International Conference on Artificial Reality and Telexistence
Eurographics Symposium on Virtual Environments (2015)
M. Imura, P. Figueroa, and B. Mohler (Editors)

Roles in Collaborative Virtual Environments for Training

Guillaume Claude†,[1] Valérie Gouranton‡[1] and Bruno Arnaldi§[1] [1]INSA de Rennes, IRISA/INRIA, Rennes, France

Figure 1: *Our Role Model is integrated in a Collaborative Virtual Environment for Training in neurosurgery procedure.*

Abstract

In this paper we present a novel approach to role modelling for Collaborative Virtual Environments with an Action Oriented Scenario Engine. Our role model is able to take into account different data about the actions such as the abilities, rights or resources required by an actor to determine whether he or she can execute an action. Furthermore, it considers that any data about an actor can have an influence on his or her role in the simulation. We provide also a team organisation model to define the impacts of the rules of the team on the role of the actors as suggested by the role theory. To illustrate our work, we used our model in a Collaborative Virtual Environment for the Training of a neurosurgery procedure.

1. Introduction

One of the main advantages of training using Virtual Reality is to provide multiple situations using one scene. These variations can come from changes in the scenario, and therefore in the events that occur in the environment such as a procedure. They can also come from changes in the organisation and definition of the actors: what they are able to do in the environment depending on the state how the simulation is unfolding. The common solution is to integrate the notion of role to define the position of the actors in the team. In this paper we focus on the modelling of the role of the actors to provide multiple training situations using one scene and one scenario. We illustrate our discussion using the case of a Collaborative Virtual Environment for the Training of a neurosurgery procedure (see Figure 1).

In this paper, we use some specific vocabulary: An *actor* is a virtual human or a user interacting in the virtual environment. The *definition of an actor* is a set of data that defines his state

- physically, for example his or her coordinates in space,
- as a team member, for example he or she is the leader of a team,
- in term of abilities, for example he or she is highly skilled in surgery.

Finally, we use the term 'Position' to define how an actor is placed in a team

Our model is generic and applicable to any domain. However, we use the surgical context as an illustration of our work (see Figure 1). In surgery, the core of the team is composed of the surgeon, the intern and the instrument nurse. The surgeon executes the main procedure of the surgery. The intern can be seen as the surgeon's third and fourth hands. The instrument nurse gives, takes back and prepares the instruments. The main goal of the instrument nurse is to give the right tool to the right receiver at the right time. It requires being able to anticipate the next actions of the two other ac-

† guillaume.claude@irisa.fr
‡ valerie.gouranton@irisa.fr
§ bruno.arnaldi@irisa.fr

DOI: 10.2312/egve.20151313

tors depending on the advancement of the surgery and on the state of environment, especially the patient. For example, if any bleeding occurs during the procedure, the team will have to stop the ongoing task, execute a cauterization procedure and go back to the previous task. Another element that can vary is the organisation of the team. In surgery, the steps of a procedure are strictly defined and organized. However, some changes can exist from team to team in the assignment of the tasks mainly caused by the habits of the surgeons. For example, some surgeons may prefer to be assisted differently for the cauterisation procedure depending on the state of the surgery and others may want to keep the exact same procedure throughout the surgery.

In our use case we provide a collaborative virtual environment for the training of the instrument nurse during a neurosurgery procedure. As stated, the main job of the instrument nurse is to anticipate the needs of the surgeon and the intern. His or her actions will be driven by:

- The steps of the procedure.
- The organisational habits of the team.
- The random events occurring during the procedure.
- Possible emergency depending on the procedure.

From the point of view of a Virtual Environment, all these elements increase the combinatorics of the possible sequencing of the actions of actors.

