Multi-Agent Programming

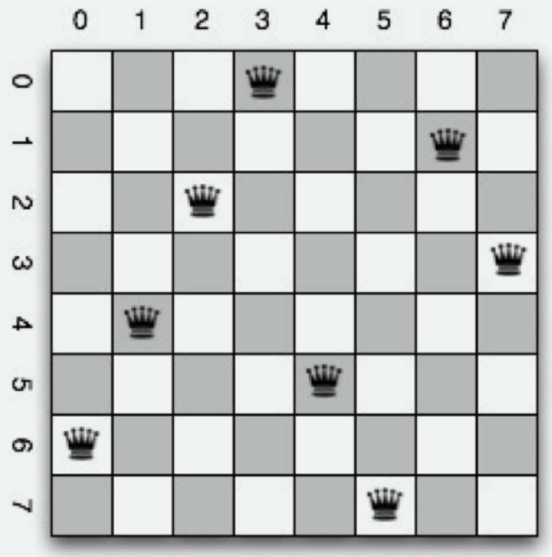
Chinmay Phutela (13103467), Shubham Kumar Jain (13103481)
Bachelor of Technology,
Computer Science Engineering,
Jaypee Institute of Information Technology, Noida

Background

An agent is anything that can perceive its environment through sensors and acts upon that environment through effectors. Multi-agent systems are specified in terms of individual agents and the environments with which they interact. At the individual agent level, it provides programming constructs to implement cognitive agents based on the BDI architecture. In particular, it provides programming constructs to implement an agent's beliefs, goals, plans, actions, events, and a set of rules through which the agent can decide which actions to perform.

Agent-oriented programming (AOP) is a programming paradigm where the construction of the software is centered on the concept of software agents. In contrast to object-oriented programming which has objects at its core, AOP has externally specified agents at its core.

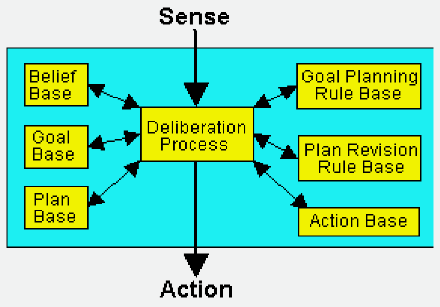
The N-Queen Problem extends the famous chess problem where we have to place N queens on a NxN chess board such that no two queens can cross each other. For example, following is a solution for 8 Queen Problem.



What is 2APL?

2APL is a BDI-based agent-oriented programming language that supports an effective integration of multi-agent system programming constructs such as belief and goals and imperative style programming constructs such as events and plans.

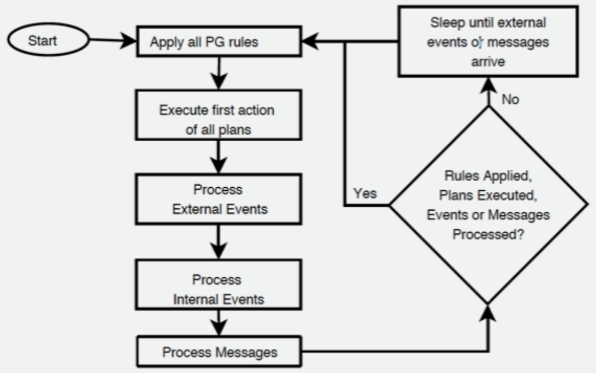
Agents hosted on different 2APL platforms can communicate with each other. 2APL interpreter is built on the JADE platform.



2APL (A Practical Agent Programming Language) is a modular BDI based programming language that supports the development of multi-agent systems. 2APL provides a rich set of programming constructs allowing direct implementation of concepts such as beliefs, declarative goals, actions, plans, events, and reasoning rules. 2APL provides programming constructs to specify both multi-agent systems and individual agents.

How it works?

We have a 2APL file named ABT.2apl and BF.2apl includes our agents beliefs goals and plans and agents here are the 4 queens the individual queen is a single agent who has its own belief, goals, intentions and these together working on an environment named env.jar file. In 2APL file our design is Queens pass along control to solve the puzzle. First off, the second queen q adjusts herself to prevent violating the rules with the queen on her left (the first queen), after which q tells the one on her right to adjust. If one of the queens q' cannot find a legal spot to move to, then q' can tell the one on the left to move to another legal spot. However, if that queen q'' cannot do that, q'' will tell the one left of q'' to move etc. Until one can move legally and then reports back to the right. So for making this agents have different beliefs plans and some rules.


Start
Apply all PG rules
Execute first action of all plans
Process External Events
Process Internal Events
Process Messages
Rules Applied, Plans Executed, Events or Messages Processed?
Yes
No
Sleep until external events or messages arrive

Applications

Multi-agent systems are applied in the real world to graphical applications such as computer games. Agent systems have been used in films. They are also used for coordinated defence systems. Other applications include transportation, logistics, graphics, GIS as well as in many other fields. It is widely being advocated for use in networking and mobile technologies, to achieve automatic and dynamic load balancing, high scalability, and self-healing networks.

Conclusion

N queen problem using BDI agent programming is implemented in 2APL framework and is working properly. All the agents which is assigned to each queen are communicating and functioning in the way they are supposed to work.

Acknowledgements

We would like to place on record my deep sense of gratitude to Mr. Kishore Kumar Yarramshetty, Jaypee Institute of Information Technology, Noida for his generous guidance, help and useful suggestions. His continuous encouragement and supervision throughout the course of present work is the main reason behind the success of the project.

References

[1] Mehdi Dastani - 2APL manual: a pratical agent programming language, Proceedings of theInternational Joint Conference on Artificial Intelligence, 2011, pp. 34-45.

[2] Peter Novák, Jurgen Dix - Modular BDI Architecture, Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems, 2006

[3] Vikas Thada, Shivali Dhaka - Performance Analysis of N-Queen Problem using Backtracking and Genetic Algorithm Techniques, International Journal of Computer Applications (0975 – 8887), Volume 102– No.7, September 2014

[4] Vishal kesri, Vaibhav kesri, Prashant Kumar Patnaik - A Unique solution for N queen problem, International Journal of Computer Applications (0975 – 8887), Volume 43– No.12, April 2012