Reasoning with Ontologies for Non-Player Character Decision-Making in Games

EXAG - AIIDE 2022, Cal Poly Pomona, USA

PhD Student: Sylvain LAPEYRADE
Supervisors: Christophe REY, Bruno BACHELET, Loïc YON

LIMOS – Clermont Auvergne University, France

24 october 2022

Context: The strategic reasoning of virtual agents is often poor

[1] From reference books in Game AI: Yannakakis and Togelius [2018], Millington [2019] and review article: Simonov et al. [2019]

Context: The strategic reasoning of virtual agents is often poor

Game AI behavioural techniques state of the art[1]

- Ad hoc algorithms (Hard-coded, poor reusability)
- Finite-State Machines (Hard-coded, poor scaling)
- Behavioural Trees (Hard-coded, poor scaling)
- Utility-based AI (Tuning utilities can be laborious)
- Action Planning (Difficult to use, can be expensive)
- Learning-based AI (Resource intensive, black box)

[1] From reference books in Game AI: Yannakakis and Togelius [2018], Millington [2019] and review article: Simonov et al. [2019]

Context: The strategic reasoning of virtual agents is often poor

Game AI behavioural techniques state of the art[1]

- Ad hoc algorithms (Hard-coded, poor reusability)
- Finite-State Machines (Hard-coded, poor scaling)
- Behavioural Trees (Hard-coded, poor scaling)
- Utility-based AI (Tuning utilities can be laborious)
- Action Planning (Difficult to use, can be expensive)
- Learning-based AI (Resource intensive, black box)

Obversation: Logic-based methods are almost absent from SOTA

[1] From reference books in Game AI: Yannakakis and Togelius [2018], Millington [2019] and review article: Simonov et al. [2019]

Logic-based methods are almost absent

Perspective: Is the trend changing?

Logic-based methods are almost absent

Perspective: Is the trend changing?

Recent projects using logic-based methods

- **Versu**, Evans and Short [2014]
- **MKULTRA**, Horswill [2015]
- **EmbASP**, Calimeri et al. [2018]
- **UnityIIS**, Brännström and Nieves [2021]
- **VEsNA**, Gatti and Mascardi [2022]
- **ThinkEngine**, Angilica et al. [2022]

Logic-based methods are almost absent

Perspective: Is the trend changing?

Recent projects using logic-based methods

- **Versu**, Evans and Short [2014]
- **MKULTRA**, Horswill [2015]
- **EmbASP**, Calimeri et al. [2018]
- **UnityIIS**, Brännström and Nieves [2021]
- **VEsNA**, Gatti and Mascardi [2022]
- **ThinkEngine**, Angilica et al. [2022]

Obversation: This is not yet adopted by the general industry.

Declarative logic programming helps intuitive rules representation:

Transformation of a game rule

From our game prototype: "A **human** in the same room as a **monster** or a **pit**, dies"

Listing 1: The same rule in Prolog

```
1 dies(Agent):-
2     location(Agent, [X, Y]),
3     (
4         location(pit, [X, Y]) ;
5         location(monster, [X, Y])
6     ).
```

1. X is an element if X is an object or a being
2. X is a being if X is an animal or a monster
3. X is a monster if X is a dragon
4. X is an animal if X is a dog

Listing 2: Simple Ontology in Prolog

```
1 element(X):- object(X). %1
2 element(X):- being(X). %1
3
4 being(X):- animal(X). %2
5 being(X):- monster(X). %2
6
7 monster(X):- dragon(X). %3
8 animal(X) :- dog(X). %4
```

Representing Uncertainty

The *Well-Founded Semantics* (WFS) from Van Gelder et al. [1991] is a 3-valued version of the *Stable Model Semantics* from Gelfond and Lifschitz [1988] which uses *Negation as Failure (NAF)*.

