Review report of the PhD thesis presented by Jakub Gemrot

Controlling Virtual People

The motivation of the research presented in this thesis comes from the very practical observation that currently there are no programming languages or platforms that truly support all aspects of programming behaviours of characters in games (economically). On page 152 a list of desirable features is given and it is shown that current tools do not provide natural or economic support for all these features. This seems a kind of natural conclusion of the first part of the thesis in which it is shown what are the requirements and why existing tools do not fulfill these requirements. After this chapter follows a chapter on xTREE. I can see some technical use of this chapter, but after reading the rest of the thesis I am left wondering which part could not be done without having the background of this chapter 6.
This leads to the following question:

> *Could you describe how the definitions of BDL fundamentally depend on xTREE or could BDL also be defined just using the concepts developed in chapter 6 directly without having this kind of divergence in the thesis?*

In chapter 7 a concise description is developed of BDL. However, I really missed a kind of formal description of the actual language constructs. Only on page 338 there is a kind of overview table with all constructs developed in BDL so far. It gives me the impression that BDL is more a conceptual construct and its implementation depends on which other languages one wants to model with it. Would it be the case that if one would try to model an agent programming language like 2APL with BDL constructs that one would get yet another construct in BDL?

> *The way the BDL is described in the thesis it is very much reasoned from the perspective of covering all constructs that other tools can cover rather than providing those constructs that users would find handy and necessary. Would users know which constructs to use at what time and is it possible to implement the same use case in completely different ways using BDL?*

In chapter 8 it is shown that BDL is powerful enough to cover all constructs of several existing tools and that in general it does so economically. Now, this raises a question about the metrics. It is long been argued by proponents of functional languages like LISP and Haskell that they provide very natural and economic implementations. However, people do not always find them easy to program, because they hide features that people then tend to ignore and lead to difficult to trace bugs. Thus, using less constructs does not always provide more productive programmers, because they might have more difficulty to 1) create a program that does what they want and 2) find out why a program is not doing what they thought it would do.

> *Could you (after having gone through the experience of this research) think of some metrics to also take into account these two more mental aspects of productivity involved in programming? And will these aspects disappear with sufficient training in the new language?*

Finally, in chapter 11, a methodology is described to compare different action selection mechanisms. I agree that it is very hard to get any conclusive data out of these experiments. If, at the time the CS community was moving from machine language to higher level languages, one would have done an

experiment to compare using machine language and e.g. Pascal or C it is not clear what would be the result. Now, after many years and in a time where systems have become orders of magnitude more complex it is clear machine language would not suffice. It is therefore also not very clear how to interpret the results of chapter 11. They seem to give very specific results for a very particular type of context. Unfortunately the experiments did not involve BDL, but other languages and thus the relevance for the thesis is also not completely clear. My question here becomes:

> *The presented experiment compares some very specific dialects of POSH together with Java in a setting where very specific challenges play a role (that have been addressed by yaPOSH). Thus how general is this applicable?*
> *Similarly, the question comes up very quick how much of the features should be fundamental features of a programming language and how much can be done through developing (graphical) support tools for the language.*

In order to conclude my review I first want to indicate that I really appreciate this type of applied research. It starts from current practice and problems encountered in that context. The thesis then shows that all currently available tools have some inherent shortcomings to support the type of constructions needed to implement intelligent, believable behavior in NPC. (It is a pity that table 3 giving a summary of all the (missing) features of commonly used tools only appears on page 337.) Thus there is a need to rethink the way NPC should be implemented in an economic way. Some original framework based on decision spaces is developed to support the type of constructs that need to be used by the decision making process of the NPC. This leads to the developed BDL in a natural way and gives at least the feeling that BDL is not just another programming language but specifically tailored for the context of programming NPC.

The presented thesis provides a state-of-the-art review as well as a rigorous presentation of the designed Behavior Decision Language used to model the agent decisions in reactive planning. The presented work has been in parts presented to the artificial intelligence and gaming scientific community, and has been positively accepted and cited. It would have been nice if the developed BDL would have been tested empirically rather than showing how this can be done for POSH and yaPOSH. It leaves one with some questions about the usability of this language and also whether it would really lead to increased productivity.
However, Jakub Gemrot demonstrated his capability to perform independent research work resulting in original results.

I recommend

accepting his thesis and awarding him a title of a doctor of philosophy (Ph.D).

Dr. Frank Dignum
Department of Information and Computing Sciences
Faculty of Science
Utrecht University

The Netherlands