MISER: Mise-En-Scène Region Support for Staging Narrative Actions in Interactive Storytelling

Jamie Matthews
Teesside University, UK
jamie.matthews@tees.ac.uk

Fred Charles
Bournemouth University, UK
fcharles@bournemouth.ac.uk

Julie Porteous
Teesside University, UK
j.porteous@tees.ac.uk

Alexandra Mendes
Teesside University, UK
a.mendes@tees.ac.uk

ABSTRACT

The recent increase in interest in Interactive Storytelling systems, spurred on by the emergence of affordable virtual reality technology, has brought with it a need to address the way in which narrative content is visualized through the complex staging of multiple narrative agents' behaviors within virtual story worlds. In this work we address the challenge of automating several aspects of staging the activities of a population of narrative agents and their interactions, where agents can have differing levels of narrative relevance within the situated narrative actions. Our solution defines an approach that integrates the use of multiple dynamic regions within a virtual story world, specified via a semantic representation that is able to support the staging of narrative actions through the behaviors of the primary and background agents' that are involved. This encompasses both the mechanics of dealing with the narrative discourse level as well as the interaction with the narrative generation layer to account for any dynamic modifications of the virtual story world. We refer to this approach as MISE-en-scène Region (MISER) support. In this paper, we describe our approach and its integration as part of a fully implemented Interactive Storytelling system. We illustrate the work through detailed examples of short narrative instantiations. We present the results of our evaluation which clearly demonstrate the potential of the MISER approach, as well as its scalability.

Keywords

Virtual Agents; Interactive Storytelling; Narrative Staging; Virtual Reality; Crowd Simulation

1. INTRODUCTION

Interactive Storytelling (IS) systems are multimedia systems in which users can interact and influence, in real-time, the evolution of a narrative as it is presented to them. This can be via 2D or 3D animation [17], filmic content [19] or text [10], however we restrict discussion here to real-time interactive 3D animation-based storytelling systems. Recently there has been an increase in interest in IS systems as they move from research prototypes [13, 16, 17] to commercial products. This has in part been spurred on by the emergence of affordable virtual reality (VR) technology which has re-initiated the need to address the way the behaviors of narrative agents are staged within 3D worlds, especially given VR's potential to include large number of virtual agents.

The prevailing approach to the staging of narrative content in 3D animation-based IS, via the visualization of narrative actions, has tended to be somewhat based on *ad-hoc* scripting (i.e. specifying a set list of instructions for agents to follow). Here we consider the visualization of narrative actions as the resulting staging of the virtual agents taking part in the narrative action through cinematic staging. This includes virtual agents defined as *primary* agents within the scene, as well as *background* agents included as supporting narrative agents. Scripting agents' behaviors, whilst allowing for the inclusion of relevant background agents within a scene, still creates several problems. Firstly, the authoring time required to script the behaviors of background agents vastly increases with the amount of agents required for a scene, meaning that creating a virtual environment populated with hundreds of scripted background agents can often require more time than is available to write individual scripts. Variance and believability in terms of agents' motion and behaviors is also negatively affected by the scripting of agents. However, the simple solution of excluding background agents from the narrative staging fails to deliver a believable visualization of any narrative as background agents are used both to support a narrative action taking place and to add atmosphere to an environment [15].

There are a number of challenges to the staging of relevant behaviors for background agents such as: i) the problem of virtual agent interactions in the background of a visualized scene; ii) ensuring that the motion of background agents is both varied and believable; and iii) ensuring that background agents' behaviors are synchronized with those of the primary agents of the narrative. Hence we were motivated to tackle the problem of automating the staging of background agents in IS systems. A key objective was to remove the need to script behaviors of these agents, providing all the benefits of other crowd simulation systems (such as real-time navigation and dynamic choice of actions an agent can perform) whilst creating a believable visualization narrative.

Appears in: *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017), S. Das, E. Durfee, K. Larson, M. Winikoff (eds.), May 8–12, 2017, São Paulo, Brazil.*

The solution we outline in this paper is a generic approach that permits the dynamic generation of background agents that are able to support the narrative as it is generated by the system and that appear to perform intelligent and semantically relevant actions. This is achieved via the use of regions, which we refer to as Mise-En-Scène Regions (MISERs), which provide relevant localization for narrative agents' behaviors and which can be dynamically instantiated at run-time. MISERs are used to provide agents with sets of behaviors that are appropriate given their location, narrative context, and role.

Our approach is fully implemented in a prototype IS system using the Unreal Engine 4 game engine, using two popular TV shows as contexts of illustration: one based on *The Big Bang Theory* [4] and the other on *Community* [5]. Different episodes from each series were selected for our evaluation and are used as illustrations throughout the paper.

The paper is organised as follows: in the next section we discuss closely related work, followed by the rationale underlying our approach in section 3. In section 4 we discuss the architecture of the system in which our staging approach is implemented and discus the specification of staging regions in section 5. We illustrate the operation of the staging manager via a couple of case studies in section 6 and report our evaluation in section 7. In section 8 we discuss conclusions.

2. RELATED WORK

One area of research which has benefited from the recent rise of available computational processing resources is the study of complex autonomous agents populations. The study of human crowd dynamics has received interest in many research fields, such as architecture, safety and security, computer graphics, sociology and interactive entertainment. For instance, Shao and Terzopoulos [24] address the problem of simulating the behavior of virtual pedestrians in a simulated train station, whilst Sung *et al.* [26] investigate the computational scalability of crowds with increasingly complex behaviors; and Schultz and Fricke [23] are working towards improving passenger management and movements at airports. Navarro *et al.*[18] proposed a flexible framework to overcome the trade-off that large scale agent-based simulations typically face between the level of detail in agent representation and the scalability in the number of agents that can be simulated. Similarly, our approach facilitates the implementation of autonomous agents behaviors for scalable populations, as asynchronous crowds.