Our main concern is to provide accurate assignment of actions depending on the unfolding of the simulation. Existing Collaborative Virtual Environments use the notion of role to model the actions available to an actor depending on the state of the simulation and the unfolding of the scenario. If we take a closer look to the role as depicted by role theorists we can find multiple definitions of roles [Bid86]. However, Biddle and Thomas' definition [BT66] seems to fit to our needs in Virtual Reality: "A role is defined as concepts held by anyone about the behaviours of a person or a position." These concepts can be seen as several pieces of knowledge about an actor, such as his or her abilities, his or her position in the team or the tool he or she actually holds. Using this data and the context, it provides more informations about the behaviours of an actor such as:

- What he or she can do because he or she knows how to do it (abilities) or because he has the right tools (resources).
- What he or she is allowed to do (rights) because of his position in the team or because he or she has been allowed to by his or her hierarchy.

Role theory also states that the role of an actor can change with time owing to social or environmental pressure. An actor can obtain or lose duties, abilities or rights. These changes will affect his or her behaviour. For example, during a cauterisation procedure, the surgeon asks to his or her assistant to trigger the cauterisation by saying "Fire". The order of the surgeon makes the definition of his assistant change by giving him or her new rights.

We need our role model to match the following features:

- Expressiveness of the model defines versatility. Our model must be usable in as many cases as possible.
- Action oriented scenario engines have been proven efficient for procedural training. We want our model to be able to, at least, handle action information related to roles: conditions defining whether an actor can or cannot execute an action.
- Roles can evolve with time due to environmental or social causes.
  - Our model must be able to use the actions executed in the environment to model the environmental causes.
  - We do not plan here to model a full social model, but to express the rules defined in a team as the social causes.

2. Related Work

The concept of roles is widespread in the Collaborative Virtual Environment community [GMA07] [BQDLC03] [CGBA14]. Usually, the role of an actor provides information about what he or she is allowed to, able to or must execute. Some of the models [CLPC07] operate using data about the actor at a given time and data from the scenario.

In addition to Collaborative Virtual Environments, we have looked at Multi-agents Systems, especially in Organisation Centered Multi-agents Systems as they use the concepts of roles and social organisation [FGM04]. Furthermore, some works on Collaborative Virtual Environments are based on principles extracted from Multi-agents systems. For example, MASCARET [BQDLC03] is based on VOWEL [Dem95].

In this section, we take a look at the existing role models. We interest ourselves to the organisation of the data used to model the role. We then look at how these data are used to modify the behaviour of the actors and finally, how they can evolve with time to adapt the role to the simulation context.

2.1. Role Representation

The data used to define the roles and their organisation has a huge impact on the expressiveness of the model. In the literature, we have identified three different types of roles models : static set, combined static sets and dynamic set.

Static set models offer the less expressiveness. The data related to the role of an actor are organised in a set that cannot be modified. If the role of an actor changes, a complete new set of data must be defined. For each possible modification that may occur during the simulation, a specific set of data must be defined. Yet, there is very little chance that the new role of the actor is drastically different from the previous one. This solution is used in systems such as ABL [MS02], LORA++ [GMA07] or CASSIOPEIA [CD98].

Combined static sets offer to model the role of an actor using the union of several small static sets of data. Each small set contains a part of the data of the role. The role of an actor can be modified by adding or removing these sets of data. However, all the required subsets in the simulation must be defined in advance if the roles change during the simulation. Still, this modelling offers more expressiveness than static set modelling by its combinatorics feature. This model is in use in systems such as MASCARET [BQDLC03], IMS-LD [KO03] or AALAADIN [FG00].

Dynamic set models the role of an actor as a unique set of data specific to the actor. At any time, data of this set can be added, removed or modified. This modelling offers more expressiveness than the two previous ones as it allows specifically the data related to the change to be modified. Before the simulation, initial states of the set of each actor must be defined. This is the type of models used in [CLPC07] or in #SEVEN [CGBA14]. However the first one relies on multiple types of data and the second one uses only boolean values.

2.2. Role Model: Functioning and Relation with the Scenario

Concretely, a role model offers to define which scenario data must be provided to the actor. In that case we have identified two solutions : scenario oriented modelling and actor oriented modelling.

