Thesis topic: Reasoning with ontologies in real time for non player characters artificial intelligence in games.

Thesis supervisor: Bruno Bachelet, MCF HDR en Informatique  
Co-supervisors: Christophe Rey et Loïc Yon MCF en Informatique  
Laboratory: LIMOS - UMR 6158 Université Clermont Auvergne

Email: bruno.bachelet@uca.fr, christophe.rey@uca.fr, loic.yon@uca.fr  
Phone: +33 4 73 40 50 44, +33 4 70 30 43 92, +33 4 73 40 50 42

Detailed information

<table>
<thead>
<tr>
<th>Last Name</th>
<th>Bachelet</th>
<th>Rey</th>
<th>Yon</th>
</tr>
</thead>
<tbody>
<tr>
<td>First name</td>
<td>Bruno</td>
<td>Christophe</td>
<td>Loïc</td>
</tr>
<tr>
<td>Position</td>
<td>Maître de conférences en Informatique HDR</td>
<td>Maître de conférences en Informatique</td>
<td>Maître de conférences en Informatique</td>
</tr>
<tr>
<td>Management</td>
<td>Université Clermont Auvergne (UCA)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Laboratory</td>
<td>LIMOS (UMR 6158 UCA-CNRS)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| Full address | LIMOS  
Campus Universitaire des Cézeaux  
1 rue de la Chebarde  
TSA 60128 - CS 60026  
63178 Aubière Cedex |
| e-mail | bruno.bachelet@uca.fr | christophe.rey@uca.fr | loic.yon@uca.fr |
| Phone | Portable : 06 71 94 81 31  
Composante d’enseignement : 04 70 30 43 96  
LIMOS : 04 73 40 50 44 | Portable : 06 84 48 59 61  
Composante d’enseignement : 04 70 30 43 92  
LIMOS : 04 73 40 50 44 | Portable : 06 66 31 87 82  
Composante d’enseignement : 04 73 40 50 42  
LIMOS : 04 73 40 50 42 |

Key words: Reasoning, ontologies, existential rules, artificial intelligence, video games, serious games, logic programming, real time, code and performance optimization, parallel computing, metaprogramming.

Abstract: In the context of logical programming and the semantic web, the conception of real-time logical reasoning, in particular deductive, abductive and inductive reasoning from rules and ontologies, is a true challenge. The target application is the creation of a real-time logic artificial intelligence (AI) engine for video games (commercial entertainment and serious games). The first scientific objective is to show that logical AI approaches based on rules and ontologies are compatible with real-time applications. The second objective is to show the interest of these approaches in terms of AI credibility and support for software engineering, compared to the approaches commonly used in games (which are not based on a logical representation of knowledge and reasoning).
I. Context, challenges, problems

Among all the issues that must be addressed to design a video game, that of artificial intelligence (AI) is one of the oldest. It still faces many challenges, however. In skill games (car races, shooters, sports games), the AI must manage the player’s software opponents, giving the latter the impression that they have a tactic and that they react intelligently to his actions. In games where the player evolves within a story, the natural ambition is to further accentuate his feeling of interacting with software beings endowed with a certain intelligence, even if the story complicates the nature of the behavior to be generated by machine. In strategy games, computer-controlled opponents must establish a global attack strategy, implemented by local and temporary tactics, in constant adaptation to the behavior of the player. Credibility therefore requires the finest possible multi-level planning. Finally for serious games, the credibility of the software agents with whom the player interacts is also crucial. This involves, for example, the dynamic development of an educational strategy linked to the player’s profile as a learner in the game, or the generation of realistic behavior if the software agent embodies a person with whom the player is likely to interact in reality.

Artificial intelligence (AI) approaches that generate these behaviors in video games face two major issues: (i) they are very diverse and (ii) they are constantly confronted with computing resources limited by other needs (graphic, sound synthesis, network management, input and output management, etc.). Among the scientific fields underlying the AI approaches used in games, we find for example planning, finite state automata, tree research, learning (supervised, unsupervised, by reinforcement), genetic algorithms, multi-agent systems and Markov chains. An overview of all these approaches can be found in (Togelius & Yannakakis, 2018) which is one of the main academic textbooks on artificial intelligence in games.

Surprisingly, while it is well known that the logical approaches to represent knowledge and reason on this knowledge are historically at the basis of the notion of AI, these are almost absent from the techniques currently used for game AI. It seems that the video game industry accuses them of their greed in computation time which would make them incompatible with the real time constraint inherent in this type of applications. However, with the constantly evolving computing capacity of computers, one might think that the argument of insufficient resources that can be allocated to inference calculations is less and less founded.

In this context, the objective of this thesis is twofold: first, we wish to experiment in the ludic context with semantic technologies, representing the most recent logical approaches to AI, and show that the behaviors they generate may be qualitatively good or even superior in terms of credibility compared to traditionally used AI techniques, and second, we wish to take up the challenge of making them compatible with the real-time constraint by studying various techniques of code optimization: the parallelization of algorithms in order to exploit the multiple cores generally available on current processors (the use of coprocessors, of GPGPU type, is not envisaged here, because they are often mainly used by the graphic rendering); and metaprogramming to write or rewrite more specialized dynamic codes of algorithms.