Representing Uncertainty

The *Well-Founded Semantics* (WFS) from Van Gelder et al. [1991] is a 3-valued version of the *Stable Model Semantics* from Gelfond and Lifschitz [1988] which uses *Negation as Failure (NAF)*.

- In the WFS a proposition can be *true*, *false* or *undefined*
- *Undefined* values are used for ignorance or contradictory info

Representing Uncertainty

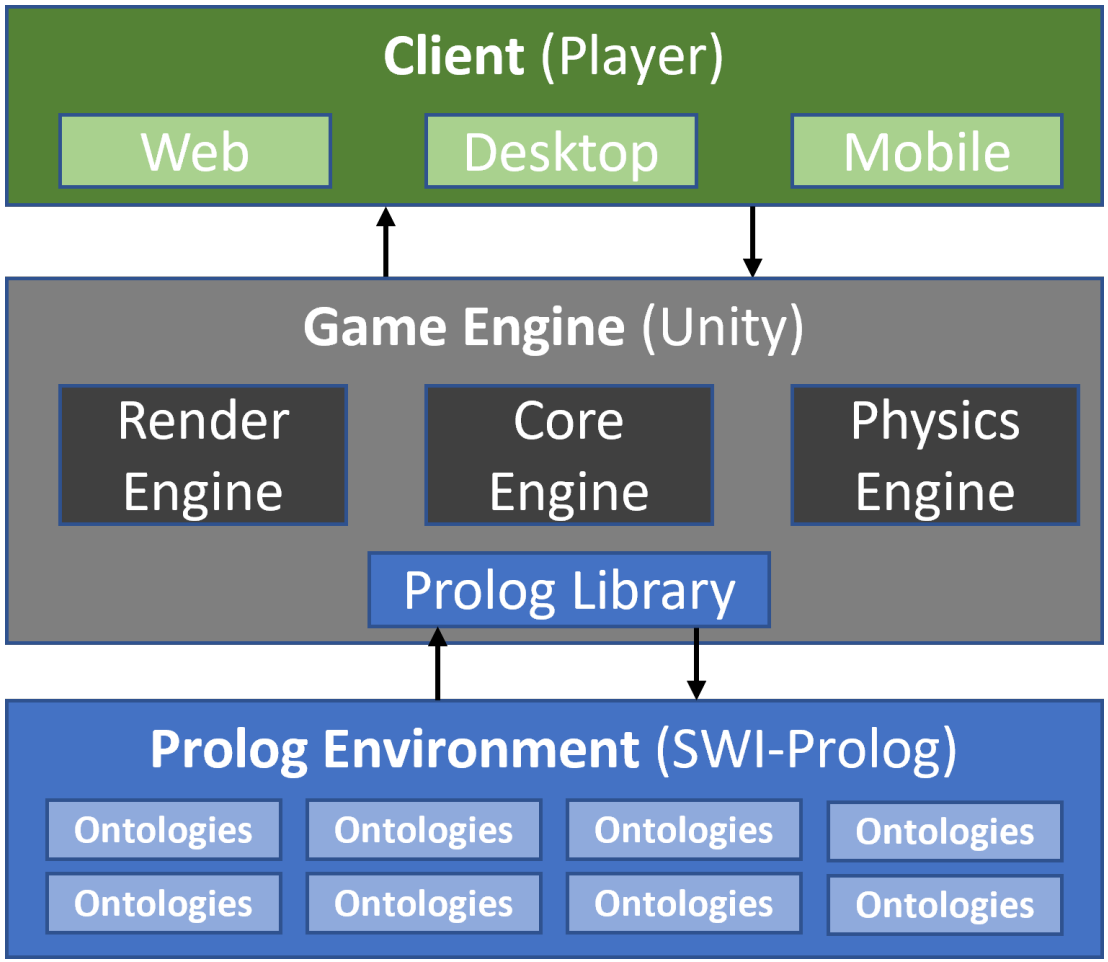
The *Well-Founded Semantics* (WFS) from Van Gelder et al. [1991] is a 3-valued version of the *Stable Model Semantics* from Gelfond and Lifschitz [1988] which uses *Negation as Failure (NAF)*.