Scenario-oriented modelling directly uses the data extracted from the scenario as data for the role definition. If an action "INCISE" in our case is a part of the role definition, the actor can execute it. This solution is used in MASCARET [BQDLC03], IMS-LD [KO03], LORA++ [GMA07], ABL [MS02] or TEATRIX [PMP01]. The drawback of this method is that the role depends directly on the scenario. If the scenario is modified, the role of the actors must also be.

Actor-oriented modelling relies on the definition of the actors. For example an actor can have the attribute "SURGEON=true" or "SURGERY_SKILLS=EXPERT". These attributes are then used to define preconditions attached to the actions defined by the scenario. In [CLPC07], the actors must be defined before writing the conditions. As the conditions are integrated in the actions, even the interaction could not be defined before the actors. In #SEVEN [CGBA14] the scenario is adapted to the actors as a second part of the authoring using attributes on the actions. The simulation unfolding can be defined as without knowing the actors. The attributes can be added later.

Scenario oriented modelling seems easier to use because directly related to the scenario. However, *Actor-Oriented* modelling is closer to role theory as it focuses on the data related to the actors. Existing models are not getting closer to the role theory [BT66]. To our knowledge, there is no model providing more detailed data about an action such as rights or abilities constraints. Only LORA++ [GMA07] provides a related feature: an action in the scenario can hold attributes to inform that roles are 'forbidden', 'allowed' or 'allowed where appropriate'. This attribute feature is also used, in a lesser way, in #SEVEN [CGBA14] and MASCARET [BQDLC03]. They inform on the roles that can have access or not to the action but not more.

2.3. Role Evolution and Team Modelling

If we refer again to role theory [BT66], the role of an actor can change with time depending on the changes in the actor's both physical and social environments. The different existing models propose several ways to deal with role evolution. A part of these model, example TEATRIX [PMP01], do not handle role evolution. We have classified the others into three families: Scenario-driven Evolution, Action-driven Evolution and Team-Structure-driven evolution.

In systems using *Scenario Driven Evolution*, the roles are modified by the unfolding of the scenario. The scenario affects the roles of the actors as a consequence of the events that occur in the environment. This is the case in systems such as ABL [MS02]. In ABL, the role of the actors is completely defined by the state of the scenario.

Systems using *Action Driven Evolution* use the actions to impact the roles as with any other data from the environment. This is the case in [CLPC07]. These systems can be tightly related to Scenario Driven as some of the existing scenario models such as #SEVEN [CGBA14] can trigger actions in the environment that affects the role. But the actions can also be triggered by the actors: in that way they offer more possibilities than Scenario-driven ones. However, writing one scenario for both procedure description and team rules greatly increases the combinatorics of the possible unfoldings. Furthermore, one needs to write one scenario for each possible team organisation.

Team-Structure-driven evolution relies on a model that defines the relationships between the actors. The idea is to model groups such as teams and sub-teams. In these structures, actors are associated to several positions. In our case, we define a structure called "surgery team". In this team, an actor is related to the position of surgeon, another to intern and a third to Instrument Nurse. In CASSIOPEIA [CD98], these structures adapt the role of the actors depending on their availability and on the tasks to be done. In some systems, an actor can enter or leave a structure if he or she matches to some parameters. In MASCARET [BQDLC03] or in AALAADIN [FG00], an actor can ask to enter a structure. If the request is granted, the actor obtains a new set of actions (see combined static sets in 2.1). Team Structures provide evolution possibilities independently of the scenario. The simulation unfolding can change by altering the behaviour of the structure without altering the scenario. It is an important feature to model the rules of the team while the procedure remains the same. However, few existing solutions use the team structures as reactive systems to modify the roles during the simulation.

2.4. Synthesis

To our knowledge, there is no existing model that matches to all of our needs:

- Providing versatility.
- Attaching role condition information to actions.
- Impacting role related data using actions.
- Modelling the team rules and their effects on roles.