In recent years, popular game engine technology has been used for agent-based crowd simulations within similar traditional contexts of application such as an airport terminal [27]. This gaming technology provides all the required software components (graphics, physics, animations, interaction, etc) to create a scalable context to implement AI techniques to support the definition of agents' behaviors as well as provide scalability with increasing number of agents. The implementation of IS systems has followed the same approach of relying on rich game engine technologies. It has also led to extensions to their functionality: e.g. Hartholt *et al.* [11] proposes the ICT Virtual Human Toolkit, which offers a flexible framework for exploring a variety of different types of virtual human systems. Similarly, Shoulson and Badler [25] present a framework for controlling the varied activities of groups of background agents which is built upon an event-centric agent control model.

Closer to our work, the use of virtual agents to provide ambient life in a virtual environment was described by de Sevin *et al.* [7] in introducing the concept of *Smart Zones* used to define consistent autonomous agents' behaviors. Their aim was to create *living scenes* within a virtual environment to provide a sense of life to the otherwise empty virtual worlds. The work we present in this paper was inspired by this concept though our context of application goes beyond the simple need to use autonomous agents as background *fillers* for generic virtual environments. Ennis and O'Sullivan [8] observed that background agents are becoming a more integral part of games and identified a number of characteristics influencing the believability of virtual agents is these contexts. In particular they emphasized the role of meaningful interactions between background agents, such as conversations, as playing a significant role in adding plausibility, or a sense of presence, to a real-time simulation. However, it is not obvious from their work how best to generate and vary these kinds of groups. Limberge *et al.* [15] present a supporting agents' director for IS systems, which allows for enhancing the dramatization, improving the realism, and extending the duration of the narratives being generated. The narrative controller generates and manages different types of supporting agents, where some are able to interact with the main agents, while others are used to populate the background scene. Jan and Traum [12] describe an algorithm that generates behavior for background agents in conversation that supports dynamic changes to conversation group structure.

In the next section, we will detail the requirements of defining complex autonomous agents' behaviors within the context of IS systems, though we here provided some insights into previous works to establish the remit of our approach within the contexts of agent-based crowd simulations and IS dramatisation in interactive virtual environments.

3. RATIONALE: Mise-En-Scène Region

The problem we address is automating the staging of the behaviors of background agents in IS systems. Here we outline the rationale underlying it. However, first some terminology: we differentiate between named agents that participate directly in the ongoing narrative as *primary* agents; whereas those anonymous agents, who serve solely to support and enhance the presentation of narrative content are referred to as *background* agents (e.g. by populating the background of a scene to create atmosphere, or partially contributing to the development of the primary agents' actions). We also differentiate between: a *virtual environment*, which can consist of one or more *virtual stages* and where each virtual stage can contain one or more *locations* in it.

The general idea underlying our approach to staging the actions of both types of agents is via the use of regions within a virtual stage which are able to control agent movement into and out of the region, the number of agents in them, and also to assign appropriate behaviors and attributes for agents to perform whilst they are in those regions. We refer to these regions as **Mise**-En-Scene **R**egions (MISERs). The strength of the MISER approach is that it enables dynamic control of agent behavior (i.e. avoids the overhead of precise scripting) whilst still allowing their behaviors to be meaningful in the context of the narrative storyline and relevant to the narrative visualization.

In this work we have focused in particular on the generation and staging of episodic serial dramas and soap operas

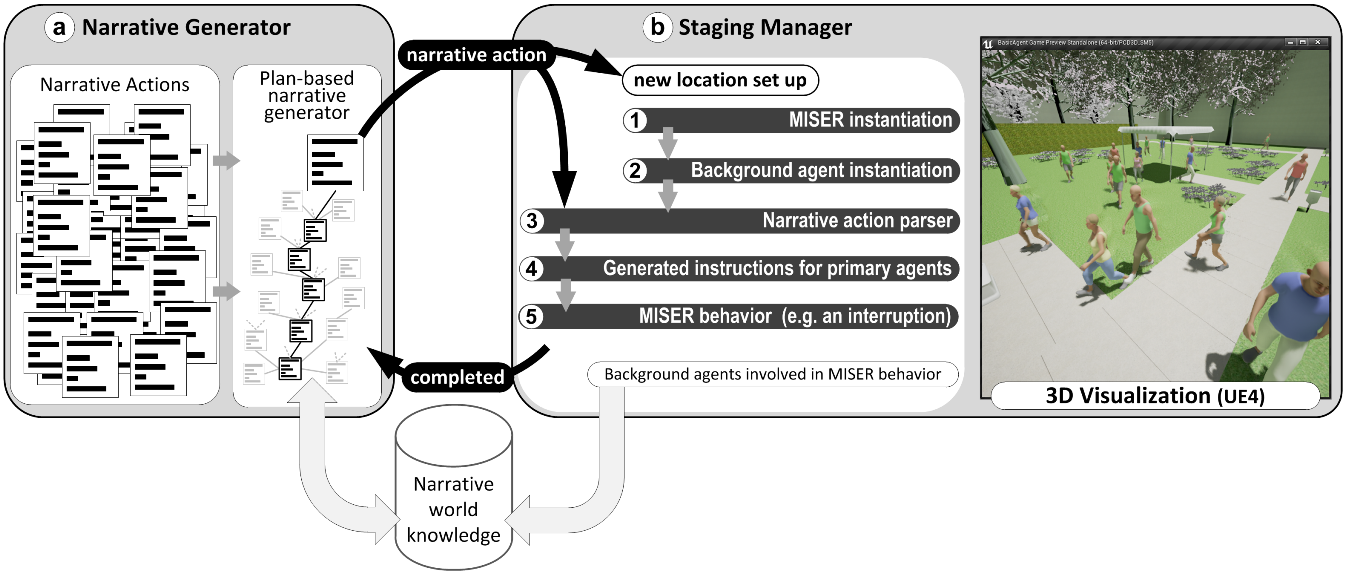


Figure 1: Architecture of the Interactive Storytelling System. There are two central components: a Narrative Generator; and a Staging Manager. Narrative World Knowledge is accessed via a shared Knowledge Base.

such as popular situation comedies (e.g. Big Bang Theory [4] and Community [5]), police procedurals (e.g. CSI: Crime Scene Investigation [2]) and medical dramas (e.g. ER [1] and House [3]). We focused on this genre due to their repetitive nature: they frequently feature different instances of the same elementary narrative actions and these narrative actions tend to be staged in a number of locations with which the audience becomes familiar.