- In the WFS a proposition can be *true*, *false* or *undefined*
- *Undefined* values are used for ignorance or contradictory info

Listing 5: Use of the Well-Founded Semantics in SWI-Prolog

```
1 is_true(Atom):- call_delays(Atom, true).
2 is_false(Atom):- \+ Atom.
3 is_undefined(Atom):-
4     call_delays(Atom, Condition),
5     Condition \== true.
```

Global Architecture



Architecture of the integration of a logic programming environment in a game engine

Interface between Prolog & Game Engine

Connecting Prolog to Unity seems like a recurring endeavour:

- **SwiPlCs** – "official" SWI-Prolog interface, *last updated 2013*
  `github.com/SWI-Prolog/contrib-swiplcs`
- **UnityProlog** – Ian Horswill, *last updated 2017*
  `github.com/ianhorswill/UnityProlog`
- **BackTraQ** – *"Prolog like", last updated 2019*
  `github.com/FacticiusVir/BacktraQ`
- **Yield Prolog** – *"Prolog like", last updated 2019*
  `sourceforge.net/projects/yieldprolog`
- **CSharpProlog** – J.Pool & J.Sakamoto, *last updated 2020*
  `github.com/jsakamoto/CSharpProlog`
- **Pengines.Client** – *F# alternative, last updated 2021(!)*
  `github.com/ninjarobot/Pengines.Client`

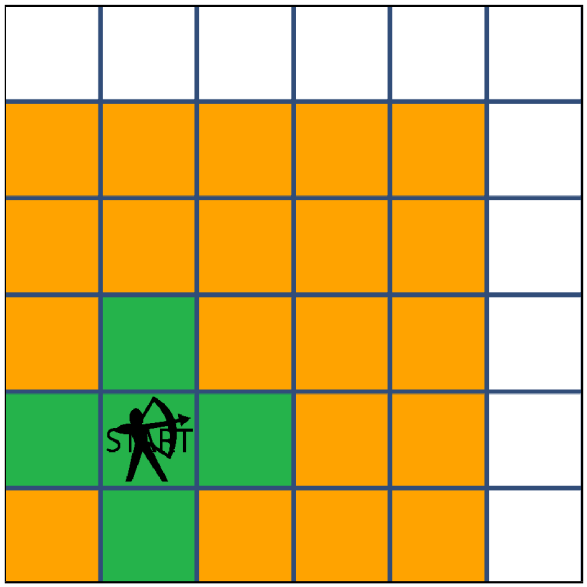
Interface between Prolog & Game Engine

Connecting Prolog to Unity seems like a recurring endeavour:

- **SwiPlCs** – "official" SWI-Prolog interface, *last updated 2013*
  `github.com/SWI-Prolog/contrib-swiplcs`
- **UnityProlog** – Ian Horswill, *last updated 2017*
  `github.com/ianhorswill/UnityProlog`
- **BackTraQ** – *"Prolog like", last updated 2019*
  `github.com/FacticiusVir/BacktraQ`
- **Yield Prolog** – *"Prolog like", last updated 2019*
  `sourceforge.net/projects/yieldprolog`
- **CSharpProlog** – J.Pool & J.Sakamoto, *last updated 2020*
  `github.com/jsakamoto/CSharpProlog`
- **Pengines.Client** – *F# alternative, last updated 2021(!)*
  `github.com/ninjarobot/Pengines.Client`

No entirely satisfying solution... so we build our own interface:
`github.com/sylvainlapeyrade/mqi_csharp`

Prototype: Wumpus World[2]



(a) View of the Agent

(b) View of the world

Principle of Wumpus World

An agent explores a cave, finds the gold and leaves without dying.