Action information related to roles is almost not defined. However existing solutions provide good properties: attributes held by the actions as in MASCARET [BQDLC03], LORA++ [GMA07] or #SEVEN [CGBA14] seems a good first step to extend the use of roles while combined with preconditions on the actions as proposed by [CLPC07]. It allows to use the scenario to define the procedure, leaving the organisation of the actors to the role model.

The evolution of the roles of the actors also needs to be driven by more than one element of the system: the actions [CLPC07] to model the environmental constraints and the team organisation as in MASCARET [BQDLC03], AALAADIN [FG00] or CASSIOPEIA [CD98] to model the social constraints. Actions also give the scenario or the actors the ability to modify the roles if needed. MASCARET [BQDLC03] uses team structures to modify the roles of the actors. However, these modifications impacts the actors only when they enter or leave a structure. This solution does not allow to model team rules or habits to handle specific changes in the environment or on the unfolding of the procedure.

3. Solution Overview

We propose to consider the role as proposed by Biddle [BT66]: Concepts held about the behaviour of a person or a position. To model these, our solution states that **the role is a very strong relationship between the scenario and the actions through an action filter** system. We consider that these 'concepts' in Collaborative Virtual Environments are the actions available to an actor depending on his definition applied to a set of conditions held by the same actions.

We consider that we use a Reactive 3D Environment. This environment is orchestrated using a Scenario Engine. The scenario engine provides actions to the actors depending on the unfolding of the simulation. This actions hold attributes defining conditions on their availability for the actors depending on their definition. Then, the action filter interprets the attributes to provide a specific set of actions for a given actor using his or her definition.

Team structures are groups of actors able to change the definition of their members as a reaction to events occurring in the simulation. These events can be:

- An actor entering or leaving the team.
- A specific step of the scenario.
- A change perceived in the 3D environment.

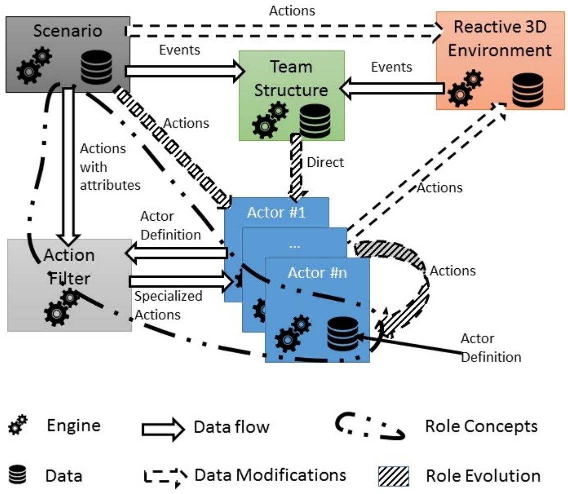

Figure 2: *General work-flow of our solution: the action filter uses the definition of an actor to specialize the actions provided by the scenario engine. The definitions of the actors can be modified by actions or by the team structure. The team structure supervises the events originating from the scenario or from the environment. It reacts by modifying the definitions of the actors.*

Figure 2 shows the general workflow of our solution.

Accordingly to the role theory [Bid86], in our solution the definition of an actor, and by extension his or her role, can change due to

- the actions executed in the environment (environmental causes)
- rules of the team (social and organisational causes, example: the surgical team)

We work in collaboration with medical staff (a neurosurgeon and an instrument nurse trainer). Each part of our solution is intensively discussed with these partners.

4. Data organisation, Model Functioning and Relation with the Scenario

Our role model relies on the definitions of the actors to filter the actions provided by the scenario system. In this section and the following, we detail all the components of our model: the definition of the actors, the attributes provided by the scenario on top of the actions, the action filter and the evolution of the definitions of the actors.

4.1. Definition of the Actors

In any Collaborative Virtual Environments, actors are defined by a set of data. In addition to usually used data (3D models, coordinates in spaces etc), our solution considers that these data can relate to at least:

- Physical description such as the tools he or she holds.
- Ability descriptions such as his or her experience in a specific domain.
- Rights descriptions such as his or her authorization to handle specific tasks.

