Modularization in Belief Desire Intention-agent programming and artifact-based environments

BACKGROUND

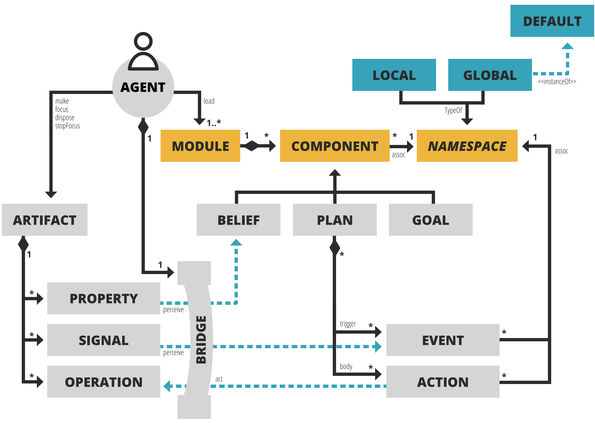
In this article we propose an extension for the Agents and Artifacts meta-model to enable modularization.

The issues that are solved by our proposal are:

- **Facilitating teamwork when developing agents**
- **Lowering the cost of maintaining and extending the code**
- **Dynamically updating agents with new functionality**

OUR MODEL

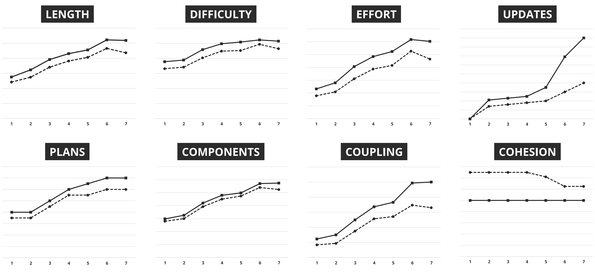
We adopt the Belief-Desire-Intention (BDI) model of agency to represent independent and reusable units of code by means of modules. **The key idea behind our proposal is to take advantage of the syntactic notion of *namespace***, i.e., a unique symbol identifier to organize a set of programming elements. On this basis, agents can decide in BDI terms which beliefs, goals, events, percepts and actions will be independently handled by a particular module.



The A&A meta-model extended for supporting modularization.

EVALUATION

The practical feasibility of this approach is demonstrated by developing an auction scenario, where source code enhances scores of coupling, cohesion and complexity metrics, when compared against a non-modular version of the scenario. Our solution allows developers to address the name-collision issue, provides an interface for modules that follows the information hiding principle, and promotes software engineering principles related to modularization such as reusability, extensibility and maintainability.



Scores of an auction MAS's according to the Halstead metric for complexity (Length, Difficulty, and Effort); where X-axis denotes the version of the MAS from the initial implementation (1) to that resulting of performing all six extensions (7) (c.f. Table 5), and Y-axis corresponds to the score. The updates chart summarizes the block additions and deletions of source code that had to be performed to implement each extension. Plans and Components charts stand for the total plans and components (i.e., beliefs, goals, events, etc.), respectively. The coupling chart shows the sum of coupling scores of all modules in the MAS. The cohesion chart presents the average cohesion per module, the maximum value for this score is 1, and higher values mean less cohesive modules.

CONCLUSION

Differently from others, **our solution allows developers to encapsulate environment components into modules as it remains independent from a particular BDI-Agent-Oriented Programming language.**

PeerJ

Modularization in Belief Desire Intention-agent programming and artifact-based environments
PeerJ Computer Science 8:e1162
DOI: 10.7717/peerj-cs.1162
http://peerj.com/article/cs-1162/

Image credits:
Icons from The Noun Project / Prettycons