We analyzed in detail a number of episodes from a selection of TV series from the serial drama genre and identified the following features of relevance for MISER staging:

- **Number of background agents and regions**: we observed that some episodic drama locations never have any background agents (e.g. Monica's apartment in *Friends* and Apartment 4a in the *Big Bang Theory* which only ever feature primary agents participating in a narrative action) whereas other locations allow for increasing numbers of regions for different types of background agent activities. For example, the coffee shop in *Friends*, and the goth bar in *The Big Bang Theory*[1] have sufficient room for different regions of activity in the background. An even larger region is the quadrangle in *Community* which can accommodate large background agent populations.
- **Appropriate background agent behaviors**: we observed that background agents can move in and out of different regions. For example, in the *Big Bang Theory* bar scene, there is a bar counter, dance floor and seating areas; in the *Community* quadrangle scene there are stalls, statues, benches, a car park, tents, path intersections and so on. These regions work as areas of interest to which background agents are drawn into and exit from on the basis of their role (such as student or staff, employer or customer). Also, whilst in these regions agents adopt appropriate behaviors both for the region and for their role. For example, in *The Big Bang Theory* bar scene, customers can move onto the dance floor and start dancing, whereas staff serve drinks and clear tables.
- **Interaction with primary agents**: it can occur that background agents enter into regions where primary agents are involved in behaviors related to a narrative action. In such situations, rather than adopting generic behaviors, appropriate to their role, they may interact in some way with the primary agents, either to make some form of interruption to convey information relevant to the narrative, or to receive information that is communicated to them. An example of this is explored in the *Community* case study and staging of the `discuss-shout-all-work` narrative action, in section 6.1.

Hence we formulated an approach that allowed for the creation of location-specific regions with related role-dependent agent behaviors and which also allowed for the creation of regions to support primary agents and allow for interaction with them. We discuss this further in section 5.

4. IS SYSTEM ARCHITECTURE

Our MISER dynamic staging approach is fully implemented within an IS system using the Unreal Engine 4 game engine for 3D realtime visualization. We start here by outlining the architecture of the IS system in which our approach is embedded. An overview of the system is shown in Fig 1: it features a Narrative Generator and a Staging Manager which operate in a generation and "execution" (i.e. visualization) loop. The key features of these components, and the communication flows between them, are detailed below:

4.1 Narrative Generator

As AI planning techniques have been used for narrative generation in many IS systems [6, 21, 22], we have adopted a plan-based approach in our prototype system, however we emphasise that our staging control mechanism is generic and is suitable for use in IS systems irrespective of how the narrative is generated. Within this framework a plan can be viewed as a sequence of pre- and post-condition actions (as described in [22]). At run-time, narrative plans are generated incrementally using an approach similar to [20]. This allows for monitoring and updating of the narrative state

---

[1] Big Bang Theory (Season 3 episode 2) Gothowitz Deviation

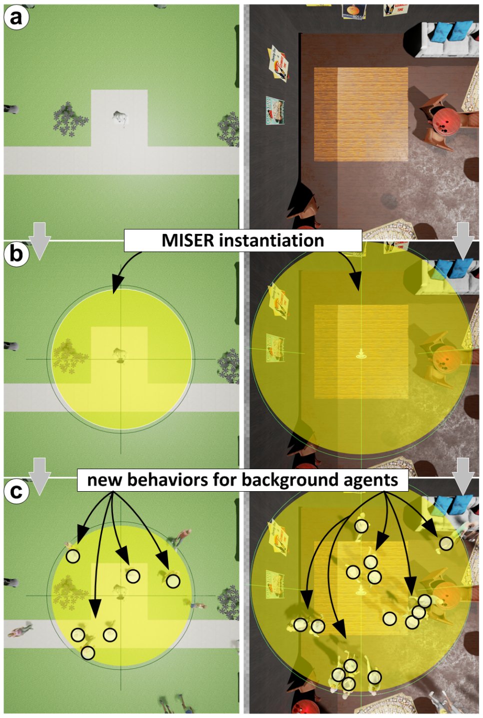


Figure 2: LOCATION-based MISER instantiation for *Community* (left) and *Big Bang Theory* (right). Top-down view of: empty visual stage (a); MISER instantiation (b); background agents instantiated (c)

as actions are executed and re-generation of the remaining narrative should this be required (e.g. as a result of user interaction). As each next action of the narrative plan is generated it is passed to the Staging Manager (see Fig 1).

After each narrative action is sent to the Staging Manager, the Narrative Generator waits for a response: either signalling that the state of the narrative domain has been updated (i.e. in the shared knowledge base as shown in Fig 1) or a simple acknowledgement. The information update is required whenever the Staging Manager draws a background agent into the narrative to support its staging (e.g. the interruption action discussed in section 6 results in a change from `background` to `primary` for that agent).

4.2 Staging Manager

The Staging Manager is responsible for the visualization of the narrative actions as they are received from the Narrative Generator and communicating with the narrative generator (receiving narrative actions and sending acknowledgements once staging is completed) and the shared knowledge base.