Other data related to the actor can be also used such as his or her position in the team. The abilities and the rights are integrated in his or her knowledge database. They can be modified by components of the virtual environment such as an action or a team structure (see section 4.3).

In our simulation one of our actors can be a senior surgeon who has a high level of expertise in surgery, knows how to use motorized instruments, and is the leader of the surgery team. We can also use a less experienced surgeon or even a junior surgeon by decreasing his level of expertise in his initial definition. It has a direct impact on the simulation by modifying actions accessible to this actor (see Section 4.2) and his or her social relation with the other actors (see Section 4.3).

4.2. Informed Scenario and Action Filter

Our first concern was to provide a role model closer to the role as seen in reality. Role theorists [Bid86] see roles as a set of concepts defining the expected behaviours of an actor. We propose here to see those behaviours as the actions an actor can execute because he or she fulfils conditions in term of abilities, rights or resources. We also consider that, depending on the actor, some actions may have more importance than others.To model this, our solution relies on Action oriented Scenarios we call Informed Action Oriented Scenarios. *Informed Action Oriented Scenarios* provide more data to the actors than the available actions. We propose to model the additional data as attributes held by the actions. These data provides information such as conditions for an actor to be able to execute an action. We propose here four types of attributes: *Abilities*, *Rights*, *Resources* and *Weightings*. However, it is possible to define other attributes to fit to the actors model, to specificities of the domain or to the simulation itself.

Abilities refer to technical or physical abilities defined in the actor's definition. In our solution, an actor that do not fulfil the condition can not execute the action.

Rights refer to social positions or conditions defined in the actor definition. If an actor is technically able to execute an action (because he or she fulfil the abilities conditions), he or she can execute it even if he or she is not allowed to. This is equivalent to him or her going beyond his or her rights.

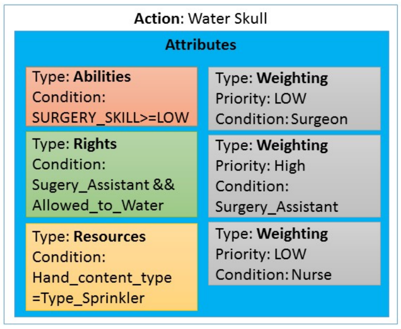

Figure 3: *An example of action provided by an Informed Scenario Engine. Attributes provide condition related to the definition of the actors that allows to decide if an actor is able and/or allowed to execute an action, what he or she needs to execute this action and what makes this action more or less important than another.*

Resources define an object, tool or element that must be in the possession of the actor for him or her to execute the specified action. As with abilities, resources are technical constraints. However, the resources can be obtained by an action. Resources allows an actor to knows what are the needed tool for an action to be executed. This feature is very important in our case as it is the main duty of the Instrument Nurse to provide the right resources to the right actor at the right moment.

Weightings provide data that can help an actor to decide which action is a priority over the others. Weightings provides a type such as "emergency", "difficulty" or "duration", a weighting value, and a condition on actors definition used to defined which actors must consider this weighting at this value.

Attributes contains conditions upon the content of the definition of the actors. The action filter interpret these data to adapt them to a specific actor using a given actor definition (see Figure 2). The idea is to interpret the conditions to provide data defining that an actor fulfil or not the conditions to execute an action or to know how he or she is concerned by a specific weighting or if he or she possesses the right resources. Figure 3 shows an example of an action 'WATER SKULL' extracted of our use case.

4.3. Actors Definition Evolution and Team Structures

According to role theory, a change in the role of an actor can be the consequence of events in his physical or social environment. We propose to handle the evolution of role by modifying the definition of the actor.(see section 4.1). These modifications rely on two types of elements: actions and social organisation.