II. Proposed methodology

Concerning semantic technologies, the proposition is use the framework of existential rules (Mugnier & Thomazo, 2014). Existential rules, also called datalog ±, are at the junction between
semantic technologies based on the OWL language of ontology modeling (Hitzler, Krötzsch, Parsia, Patel-Schneider, & Rudolph, 2009) and the formalism of description logics (Baader, Calvanese, McGuinness, Nardi, & Patel-Schneider, 2007), and logic programming (Nilsson & Maluszynski, 1995). They thus make it possible to model knowledge easily while ensuring, under certain conditions, polynomial time reasoning algorithms. One simple transformation, called skolemization, makes it possible to pass from knowledge in the form of existential rules to logical programs. It also makes it possible to envisage extensions in the expressiveness of modeled knowledge (addition of negation for example) at the cost of potentially greater complexity, but not necessarily. It also provides other inference algorithms that have been proven efficient for several decades. In addition, certain problems dealt with by this rule language are close to tree research or planning problems. For all of these reasons, we think it's an ideal formalism for gaming AI.

In terms of taking into account the real-time constraint, we envisage two ways for algorithm optimization. The first is parallelization in order to reduce the theoretical complexity of algorithms in a multi-core shared memory context (typically an MIMD architecture with a multithreaded execution model) (Gupta, Pontelli, Ali, Carlsson, & Hermetegild, 2001) (Santos, 2016). Second is the use of metaprogramming to produce the most dedicated algorithm possible: in general, for each computation, some part is known at compilation time and does not evolve during the execution of the program. It is the "static" information on the structure the algorithm and the data to be processed. This kind of information is not necessarily used by the compiler, since it is not explicitly present. So by integrating this information, it is possible to write a more dedicated and therefore more efficient algorithm. The developer can manually carry out this optimization, but the metaprogramming, by exploiting the static information which will be provided to him, for example in the form of decorators, allows the automatic generation of the dedicated program, thus relieving the developer from this task, but especially allowing him to keep a high level of abstraction, which is fundamental in development. Depending on the programming language used, it is possible to produce this program dedicated at compilation time, i.e. static metaprogramming (e.g. template metaprogramming in C++ (Pereda, Hill, Mazel, & Bachelet, 2019)), or at execution time, i.e. dynamic metaprogramming (e.g. via the reflexivity of C# (Hazzard, 2012)).

In practical terms, the study should be done with professional middleware for developing multimedia applications (such as UNITY or UNREAL Engine for example). Thus, the developments carried out within the framework of this thesis can subsequently be integrated very quickly into industrial projects. In addition, the study will focus on the AI of software agents embodying non-player characters (NPCs) of human type (unlike an AI which could manage the behavior of animals or robots for example). This constraint accentuates the needs of credible AI, but also makes it possible to meet the AI needs of a more varied range of game types, notably serious games where the player interacts with humans managed by the computer. We could imagine for example a serious game whose goal would be to learn the manufacturing processes installed on a new production line where the player would have to interact with workers managed by AI.

The doctoral student's missions will therefore be as follows:

- Propose and implement a reasoning based on knowledge modeled in the form of existential rules, allowing to propose a credible artificial intelligence within the framework of an AI of human NPC.
- Study this reasoning in theory (theoretical and algorithmic complexity), and in practice (qualitative and quantitative tests).
- Study this reasoning compared to the algorithms commonly used in the video game industry.
- Propose and implement an optimization in order to make this reasoning compatible with the constraint of real time and serious games.
- Study this optimization in theory (theoretical and algorithmic complexity, reciprocal influence of the time allocated to the optimized algorithm and the quality of the results), and in practice (qualitative and quantitative tests).

The scientific issues are:
- theoretical:
  - prove the credibility of semantic AI in the field of games,
  - and prove the compatibility of semantic AIs in real time constraint;
- practical:
  - implement a real-time semantic AI library based on reasoning from existential rules.

Beyond these main missions, the doctoral student will be encouraged to develop a game prototype (for example a serious game) allowing him to facilitate the test campaigns for the AI he is going to design. A search for an industrial partner is clearly an option here. One possible choice is the WAKO Factory video game development studio, with which LIMOS has just signed a collaboration contract. Another possibility would also be to find a regional industrial partner interested in creating a game for promotional purposes or a serious game (see section III.).

III. Situation in the laboratory context

Relatively to Cap 2025, this thesis project is part of the "Generic key technologies" section of Challenge 2 and also in the transverse axis "Instrumental development". Indeed, the development of credible AI for games is an asset for the development of serious industrial games. In addition to favoring business process learning in an industrial context, the challenge is also the explanation for the use, the preservation and the transmission of business knowledge in companies. In our example of a serious game of workers training on a new production chain, the use of semantic AI necessarily induces the creation of an ontology in which business rules of operation of this production chain will be modeled. All this modeled knowledge is precious both for the operational aspect of training, but also for the inheritance aspect of business knowledge preservation. In addition, it can also be used for simulation study of the chain itself, for example for updating purposes. Thus the game dimension of this thesis subject offers a gateway to semantic technologies in the industrial world.

