, An energy efficient fuzzy logic cluster formation protocol in wireless sensor networks, Procedia Computer Science 10 (2012) 255 – 262.

[38] C. Müller-Tomfelde (Ed.), Tabletops - Horizontal Interactive Displays, Springer, 2010.

[39] D. Mostaccio, R. Suppi, E. Luque, Simulation of ecologic systems using mpi, in: Proceedings of the 12th European PVM/MPI users' group conference on Recent Advances in Parallel Virtual Machine and Message Passing Interface, PVM/MPI'05, Springer-Verlag, Berlin, Heidelberg, 2005, pp. 449–456.

[40] A. Ortigosa, R.M. Carro, J.I. Quiroga, Predicting user personality by mining social interactions in facebook, Journal of Computer and System Sciences 80 (2014) 57–71.

[41] J. Pascoe, K. Thomson, H. Rodrigues, Context-awareness in the wild: An investigation into the existing uses of context in everyday life, in: R. Meersman, Z. Tari, P. Herrero (Eds.), On the Move to Meaningful Internet Systems 2007: OTM 2007 Workshops, volume 4805 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2007, pp. 193–202.

[42] J. Patten, B. Recht, H. Ishii, Audiopad: a tag-based interface for musical performance, in: Proceedings of the 2002 conference on New interfaces for musical expression, NIME '02, National University of Singapore, Singapore, Singapore, 2002, pp. 1–6.

[43] A.M. Piper, E. O'Brien, M.R. Morris, T. Winograd, Sides: A cooperative tabletop computer game for social skills development, in: Proceedings of the 2006 20th Anniversary Conference on Computer Supported Cooperative Work, CSCW '06, NY, USA, pp. 1–10.

[44] F.J. Ramis, F. Baesler, E. Berho, L. Neriz, J.A. Sepulveda, A simulator to improve waiting times at a medical imaging center, in: Proceedings of the 40th Conference on Winter Simulation, Winter Simulation Conference, Miami, Florida, 2008, pp. 1572–1577.

[45] J. Rekimoto, B. Ullmer, H. Oba, Datatiles : a modular platform for mixed physical and graphical interactions, in: ACM (Ed.), Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '01, New York, pp. 269–276.

[46] A. Schmidt, M. Strohbach, K.L. van, A. Friday, H.W. Gellersen, Context acquisition based on load sensing, in: Proceedings of the 4th international conference on Ubiquitous Computing, UbiComp '02, Springer-Verlag, London, UK, 2002, pp. 333–350.

[47] A.S. Shirazi, T. Döring, P. Parvahan, B. Ahrens, A. Schmidt, Poker surface: combining a multi-touch table and mobile phones in interactive card games, in: Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services, MobileHCI '09, ACM, NY, USA, 2009, pp. 73:1–73:2.

[48] T. Stratulat, J. Ferber, J. Tranier, Masq - towards an integral approach to agent-based interaction, in: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 2, AAMAS '09, International Foundation for Autonomous Agents and Multiagent Systems, Budapest, Hungary, 2009, pp. 813–820.

[49] K. Tumer, A. Agogino, Distributed agent-based air traffic flow management, in: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems, AAMAS '07, ACM, NY, USA, 2007, pp. 255:1–255:8.

[50] M. Vasirani, S. Ossowski, A market-inspired approach to reservation-based urban road traffic management, in: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 1, AAMAS '09, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2009, pp. 617–624.

[51] A. Ward, A. Jones, A. Hopper, A new location technique for the active office, in: IEEE Personal Communications, volume 4, pp. 42–47.

[52] R.H. Weber, R. Weber, Internet of Things: Legal Perspectives, 1st ed., Springer, 2010.

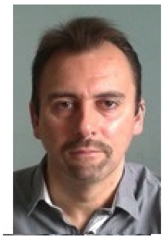
[53] J. Wei, X. Wang, R.L. Peiris, Y. Choi, X.R. Martinez, R. Tache, J.T.K.V. Koh, V. Halupka, A.D. Cheok, Codine: an interactive multi-sensory system for remote dining, in: Proceedings of the 13th international conference on Ubiquitous computing, UbiComp '11, ACM, New York, NY, USA, 2011, pp. 21–30.

[54] M. Weiser, The computer for the twenty-first century, in: SIGMOBILE Mob. Comput. Commun. Rev., volume 3, pp. 3–11.

[55] M. Weiss, F. Schwarz, S. Jakubowski, J. Borchers, Madgets: actuating widgets on interactive tabletops, in: Proc. 23nd annual ACM symposium on User interface software and technology, UIST '10, ACM, NY, USA, 2010, pp. 293–302.

[56] B.P. Zeigler, H. Praehofer, T.G. Kim, Theory of Modeling and Simulation. second edition, Academic Press, 2000.

Yoann Lebrun is a computer science and new interactions researcher at the Play Research Lab (CCI Grand Hainaut). His work focuses mainly on the design of innovative approaches in mixed (i.e., tangible and virtual) environments and on the promotion of intuitive use. After completing his studies in new information technologies and decision-making systems, he earned his PhD in computer science in 2012. His thesis was on the design of intelligent systems and their adaptation to interactive tabletops with tangible objects. After, completing his studies, Yoann became a technical platform manager at LAMIH, where he specializes in multi-user and distributed application design for tabletops.



Emmanuel Adam has been an Associate professor in Computer Sciences since 2003. His research focuses on holonic multiagent organizations. These flexible, open and hierarchical organizations in which each element is free to choose its actions to reach its individual goal while being constrained by the organization's need to reach its collective goal. The fields of application are transport simulation and manufacturing systems management. He is member of international conference program committees (including AAMAS and PAAMS) and a reviewer for international journals (such as EAAI, IJPR, JIMS and COMIND).

René Mandiau is currently a Professor in Computer Science at the Université de Valenciennes. He is the head of the DIM ("Decision, Interaction and Mobility") Computer Science Laboratory Team. Dr. Mandiau is a member of various program committees and has participated in more than 40 international conferences, including AAMAS, IAT and PAAMS. He is also a reviewer for international journals (such as JAAMAS, SIMPAT, EAAI and Applied AI). His research focuses on multi-agent systems, and mainly those systems applied to road traffic-simulation.

Christophe Kolski obtained his PhD in 1989. Dr. Kolski specializes in human-computer interaction, software engineering for interactive systems design and evaluation, and adaptive UI. He is a professor of Computer Science at the Université de Valenciennes and the head of the LAMIH "Human-Computer Interaction and Automated Reasoning" research group. He has authored and edited several books, and has authored or co-authored many book chapters, journal and conference papers and industry-related research reports.