The actions have a straightforward impact on the actors

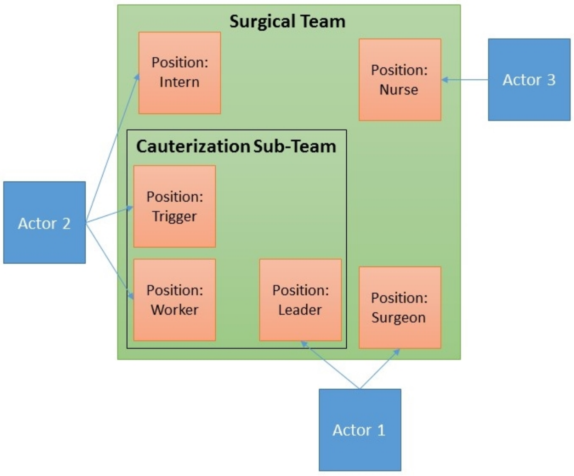

Figure 4: *In our use case each actor is assigned to a position in the main team. During the cauterisation procedure, a sub-team is created and some of the actors are assigned to a second position in it.*

that participate in their execution. As a simple example, the action "GIVE" that allows an actor to give an object to an other actor modifies the data, in the definition of the actors, related to the content of their resource "hand".

The social organisation of the actors defines rules that modifies them as reactions to events occurring in the simulation. We propose to limit the social organisation to the team structure and rules.

Our team structure model is able to handle role modification in several cases:

- Modifications in the team structure of the actors (an actor can enter or leave a social structure).
- Canges in the state of the simulation environment.
- Unfolding of the scenario.

Team structures are groups of actors supervising the events occurring in the environment (see Figure 2). They modify the definition of their members as a reaction of these events.

Positions relate actors to a structure. A structure can have multiple positions and an actor can be assiagned to several positions in several structures. An actor may need to fulfil some conditions in his or her definition to be assigned to a position. In the same way, tests can be executed during the simulation to know if the actor still matches to these conditions or if he or she can leave the position. When entering or leaving in a position, the actor can have his or her definition modified.

Team Structure are organized **Hierarchically**. A structure can create sub-structure to react to specific events. The actors assigned to the positions of the sub-structure are selected among the actors of the parent structure. The parent structure can also change the position assignments of the sub-structure or dismantle it.

When a structure is no longer needed it can be **dismantled**. All the actors are removed from their positions in it. It implies that the actors are modified as usual when they leave the position.

A structure can perform a **consistency check**. A consistency check verify that some conditions in the state of the structure are fulfilled.

In our case, our three actors are assigned to a position in the main team structure when the simulation start. At this point, each position in this structure checks that the assigned actor fulfils some conditions. For example, the surgeon must be experienced in surgery. If one of them do not fulfils the conditions, he or she is not assigned to the position. If it appends, the consistency check of the main team fails because one of the position is not assigned and the simulation stops. If they managed to be assigned to the positions, their definition is modified. For example the intern is provided by the rights of the surgery assistant.

During the simulation, if the patient starts bleeding, the main team creates a new "CAUTERISATION" sub-team with its own specific rules and assign the different positions in it to the members of the team. Depending on the behaviour of the main team, the sub-team may have different position assignments. When the leader of the cauterisation sub-team announce "FIRE" the actor assigned to the "TRIGGER" position is granted the right to push the trigger pedal by the sub-team. When the cauterisation is over, the main team dismantle the sub-team. The actors then leave their positions in it and their definition is modified in consequence. Figure 4 shows a snapshot of a possible team structure in our use case.

5. Use Case

As an illustration of our work, we propose a virtual reality application providing a Collaborative Virtual Environment for the training of instrument nurses (see Figure 1). The main focus of this system is the anticipation of the needs of the two other members of the surgical team: the surgeon and the intern. The nurse must provide the right tool at the right time to the right team member and organise the table to leave free spaces for them to release the tools they no longer needs. This use case is only an illustration of our work and our solution can be applied to any domain.

Our Virtual Environment relies on the framework [BGB*] to model the interaction with the environment. We have developed an extended version of the scenario engine #SEVEN [CGBA14] [CGA15] integrating the actions attributes. Our team and role models has been implemented in a new library, independently of the two first technologies. We then integrated all of them using the Unity3D framework. The 'surgeon' and the 'intern' are autonomous actors while the 'instrument nurse' is the user of our system. The user interacts with the environment using a motion tracked controller. A video demonstration of our application can be found at http://youtu.be/HNH1D8hN71o.