[2] Example extended from Russell and Norvig [2021], and Warren [1999]

Small Example of a Wumpus World Game Situation

Listing 6: Human agent knowledge of the world

```
visited(1, 1). visited(1, 2). visited(2, 1).
visited(2, 2). stench(2, 2).

safe(X, Y):- tnot(monster(X, Y)).
safe(X, Y):- tnot(pit(X, Y)).
safe(X, Y):- visited(X, Y).
safe(X, Y):- neighbor(X, Y, X2, Y2),
             visited(X2, Y2),
             no_breeze(X2, Y2),
             no_stench(X2, Y2).
```

Listing 7: Prolog query for monster given new knowledge

```
?- monster(3, 2).
true.
```

What's next?

Current work

- Improve the prototype to enable even more complex behaviour, espcially in MAS context
- Better formalise the creation and interaction with ontologies

What's next?

Current work

- Improve the prototype to enable even more complex behaviour, espcially in MAS context
- Better formalise the creation and interaction with ontologies

Future work

- Integrating and testing AI in Wako Factory's commercial game
- Making Logic Programming easier to use for game designers outside academia

What's next?

Current work

- Improve the prototype to enable even more complex behaviour, espcially in MAS context
- Better formalise the creation and interaction with ontologies

Future work

- Integrating and testing AI in Wako Factory's commercial game
- Making Logic Programming easier to use for game designers outside academia

Thank You!

sylvain.lapeyrade@uca.fr

References I

Denise Angilica, Giovambattista Ianni, and Francesco Pacenza. Declarative ai design in unity using answer set programming. In *2022 IEEE Conference on Games (CoG)*, pages 417–424, 2022. doi: 10.1109/CoG51982.2022.9893603.

Andreas Brännström and Juan Carlos Nieves. UnityIIS: Interactive Intelligent Systems in Unity. 12 2021. URL `https://git.io/JMpzr`.

Francesco Calimeri, Stefano Germano, Giovambattista Ianni, Francesco Pacenza, Simona Perri, and Jessica Zangari. Integrating rule-based ai tools into mainstream game development. In *RuleML+RR*, 2018.

Richard Evans and Emily Short. Versu, 2014. URL `https://versu.com`.

References II

Andrea Gatti and Viviana Mascardi. Towards VEsNA, a framework for managing virtual environments via natural language agents. *Electronic Proceedings in Theoretical Computer Science*, 362: 65–80, jul 2022. doi: 10.4204/eptcs.362.8. URL `https://doi.org/10.4204%2Feptcs.362.8`.

Michael Gelfond and Vladimir Lifschitz. The stable model semantics for logic programming. pages 1070–1080. MIT Press, 1988.

Ian Horswill. Mkultra (demo). *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 11(1):223–225, 2015. URL `https://ojs.aaai.org/index.php/AIIDE/article/view/12776`.

Ian Millington. *AI for games*. Taylor & Francis, a CRC title, Boca Raton, third edition edition, 2019. ISBN 978-1-138-48397-2.

References III

Stuart J. Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Pearson series in artificial intelligence. Pearson, Hoboken, fourth edition edition, 2021. ISBN 978-0-13-461099-3.

Andrey Simonov, Aleksandr S. Zagarskikh, and Victor Fedorov. Applying behavior characteristics to decision-making process to create believable game ai. *Procedia Computer Science*, 2019.

Allen Van Gelder, Kenneth A. Ross, and John S. Schlipf. The well-founded semantics for general logic programs. *Journal of the ACM*, 38(3):619–649, July 1991. ISSN 0004-5411, 1557-735X. doi: 10.1145/116825.116838. URL `https://dl.acm.org/doi/10.1145/116825.116838`.

David S. Warren. Programming in tabled prolog, 1999.

References IV

Georgios N. Yannakakis and Julian Togelius. *Artificial Intelligence and Games*. Springer International Publishing : Imprint: Springer, Cham, 1st ed. 2018 edition, 2018. ISBN 978-3-319-63519-4. doi: 10.1007/978-3-319-63519-4.