Narrative actions include information about the primary agents involved in the action to be staged as well as an appropriate location. For staging of the first action in a narrative or later actions that require a change of location, the staging manager is responsible for setting up the scene. As shown in Fig 1(b), this includes the instantiation of MISERs in the location and the instantiation of the primary agents for the action along with an appropriate number of background agents for the relevant virtual stage (the specification and instantiation of MISERs are discussed in more detail in sections 5 and 6). Each agent is assigned a unique controller which is responsible for the following:

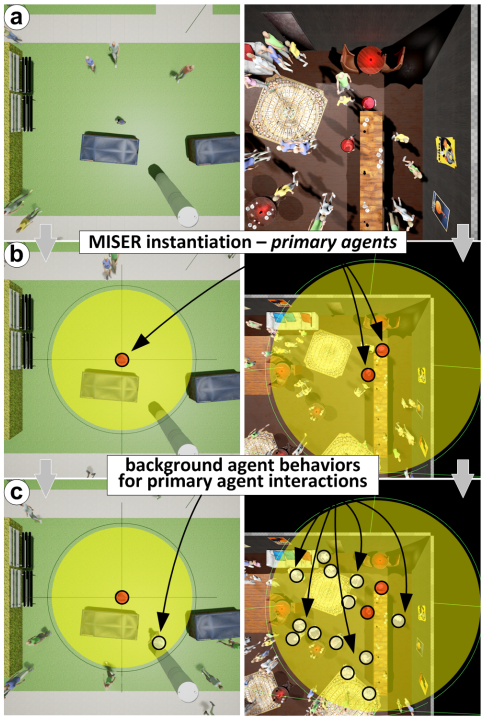


Figure 3: ACTION-based MISER instantiation for *Community* (left) and *Big Bang Theory* (right). Top-down view of: empty visual stage (a); MISER instantiation (b); background agents instantiated (c)

- **Behavior Management**: for those behaviors that agents can perform in the virtual stage as part of staging a narrative action. For primary agents involved in the action this relates directly to the content being staged. For background agents this relates to appropriate activities in different locations which serve to support the narrative content or to enhance the staging of the action in some way. For example in *Community*, background agent behaviors include: wandering around in the quadrangle; communicating with other agents at the statue; listening to the primary agents, located at a stall.
- **Motion Management**: this relates to the specific game engine mechanisms used to compute movement and animation speed. Examples of these behaviors include: perception of agents in the environment for collision detection and creation of navigation paths the agent will follow to specified locations in the virtual environment.

In some instances agent behavior and motion can be combined. A common example of this is agent movement between points in a virtual stage. Movement to the destination is specified as part of a behavior, with navigation and path finding being handled by motion control.

5. SYSTEM CONTENT SPECIFICATION

In order to implement an IS system with MISER supported dynamic agent staging (as shown in Fig 1), co-ordination is required over the specification of some content used by both parts of the system as follows:

- **Narrative Actions**: Within the Narrative Generator domain model, the narrative actions are parameterized pre-

and post-condition actions, specified using PDDL3.0 [9]. At run-time they are instantiated to domain objects (such as agents and locations), as the narrative is generated and the action name and instantiated parameters are communicated to the Staging Component. Thus co-ordination is required between the specification of both components to ensure appropriate action parsing (see section 6).
- **State Information:** Within a plan-based approach to narrative generation, states of the narrative world are represented as sets of ground predicates (e.g. relating to relationships, personality traits, and narrative events). As part of the specification, any predicates which are important for the staging of a given narrative action are tagged and then communicated when that action is staged.
- **Agent Role and Location Information**: Within the Narrative Planning Domain Model, agent *role* and *locations* can both be specified as object typing information (a feature of PDDL). This allows for hierarchies of types to be specified and reasoned about during narrative generation but also used when required in action staging. The type hierarchy allows for flexibility of agent and location choice: the type of agent or location can be left more or less general depending on the importance in the context of the narrative. As illustration, part of the type hierarchy for the *Community* domain is shown in Fig 4. If it is important in the context of the narrative that an action is staged at a particular location (e.g. the stall in the quadrangle) then the planner will instantiate the location parameter accordingly to a constant of that type, otherwise choice of location is delegated to the Staging Manager (e.g. type `quadrangle` allows for variability).

5.1 MISER Specification

Central to our automated approach to staging of virtual agent actions is the use of regions, referred to as MISERs, whose properties are specified as part of the IS system content creation, ready for instantiation at run-time (see section 6). We distinguish between two types of MISER:

- **Action-based**: which are tied to the narrative action being staged and its primary agents and whose actions are centered around a location instantiated at run-time.
- **Location-based**: which are specified for any location in a virtual environment deemed to be of potential use during the run-time staging of any narrative action.

An important feature of our MISER approach to staging of background agents is an agent's *role*. For both primary and background agents their role is defined in terms of one or more attributes which are appropriate for the narrative context. Examples of these include generic properties such as gender, age, profession, and also more specialized roles based on the particular narrative setting (e.g. for the *Community* drama a wide range of student and faculty roles might be used, along with gender and seniority). Agents' roles are used to ensure that the generic behavior of the agents remains consistent and relevant with their context of acting.

Based on our analysis of the episodic drama genre (section 3), we identified the following core elements which must be specified for both action- and location-based MISER:

- **Role Ratio**: This is a ratio of the different types of agents (as specified by their role) expected within a particular region. For a particular MISER, different ratios can be specified depending on different agent roles, which can then be used to govern the make-up of the overall population of agents within a MISER. For example, in the *Big Bang Theory* bar scene, the requirement might be to maintain an equal ratio of male to female agents, whereas in the *Community* campus quadrangle scene a higher ratio of students to faculty may be required.

```
(:types agent location - object
      student faculty - agent
      femaleStudent maleStudent - student
      femaleFaculty maleFaculty - faculty
      location - object
      quadrangle cafeteria - location
      stall statue tent bench - quadrangle
      counter booth - cafeteria ...  )
```