5.1. Scenario of the surgery procedure

In our simulation the three different actors are assigned to three positions in the surgical team: the surgeon, the intern and the instrument nurse. The scenario describes a part of a neurosurgery procedure: the craniotomy phase. The procedure unfolds as follows:

1. Incision of the skin of the patient
2. Drilling of the skull
3. Taking off the dura matter
4. Sawing of the skull
5. Removing the bone flap
6. Storing the bone flap

These main steps are performed by the surgeon. During the steps 2 and 4, the intern must water the skull to prevent it from overheating.

At any time after the incision, a bleeding can occur. In that case, the team must cauterize it in priority before going further in the main procedure. Three positions in a sub-team are related to this sub-procedure: 'leader', 'worker' and 'trigger'. The cauterization sub-procedure unfolds as follows:

1. 'Worker' put Bi-Polar pliers on the patient
2. 'Trigger' pushes on a triggering pedal
3. the bleeding is stoped
4. 'Trigger' releases the triggering pedal
5. 'Worker' removes Bi-Polar pliers

During this sub-procedure it is up to the leader to announce "FIRE" and "STOP" to give the right to the Trigger to push or release the pedal.

5.2. Informed Scenario

The actions provided by the scenario hold attributes as proposed in section 4.2.

Abilities and **resources** attributes define the actions that are available to the actors. The abilities of an actor do not change during our simulation. However, the state of the resources of an actor changes depending on a set of generic actions allowing to pick up, give or release an object. Each surgical action hold attributes defining both abilities constraints (mainly surgery skills) and resources constraints (the required instrument). It allows the actors to know what are the actions they cannot execute because they do not know how to and what actions they cannot execute because they need a specific instrument.

Rights attributes constrain the available actions depending on the position in the team of the actor. Rights are managed by the team structure. It provides or removes rights depending on the unfolding of the simulations and on the position of the actors. The actions constrained by rights are the fallowing:

- "PICK-UP" and "GIVE" instruments: provided to the 'instrument nurse'
- "PUSH" and "RELEASE" pedal: provided to the 'trigger' when the 'leader' of the sub-team announces the right keyword (see Figure 5)
- "ANNOUNCE A KEYWORD": provided to the 'leader' of the sub-team.

Weightings attributes are used to define that the cauterisation sub-procedure must be executed before continuing the main procedure.

5.3. Actors and Team Organisation

Our application uses configuration files to be able to modify the simulation without any developments. These files defines the initial definition of each of the actors and the team configuration. The initial definitions of the actors describe their expertise level in specific domains such as their surgery skills or the initial state of their resources (their hands are empty at the beginning of the simulation). The team structure configuration defines the type of the team and sub-teams and assign the actors to the positions of the team: Surgeon, Intern and Nurse. The types of the teams defined the rules used in the simulation. Here is a partial list of the rules we have defined:

- Surgeon position is accessible only to an actor expert in surgery.
- Actors can have only one position in the team.
- When a bleeding occurs, a cauterisation sub-team is created.
- The surgeon must have done the drill action a number of time depending on his or her expertise in surgery. The more he or she is experienced, the less he or she has to perform the action.

Here are some examples of possible sub-team organisations:

- 'Surgeon' is assigned to the 'leader' position, 'intern' is assigned to 'worker' and 'trigger'.
- If 'surgeon' holds an electric instrument (a saw or a drill) 'intern' is assigned to 'worker', otherwise the surgeon is. 'Nurse' is assigned to 'Trigger'.

