Advanced intention scheduler for BDI systems

Integrating Monte Carlo Tree Search into FRAg system

Author: Ondřej Misař
Supervisor: doc. Ing. František Zbořil, Ph.D.
Brno university of technology
Faculty of information technology
Department of intelligent systems

Introduction

- BDI systems face challenges in prioritizing intentions efficiently, especially in dynamic environments.
- Traditional schedulers like Round Robin lack foresight, leading to suboptimal resource allocation.
- Introduction of Monte Carlo Tree Search scheduling for AgentSpeak(L) systems like FRAg.

Objectives

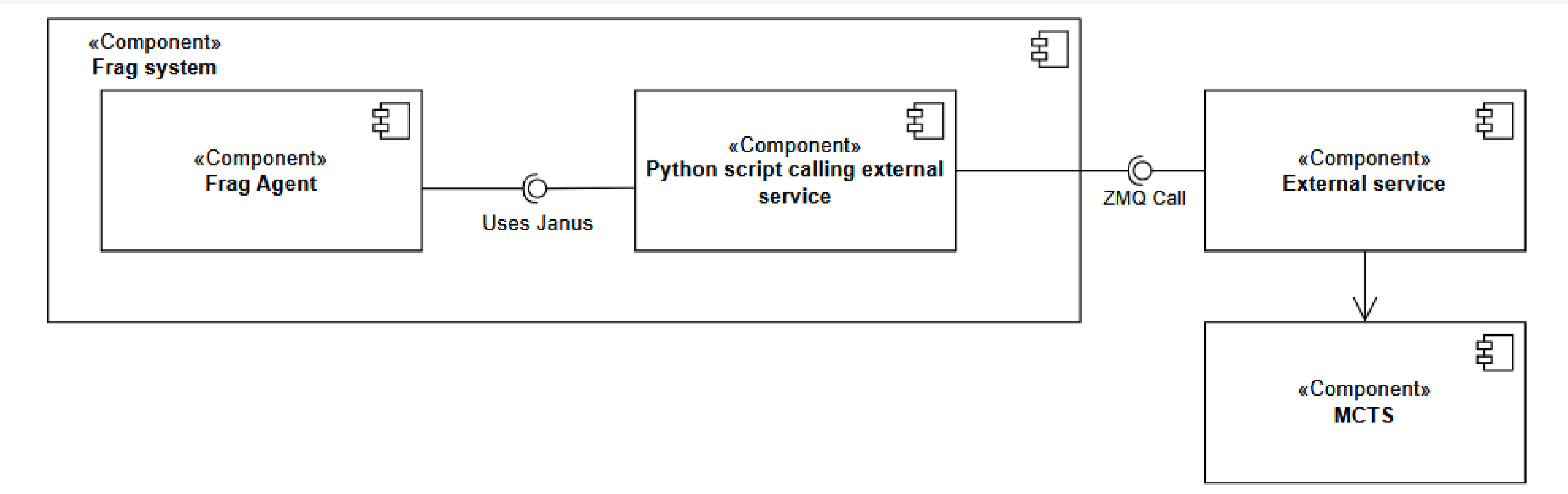
- Study current approaches for intention scheduling in BDI systems.
- Extending methods for BDI systems based on the language of first order predicate logic.
- Integration of new schedulers into FRAg system.
- Comparison of new methods against existing schedulers in FRAg.