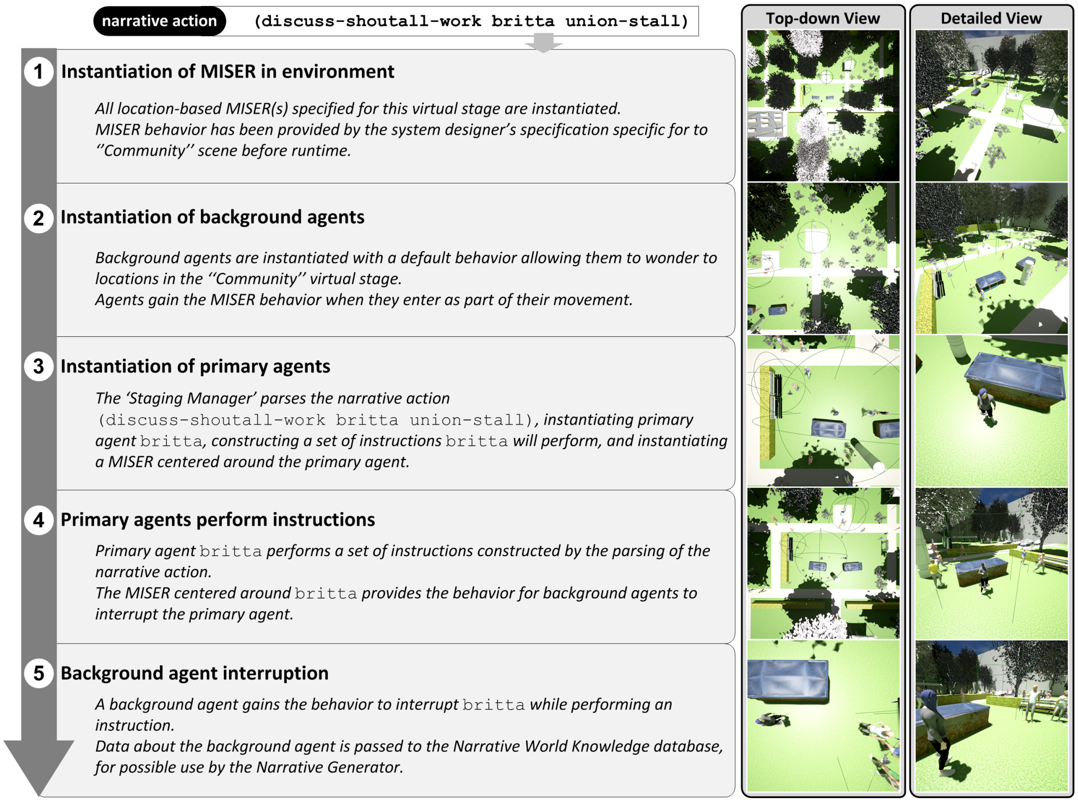
Figure 4: Part of the type hierarchy for the Community Interactive Storytelling Domain Model.

- **Expected Agent Population**: This is a value that represents how many agents should be inside each MISER at any given time. It is important to note that this is a desired number of agents used to constrain the overall number of agents within a MISER.
- **Location**: For location-based MISERs the location is specified and tied to a precise location. For action-based MISERs this is left unspecified and a precise location is supplied at run-time: either from the narrative planner if the location is important in the context of the narrative, otherwise the precise location for this MISER is instantiated at run-time by the Staging Manager.
- **Behaviors**: These are low-level actions that background agents may perform and properties they will gain upon entering the region (e.g. gaining one of a number of dancing styles when entering a dance floor depending on the region in *The Big Bang Theory* bar scene).

In our implemented approach, for each MISER, the specified agents' role ratio and expected agent population are used together to assign a cost to entering it. Then, as agents with different roles move around a scene, they can be dynamically encouraged (or discouraged) from entering a MISER by appropriate adjustments

6. MISER SUPPORTED STAGING

At run-time, narrative action staging is under the control of the staging Manager, which operates in an execution loop, receiving narrative actions one at a time from the Narrative Generator. The main steps in this process are shown in Fig 1(b), with two different possible control flows. One occurs whenever a narrative action is received which requires the narrative setting to change to a different location, (and also for the staging of the first action in a narrative), the staging manager is responsible for the setting up of the virtual stage (steps 1-5), The other occurs whenever a series of successive actions are all set in the same virtual stage then there is no need to set up the stage every time and the actions can be staged within the established virtual stage (so for example, the same background agents continue to populate the scene) and hence only require parsing of the narrative action and staging related to the behaviors of the primary agents (steps 3, 4, 5 in Fig 1(b)).



1 **Instantiation of MISER in environment**

All location-based MISER(s) specified for this virtual stage are instantiated. MISER behavior has been provided by the system designer's specification specific for to ''Community'' scene before runtime.

2 **Instantiation of background agents**

Background agents are instantiated with a default behavior allowing them to wonder to locations in the ''Community'' virtual stage. Agents gain the MISER behavior when they enter as part of their movement.

3 **Instantiation of primary agents**

The 'Staging Manager' parses the narrative action `(discuss-shoutall-work britta union-stall)`*, instantiating primary agent* `britta`*, constructing a set of instructions* `britta` *will perform, and instantiating a MISER centered around the primary agent.*

4 **Primary agents perform instructions**

Primary agent `britta` *performs a set of instructions constructed by the parsing of the narrative action. The MISER centered around* `britta` *provides the behavior for background agents to interrupt the primary agent.*

5 **Background agent interruption**

A background agent gains the behavior to interrupt `britta` *while performing an instruction. Data about the background agent is passed to the Narrative World Knowledge database, for possible use by the Narrative Generator.*

Figure 5: Example of the process of MISER instantiation and staging for the `union-stall` location within the *Community* quadrangle virtual stage for an input narrative action received from the narrative generator.

In the following sub-sections we work through two detailed case studies aimed at illustrating the process of narrative staging and MISER instantiation at run-time.

6.1 "Community" Narrative Staging Example

As discussed earlier (section 4.1), narrative plans are generated incrementally and then narrative actions are passed one at a time to be staged in the virtual environment. For example, consider the staging of the *Community* narrative world, shown in Fig 5 and the action:

`(discuss-shoutall-work britta union-stall)`

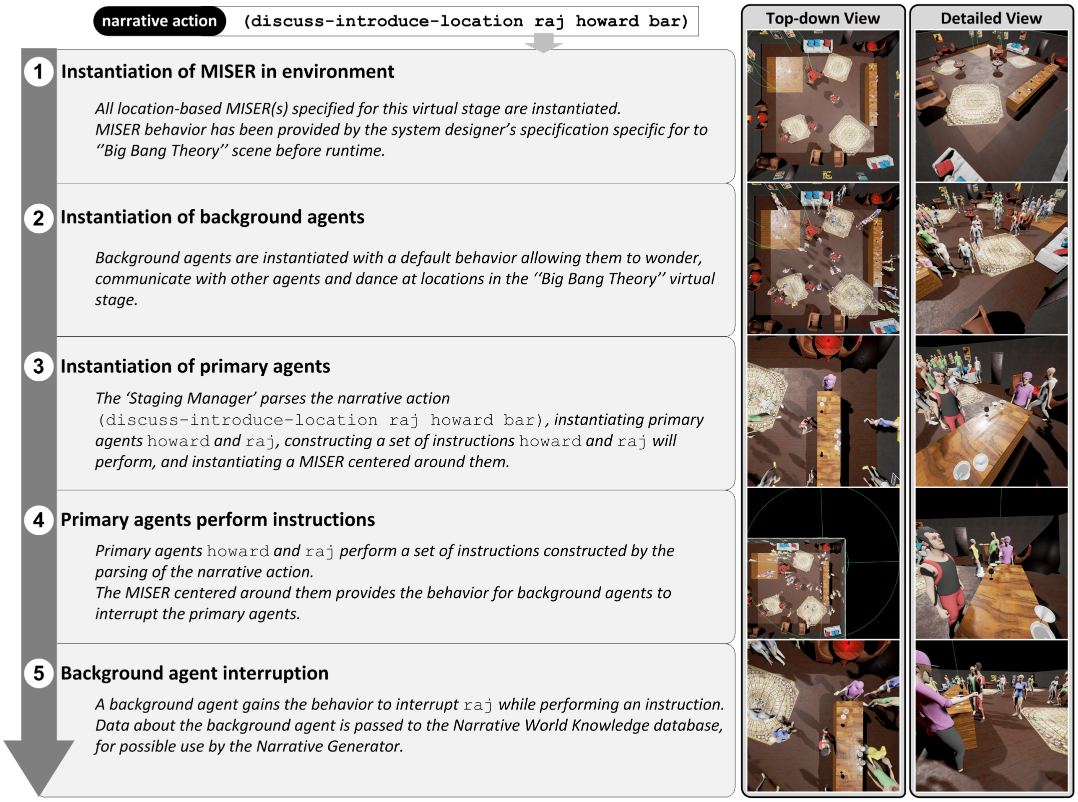
Suppose that the location for this action, namely the union-stall location within the quadrangle, represents a change of location and hence the staging manager is required to set up all aspects of the virtual stage. From Fig 1 we see that this involves firstly MISER instantiation at all locations within the quadrangle virtual stage (step 1). Then the next step is to instantiate all the required background agents. The precise number of agents and their role ratios are provided from the specification and they are each assigned a low-level controller within the game engine which handles behaviors and motion (physics, animations etc). This process, of **location-based** MISER instantiation, is illustrated in Fig 2.

Subsequently the narrative action is parsed and from this a set of low-level instructions are built. These are created by analyzing the semantic knowledge of the narrative action, the primary agents named in the narrative action parameters, and the subject of the semantic knowledge. The sets of instructions for each MISER are those specified as part of the IS content creation (as discussed in section 5.1). Low-level instructions for agents to perform on the virtual stage include moving to different locations, performing specific animations or vocalizing appropriate dialogue. The instructions are then passed to the relevant agents, as shown Fig 5.

For example, once the action `(discuss-shoutall-work britta union-stall)` is parsed, it is identified as a discussion with a single primary agent named `britta`, and with the topic of `work`. From this knowledge, the set of low-level instructions are retrieved from the specification, and appropriate work related dialogue is instantiated from a text template and passed to agent `britta` to perform.

As the low-level instructions for agents to perform are constructed, a new MISER is instantiated using the information determined from the parsing of the narrative action. This process, called action-based MISER instantiation is illustrated in Fig 3. The action-based MISER is centered around the primary agents involved in the high-level action whose behaviors have been pre-specified during system design. For this example action, the MISER behavior requires that some background agents in the "Community" scene interrupt a primary agent in the midst of speaking to agents around them. This behavior has been specified via the use of the `discuss` prefix on the action name signifying an action that involves agent discourse. When this narrative action, `(discuss-shoutall-work britta union-stall)`, is parsed and a MISER centered around agent `britta` is instantiated, the visualization chooses the "interruption" behavior to be used, as it meets the criteria as specified.

As the agent instructions are being performed and visualized on screen, our MISER centered around the primary agents involved in these actions continues to provide a be-



`(discuss-introduce-location raj howard bar)`

1 **Instantiation of MISER in environment**

All location-based MISER(s) specified for this virtual stage are instantiated. MISER behavior has been provided by the system designer's specification specific for to "Big Bang Theory" scene before runtime.

2 **Instantiation of background agents**

Background agents are instantiated with a default behavior allowing them to wonder, communicate with other agents and dance at locations in the "Big Bang Theory" virtual stage.

3 **Instantiation of primary agents**

The 'Staging Manager' parses the narrative action `(discuss-introduce-location raj howard bar)`*, instantiating primary agents* `howard` *and* `raj`*, constructing a set of instructions* `howard` *and* `raj` *will perform, and instantiating a MISER centered around them.*

4 **Primary agents perform instructions**

Primary agents `howard` *and* `raj` *perform a set of instructions constructed by the parsing of the narrative action. The MISER centered around them provides the behavior for background agents to interrupt the primary agents.*

5 **Background agent interruption**

A background agent gains the behavior to interrupt `raj` *while performing an instruction. Data about the background agent is passed to the Narrative World Knowledge database, for possible use by the Narrative Generator.*

Figure 6: Example of the process of MISER instantiation and staging for the bar scene virtual stage within our *Big Bang Theory* environment, for an input narrative action received from the narrative generator.

havior for any background agents that enter its space. If a background agent performs a behavior that interacts with the primary agent(s) and this behavior was provided by the MISER, the visualization system stores information about that agent to be passed back to the narrative planner. This has the effect of "promoting" the background agent involved in the MISER behavior to a primary agent as they are no longer anonymous and functionally equivalent to all other background agents. As a result this agent now becomes available to the narrative planner for future narrative plan generation. In our example, a background agent could enter the MISER that has been centered around primary agent `britta`, gaining the "interruption" behavior defined by the system designer. The background agent carries out this behavior and interrupts `britta` while performing an instruction. Information about this background agent is stored by the staging manager and is sent to the narrative planner on completion of agent `britta`'s instructions. For the remainder of the system runtime this background agent has the status of a primary agent and hence is available for the narrative planner for subsequent narrative generation and instantiation of narrative actions that require an agent of that type should it be necessary.