6. Conclusion and Future Work

We have presented a new model for roles in collaborative virtual environments. We propose to use informed action oriented scenarios (scenarios providing actions holding attributes) in combination with an action filter. The action filter

Figure 5: *During the cauterisation, the actor (here the user) can push or release the pedal to trigger the Bi-polar pliers. However, he or she must not execute it before the leader has given the order. This difference is the result of the use of the attributes describing rights held by the actions and the reactive behaviour of the team structure giving the right to the user when the surgeon announces the right keyword.*

uses the definitions of the actors and the actions' attributes to provide sets of actions specific to each actor. The attributes of the actions provide more specific data than any other existing model such as rights, abilities, resources and weightings. The model is also reactive. It makes the roles evolve depending on the events occurring in the environment or the unfolding of the scenario based on the rules defined by the team organisation of the actors. The simulation can be modified by providing other team organisations and other initial actors definitions.

Using this model we have developed a Collaborative Virtual Environment for Training instrument nurses in a neurosurgery procedure. Changing the unfolding of the simulation by modifying the initial definition of the actors is very easy once the 3D reactive environment and the scenario are defined. The same goes for their possible evolutions related to the rules of the team.

Currently, the authoring of an informed scenario is a long and complex task. We plan to create a scenario-authoring tool able to propose action attributes depending on the needs of the simulation. Role theory states that role is defined by the inner definition an actor has of another. It could be very interesting to study the limits of our model in simulations where the actors have partial or false knowledge about the others and the impacts of such simulations on training.

Acknowledgements

This publication is supported by the S3PM project of the CominLabs Excellence Center and the FUI PREVIZ. The prototype has been developed with the help of Florian Nouviale and Cédric Le Cam. Finally, thanks to Julian Joseph for his precious help.

References

[BGB*] Bouville R., Gouranton V., Boggini T., Nouviale F., Arnaldi B.: # five: High-level components for developing collaborative and interactive virtual environments. In *SEARIS*. 6

[Bid86] Biddle B. J.: Recent development in role theory. *Annual review of sociology* (1986). 2, 4, 5

[BQDLC03] Buche C., Querrec R., De Loor P., Chevaillier P.: Mascaret: pedagogical multi-agents systems for virtual environment for training. In *Cyberworlds* (2003). 2, 3, 4

[BT66] Biddle B., Thomas E.: Role theory: concepts and research. 2, 3, 4

[CD98] Collinot A., Drogoul A.: Using the cassiopeia method to design a robot soccer team. *Applied Artificial Intelligence 12*, 2-3 (1998). 2, 3, 4

[CGA15] Claude G., Gouranton V., Arnaldi B.: Versatile Scenario Guidance for Collaborative Virtual Environments. In *GRAPP* (2015). 6

[CGBA14] Claude G., Gouranton V., Berthelot R. B., Arnaldi B.: Short paper:# seven, a sensor effector based scenarios model for driving collaborative virtual environment. In *ICAT-EGVE* (2014). 2, 3, 4, 6

[CLPC07] Cavazza M., Lugrin J.-L., Pizzi D., Charles F.: Madame bovary on the holodeck: immersive interactive storytelling. In *ACMMM* (2007). 2, 3, 4

[Dem95] Demazeau Y.: From interactions to collective behaviour in agent-based systems. In *EuroCog* (1995). 2

[FG00] Ferber J., Gutknecht O.: Operational semantics of multi-agent organizations. In *Intelligent Agents VI*. 2000. 3, 4

[FGM04] Ferber J., Gutknecht O., Michel F.: From agents to organizations: An organizational view of multi-agent systems. In *Agent-Oriented Software Engineering IV*. 2004. 2

[GMA07] Gerbaud S., Mollet N., Arnaldi B.: Virtual environments for training: from individual learning to collaboration with humanoids. In *Technologies for E-Learning and Digital Entertainment*. 2007. 2, 3, 4

[KO03] Koper R., Olivier B.: Representing the learning design of units of learning. 3

[MS02] Mateas M., Stern A.: A behavior language for story-based believable agents. *IEEE Intelligent Systems 17*, 4 (2002). 2, 3

[PMP01] Paiva A., Machado I., Prada R.: Heroes, villians, magicians : dramatis personae in a virtual story creation environment. In *ACM IUI* (2001). 3