6
Conclusions and Future Work

6.1
Conclusions

This thesis aimed at creating a framework through which formal simulation methods could be integrated in a serious game architecture. It was argued that serious games can greatly benefit from being based on formal simulation methods. The results in this thesis contributed to increase the level of formality in the design of game dynamics, which is an important step if games are intended to be extensively used outside the entertainment realm. The thesis also contributed to the development of the InfoPAE system by implementing two working modules to test the proposed framework.

Chapters 1 and 2 enumerated the requirements of the class of serious games that simulate real situations and overviewed a few techniques and systems, selected from the areas of computer games, modeling and simulation, multi-agent systems and planning that could help fulfilling those requirements.

Chapter 3 first discussed the desirable characteristics of a framework for modeling and simulation in serious games, considering the requirements enumerated in section 1.3. The framework followed the process-oriented simulation (POS) paradigm for modeling and simulation. The results of the discussion were organized in the form of decisions, which guided the development of the Process-DEVS modeling and simulation formalism. The fact that DEVS can serve as a common basis for integrating different simulation formalisms (Vangheluwe 2000) was inherited by Process-DEVS. This showed that POS has the ability of inheriting interesting properties of object-oriented simulation (OOS). Moreover, the separation between behavior and physical state allows simulation applications to represent the physical objects of the environment in specialized data structures, such as those used in games. That capability allows the use of formal simulation models together with fast 3D rendering technology.

Chapter 4 formalized a number of ways to implement some common dynamic modeling formalisms on top of Process-DEVS. Section 4.1 presented a way in which workflows can be mapped to Process-DEVS. In fact, in the way workflow processes were defined, they represent a form of process composition where the workflow logic defines which processes are created and when. Section 4.2 discussed the issue of modularity in the domain of cell space processes and presented a formalism for dealing with it on top of Process-DEVS. It also showed how to compose cell space processes out of smaller pieces. Section 4.3 presented a formal framework in which it is possible to model multi-agent systems on top of Process-DEVS, with support for sensors and actuators. The solutions presented in these three sections suggested some patterns in which to structure processes with interesting modularity properties. Section 4.4 informally discussed these patterns, leading to interesting conclusions that still need further experiments to be fully validated.

Particularly, the contribution of section 4.2.3 is worth highlighting. It presented a formal way to compose cell space models out of smaller and independent cell space models. The separation of behavior and physical state, as well as the modular nature of POS, were essential to accomplish closure under composition in this case.

Chapter 5 described the implementation of two applications as part of the InfoPAE system. These applications were implemented on top of Process-DEVS, which allowed such a high level of modularization that almost the entire simulation model could be successfully reused by the two applications. That simulation model successfully incorporated well established formalisms, such as workflows and cell-space processes, while separating physical state and behavior, allowing it to represent the physical objects of the environment directly in the format of a 3D rendering engine. That capability allowed the use of formal simulation models together with modern 3D rendering techniques. Besides allowing the integration of independent models from different formalisms, Process-DEVS also allowed a modular definition of oil behavior, where each different aspect was implemented as an independent type of process.

Section 5.3.1 presented an interesting way to integrate cell- and vector-based models that work on the same data through the use of environment views, which were essential for isolating the logic of the dynamic models from the internals of the environment.

The *StableFpsLoop*, presented in section 5.5.2, provided a way of managing time in the presence of simulation processing peaks and when the simulation speed can be accelerated to the limits of the processing capacity. This technique showed how to keep control of the frame rate under these circumstances and how to keep the simulation time correctly synchronized when there is enough processing capacity for that.

Even though this thesis is focused on serious games, there seems to be nothing that prevents using POS or Process-DEVS in the design of entertainment games. This conclusion is based on the fact that the rendering procedure may remain totally untouched by the simulation logic. Furthermore, the technique of adjusting the frequency of periodic processes, discussed in section 5.3.2, can be used to optimize simulation performance.

In summary, the major contributions of this thesis were:

- A discussion on the requirements for modeling and simulation in the context of serious games
- The conception of the process-oriented simulation (POS) paradigm as a consequence of the discussion
- The materialization of POS in a DEVS-based formalism named Process-DEVS
- A framework for modeling cell space processes on top of Process-DEVS with composition capabilities that preserve individual independence of the sub-models

As secondary contributions, we may list:

- Mapping a workflow representation to Process-DEVS
- A framework for modeling multi-agent systems on top of Process-DEVS
- The development of a planning system and a training game for the InfoPAE system as a use case of Process-DEVS
- A technique to enable game loops to handle variable game speeds and simulation processing peaks

6.2 Future Work

As for future research, we may suggest:

- *Using the POS paradigm for modeling and simulation of real systems (outside the gaming domain).* It would be interesting to see POS being used for traditional modeling and simulation problems. It would be possible to get a more detailed comparison with OOS and AOS.
- *Extending Process-DEVS to cover the simulation of continuous processes*. In the line of hybrid systems, mentioned in section 3.2.1, it would be interesting to provide Process-DEVS with the ability of defining processes by differential equations. One possibility is to implement it based on the DEV&DESS formalism (Zeigler et al. 2000), instead of pure DEVS.
- *Providing more user-friendly languages for defining processes*. Just as in the Jason and SeSam toolkits, described in section 2.3, Process-DEVS could be equipped with a high-level language for allowing non-specialized users to define new kinds of processes.
- *Providing support for more specific workflow patterns and standards*. In section 4.1, only the basic workflow patterns were considered. More complex patterns such as loops could also be formalized on top of Process-DEVS. Moreover, the integration with workflow standards currently used by business process management (BPM) systems is certainly useful for corporations.
- *Evolving the discussion on process patterns.* More experience with different kinds of simulations could lead to the consolidation and formalization of the patterns described in section 4.4. It could also lead to the detection of other interesting process patterns.
- *Using Process-DEVS for other types of games, including entertainment games*. It would be interesting to see Process-DEVS in a game with top-quality graphics using formal simulation methods. This could also include the integration of Process-DEVS with game physics of current game engines.

- *Using Process-DEVS in a multi-player network game*. In the InfoPAE training game, all players used the same user interface. The possibility of using different player configurations could be investigated in greater detail with multi-player network games. Additionally, increasing the simulation speed in a multi-player network game would raise new problems, such as the impact of network delays.
- *Logging and debugging techniques for Process-DEVS*. Finding errors in complex simulation models is not a trivial task. This problem is made even more difficult if different formalisms are used in the same simulation, which makes logging and debugging in Process-DEVS an interesting research problem.
- *Support for distributed simulation in Process-DEVS*. As simulation models grow in complexity, so does the need for increasing the processing capacity. Distributed simulation is an active research field in practically all areas involving simulation.
- *Modeling information flow for the InfoPAE simulation*. The first actions of contingency plans in the InfoPAE system usually include alerting, reporting, evaluation and mobilization. In order to simulate these actions, it would be necessary to model multiple actors, the information they have about the situation and the communication among them. That would make the simulation more detailed and realistic.