6.2 "Big Bang Theory" Staging Example

As another illustration, here we consider the staging of narrative actions on a smaller virtual stage based on a bar scene in the *Big Bang Theory* narrative world, as shown in Fig 6 and for the narrative action:

`(discuss-introduce-location raj howard bar)`

Like the *Community* scene, for the initial setting up of the virtual stage, all the pre-specified location-based MISERs are instantiated at named locations for that virtual stage. Also created are background agents with default behaviors appropriate for their role and with numbers and role ratios as specified as part of the system content.

The action `(discuss-introduce-location raj howard bar)` is parsed, and identified as a discussion with a two primary agents named `raj` and `howard`, and with the topic of `introduce-location`. These agents are created, the set of instructions governing behavior are retrieved from the specification, and dialogue related to introducing the location is instantiated from a text template and passed to the agents involved to perform, in this case `raj` and `howard`.

As with the *Community*, a new MISER is instantiated using the information from parsing the narrative action. In this case the action-based MISER is centered around both `raj` and `howard` with its behavior selected from a list of available MISER behaviors for the *Big Bang Theory* virtual stage. A behavior is selected by the staging manager based on criteria defined by the system designer and the knowledge about the narrative action that has be determined. For our example, a system designer has created a MISER behavior that allows background agents in our *Big Bang Theory* environment to interrupt primary agents engaged in a conversation. This is specified via the action name prefix: `discuss`.

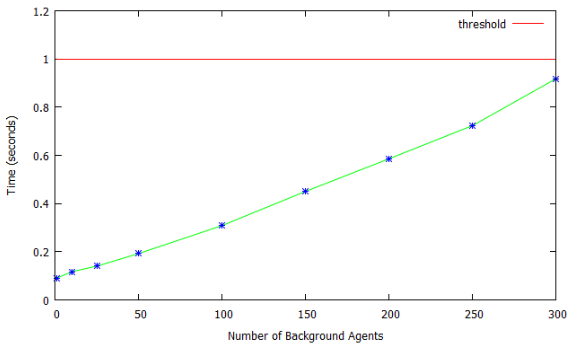


Figure 7: Impact of increases in number of background agents created on system performance (time in seconds). Performance is acceptable and remains under the required threshold. See text for details.

7. QUANTITATIVE EVALUATION

Kapadia *et al.* [14] established two broad categories for the analysis and evaluation of crowd simulations: (i) comparing simulations to real world data, is not a viable evaluation mechanism in the context of our IS application, as we are not attempting to visually recreate existing scenes from sitcom episodes; (ii) using statistical tools to analyze simulations is appropriate here. Hence we chose to evaluate the impact of background agents in our approach using objective metrics.

7.1 Performance Scalability

We were motivated to demonstrate that the performance of our MISER approach to dynamic staging of background agents would scale to the numbers of agents that might be seen in our target narrative genre (episodic dramas). In order to assess the potential scalability of our approach to the generation of sets of MISERs for a given narrative instantiation, we have evaluated the time taken to generate the overall staging for narrative actions with increasing numbers of MISERs in terms of the generation of increasing numbers of background agents. Fig 7 provides an overview of the results of generating increasing numbers of background agents from 1 to 300. In the figure we have plotted the times in relation to the threshold of 1s which is the maximum acceptable time limit allowed for the staging of the virtual scene. We conclude that this level of performance is acceptable for operation within our run-time system. Further we observe that in practice the number of background agents will be considerably lower than the 300 agent limit we used. For example, in the *Community* sitcom examples that we analysed, which were set in the college quadrangle, the number of background agents were in the 10's rather than the 100's.

7.2 Background Agent Movement

We also assessed how well our approach promoted background agent movement around the virtual stages. In order to demonstrate the degree of movement of background agents within a virtual stage we ran simulations within both the *Community* and *Big Bang Theory* virtual stages. For both narrative worlds we used an appropriate number of background agents for the size of the virtual stage: 100 for *Community* and 25 for the smaller *Big Bang Theory* bar virtual stage. For both worlds the simulation was run for 5 minutes of a narrative instantiation. The results are shown in Fig 8 and 9 with top down views of the stages: on the left hand side are the empty stage and on the right hand side agent movement paths are shown highlighted in red. The figures clearly show virtual agent distribution throughout the stage whilst moving around obstacles (e.g. trees, furniture etc).

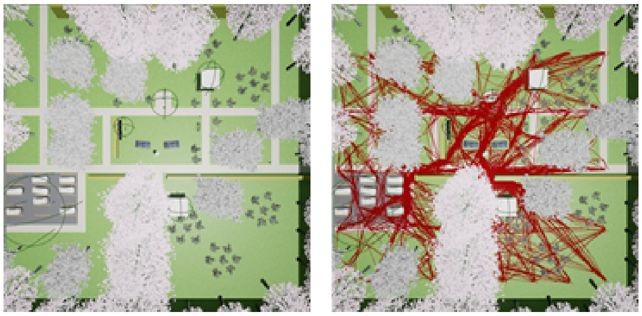


Figure 8: Top-down overview of the *Community* virtual environment, showing the empty stage (left) and the distribution of the agents' motions (right).

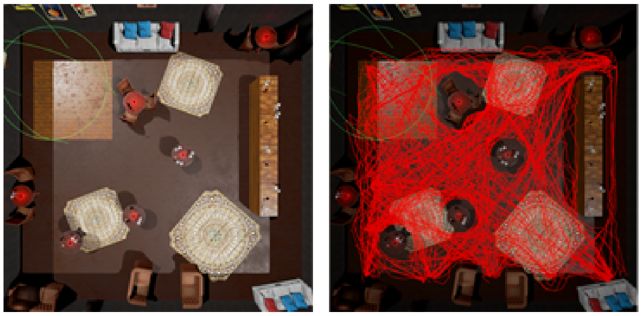


Figure 9: Top-down overview of *The Big Bang Theory* virtual environment, showing the empty stage (left) the distribution of the agents' motions (right).