In terms of LIMOS scientific policy, this work is achieved within the framework of axis 2 SIC (Information and Communication Systems), and more particularly in the DSI (Data, Services and Intelligence) theme, as well as in axis 1 MAAD (Models and Algorithms of Decision Support), more particularly in the MOCA theme (Metamodelling, Continuous Optimization and Application). In the DSI theme, Christophe Rey works in the field of the semantic web (ontologies and reasonings) and its applications in various fields, such as the military field with the IBC project (Integration of knowledge bases) in partnership with Dassault Aviation within MMT project (Man
Machine Teaming, the medical field with the company Lojélis, the field of metrology with the STAM project, or video games with the company WAKO Factory. In the MOCA theme, Bruno Bachelet and Loïc Yon work on the development of generic, flexible and efficient calculation libraries, mainly for combinatorial optimization and simulation, by proposing generic metaprogramming approaches (based on C++ language templates), in collaboration with NIST (American government agency) for the development of the HedgeHog parallelization framework, and INRAE for the planning of autonomous agricultural robots as part of the SuperRob action of the I-Site CAP 20-25 project.

IV. Organization of the thesis

During the 36 months of his thesis, the doctoral student will work in close collaboration with Bruno Bachelet (HDR associate professor), as well as with Christophe Rey and Loïc Yon (associate professors). Below is an example of a work schedule.

<table>
<thead>
<tr>
<th>Dates et durations</th>
<th>Example</th>
<th>To do</th>
</tr>
</thead>
<tbody>
<tr>
<td>T début de la thèse</td>
<td>01-09-2020</td>
<td></td>
</tr>
</tbody>
</table>
| 6 months T to T + 6 months | 01-09-2020 to 28-02-2021 | - State of the art of existential rules and languages of W3C OWL and RDF (syntax and semantics, existing reasonings)  
- State of the art of AI in games  
- State of the art of parallel programming and its use in AI, and of the various types of metaprogramming.  
Objective: understand scientifically and industrially the problem of achieving semantic AI, understand the issue of real-time constraint (how to exploit |
parallelism and on which levers should metaprogramming act to produce a tailor-made dynamic code) and proposing specifications for a reasoning and its ontology adapted to the AI of NPCs.

<table>
<thead>
<tr>
<th>Time Period</th>
<th>Dates</th>
<th>Study Details</th>
</tr>
</thead>
</table>
| 6 months
T + 6 months \(\text{à} T + 12\) months | 01-03-2021 to 31-08-2021 | Theoretical study of the proposed reasoning:
- definition of reasoning
- demonstration of important results related to reasoning and used language
- translation of results into an algorithm
- study of the theoretical complexity of the problem
- study of the computational complexity of the algorithm
- comparison with existing reasoning (functionality, complexity and fields of application)
- comparison with classic video game AI

Analysis of possible optimization tracks of the algorithm and continuation of the state of the art on these tracks
- parallelization of the algorithm
- draft metaprogramming approaches (static and dynamic)

<table>
<thead>
<tr>
<th>Time Period</th>
<th>Dates</th>
<th>Study Details</th>
</tr>
</thead>
</table>
| 6 months
T + 12 months \(\text{à} T + 18\) months | 01-09-2021 to 28-02-2022 | Practical study of the proposed reasoning:
- implementation of a first version of the algorithm without taking into account the real time constraint
- construction of a first ontology
- reasoning and ontology tests
  - qualitative tests
  - quantitative tests
- identification of blocking points
- translate the real time constraint into precise constraint for development

<table>
<thead>
<tr>
<th>Time Period</th>
<th>Dates</th>
<th>Study Details</th>
</tr>
</thead>
</table>
| 6 months
T + 18 months \(\text{à} T + 24\) months | 01-03-2022 to 31-08-2022 | Theoretical study of an optimization in view of the constraint
- real time
- proposal of parallelization strategies
- proposal of metaprogramming approaches
- complexity study
<table>
<thead>
<tr>
<th>12 months</th>
<th>01-09-2022 to 31-08-2023</th>
<th>Development of this real-time optimization</th>
</tr>
</thead>
<tbody>
<tr>
<td>T + 24 months</td>
<td>to</td>
<td>Coding of a parallel version</td>
</tr>
<tr>
<td>T + 36 months</td>
<td>31-08-2023</td>
<td>&quot;Manual&quot; code optimization</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tests</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Automation of code optimization via metaprogramming</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Formatting the code as a reusable library</td>
</tr>
<tr>
<td></td>
<td>adding comments in the source code, defining a clear structure, detailed documentation, creating tutorials, integration into middleware, ...</td>
<td></td>
</tr>
</tbody>
</table>

Bibliography

2 https://man-machine-teaming.com/
3 https://www.lojelis.com/fr/
4 https://limos.fr/news_project/118
5 https://github.com/uninistgov/hedgehog
6 https://perso.isima.fr/bachelet/superrob