8. CONCLUSIONS

In this paper, we have introduced a novel approach to the dynamic staging of background agents behaviors and its integration as part of a typical plan-based IS system. We illustrated the work through two detailed examples of narrative instantiations inspired by the sitcom genre from two different TV series demonstrating a variety of narrative actions. Finally, we reported the results of our preliminary evaluation which nevertheless clearly demonstrate the potential of the MISER approach.

Although we have been able to demonstrate the benefits of our approach within this narrative context, we would like to extend our evaluation to more qualitative aspects of users experimenting with our IS system through a wider variety of narrative instantiations. Our approach manages effectively interactions between primary and background agents of the interactive narrative generated, but our solution should also account for interactions between the agents and the user. This context of application is one of the challenges faced by immersive interactive narratives where the user is free to evolve within virtual narrative worlds.

REFERENCES

[1] ER. TV series (Warner Brothers), 1994-2009. Warner Brothers.

[2] CSI: Crime Scene Investigation. TV series (CBS), 2000-2015.

[3] House MD. TV series (Fox TV), 2004-2012.

[4] The Big Bang Theory. TV series (CBS), 2007-Present.

[5] Community. TV series (NBC), 2009-2015.

[6] R. Aylett, J. Dias, and A. Paiva. An Affectively Driven Planner for Synthetic Characters. In *Proceedings of 16th Int. Conference on Automated Planning and Scheduling (ICAPS)*, 2006.

[7] E. de Sevin, C. Chopinaud, and C. Mars. Smart Zones to Create the Ambience of Life. *Game AI Pro 2: Collected Wisdom of Game AI Professionals*, page 89, 2015.

[8] C. Ennis and C. O'Sullivan. Perceptually plausible formations for virtual conversers. *Computer Animation and Virtual Worlds*, 23(3-4):321–329, 2012.

[9] A. Gerevini, P. Haslum, D. Long, A. Saetti, and Y. Dimopolous. Deterministic planning in the fifth international planning competition: Pddl3 and experimental evaluation of the planners. *Artificial Intelligence*, 173(5-6):619–668, April 2009.

[10] P. Gervás. Computational approaches to storytelling and creativity. *AI Magazine*, 30(3):49–62, 2009.

[11] A. Hartholt, D. Traum, S. C. Marsella, A. Shapiro, G. Stratou, A. Leuski, L.-P. Morency, and J. Gratch. All Together Now: Introducing the Virtual Human Toolkit. In *13th International Conference on Intelligent Virtual Agents*, Edinburgh, UK, Aug. 2013.

[12] D. Jan and D. R. Traum. Dialog simulation for background characters. In *Intelligent Virtual Agents*, pages 65–74. Springer, 2005.

[13] A. Jhala and R. M. Young. Cinematic Visual Discourse: Representation, Generation, and Evaluation. *IEEE Transactions on Computational Intelligence and AI in Games*, 2(2):69–81, June 2010.

[14] M. Kapadia, M. Wang, S. Singh, G. Reinman, and P. Faloutsos. Scenario space: characterizing coverage, quality, and failure of steering algorithms. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 53–62. ACM, 2011.

[15] R. Limberger, C. T. Pozzer, B. Feijo, and E. S. de Lima. Supporting characters in interactive storytelling. In *Games and Digital Entertainment (SBGAMES), 2011 Brazilian Symposium on*, pages 241–249. IEEE, 2011.

[16] J.-L. Lugrin, M. Cavazza, D. Pizzi, T. Vogt, and E. André. Exploring the Usability of Immersive Interactive Storytelling. In *the 7th ACM Symposium on Virtual Reality Software and Technology*, pages 103–110, 2010.

[17] M. Mateas and A. Stern. Structuring Content in the Façade Interactive Drama Architecture. In *Proceedings of 1st Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*, 2005.

[18] L. Navarro, F. Flacher, and V. Corruble. Dynamic level of detail for large scale agent-based urban simulations. In *The 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '11, pages 701–708, Richland, SC, 2011. International Foundation for Autonomous Agents and Multiagent Systems.

[19] A. Piacenza, F. Guerrini, N. Adami, R. Leonardi, J. Teutenberg, J. Porteous, and M. Cavazza. Changing video arrangement for constructing alternative stories. In *Proc. of the 19th ACM International Conference on Multimedia*, 2011.

[20] J. Porteous and M. Cavazza. Controlling Narrative Generation with Planning Trajectories: the Role of Constraints. In I. Iurgel, N. Zagalo, and P. Petta, editors, *Proc. of 2nd Int. Conf. on Interactive Digital Storytelling (ICIDS 2009)*, 2009.

[21] J. Porteous, F. Charles, and M. Cavazza. NetworkING: using Character Relationships for Interactive Narrative Generation. In *Proc. of 12th Int. Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2013)*, pages 595–602, 2013.

[22] M. O. Riedl and R. M. Young. Narrative Planning: Balancing Plot and Character. *Journal of Artificial Intelligence Research*, 39:217–267, 2010.

[23] M. Schultz and H. Fricke. Managing passenger handling at airport terminals. In *9th Air Traffic Management Research and Development Seminars*, 2011.

[24] W. Shao and D. Terzopoulos. Autonomous pedestrians. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 19–28. ACM, 2005.

[25] A. Shoulson and N. I. Badler. Event-centric control for background agents. In *Proc. of 4th International Conference on Interactive Digital Storytelling (ICIDS)*. 2011.

[26] M. Sung, M. Gleicher, and S. Chenney. Scalable behaviors for crowd simulation. In *Computer Graphics Forum*, volume 23, pages 519–528. Wiley Online Library, 2004.

[27] O. Szymanezyk, P. Dickinson, and T. Duckett. Towards agent-based crowd simulation in airports using games technology. In *KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications*, pages 524–533. Springer, 2011.