Monte Carlo Tree Search

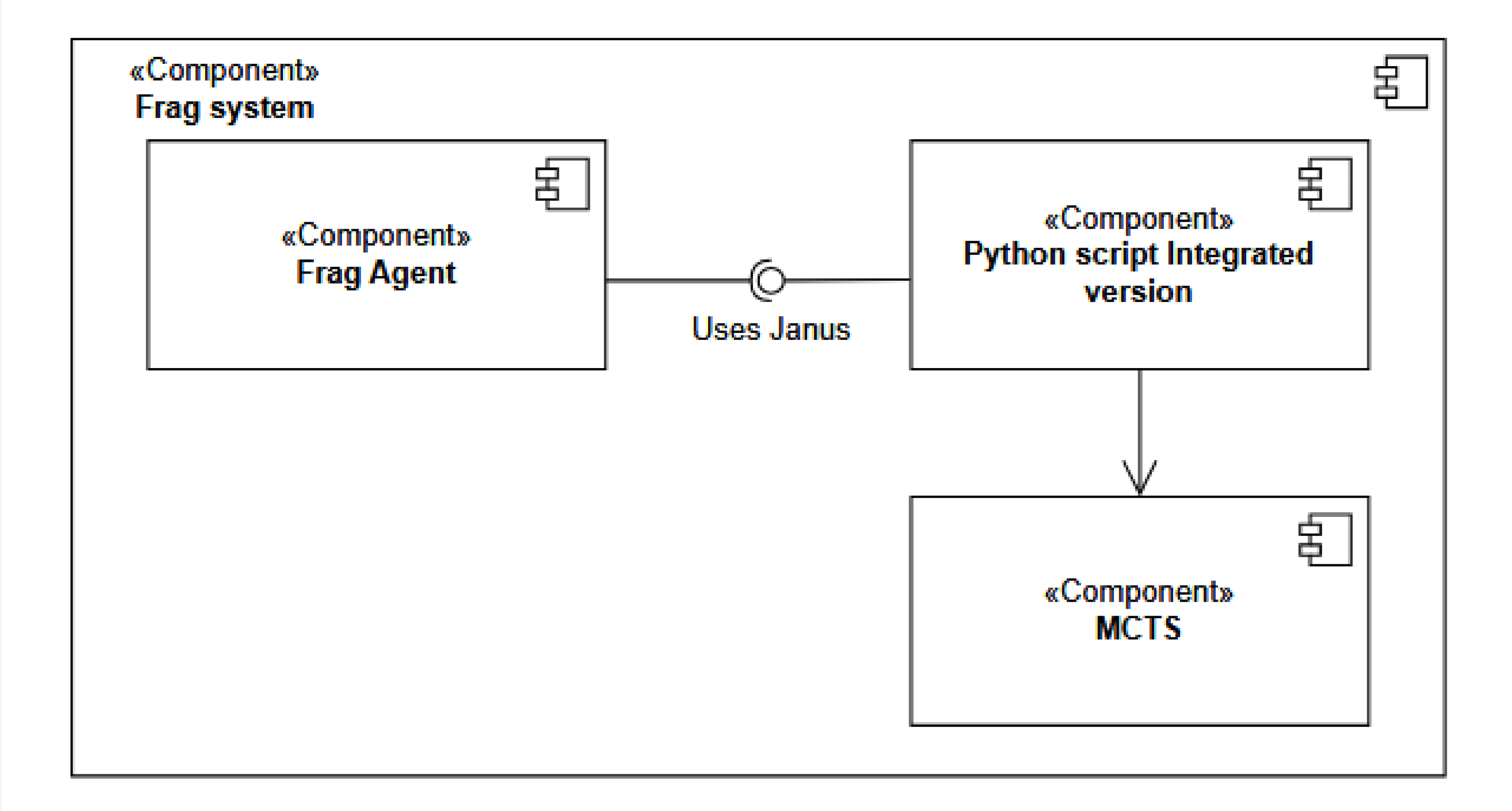
- Base version of MCTS uses action level interleaving, by scheduling actions between intention, in order to maximize the number of top-level goals and to minimize variance between completing intention.
- Online learning MCTS is currently better performing then a base version by incorporating online learning in simulation phase to introduce the use of simulation history for next simulations.

Integration with FRAg

- External architecture of integrating MCTS implemented in python with FRAg system implemented in prolog.



- Integrated architecture of MCTS implemented in python with FRAg system implemented in prolog.



Results

- Speed comparison of different MCTS methods for task maze program for 5 agent steps.

Configuration	Prolog MCTS	Integrated arch.	External arch. sequential	External arch. parallel
Alpha=1, Beta=1 (Avg)	4.4760s	9.5292s	8.5173s	12.9548s
Alpha=1, Beta=1 (Min)	3.7297s	7.6638s	7.8411s	11.5964s
Alpha=5, Beta=5 (Avg)	98.1758s	34.8420s	36.6804s	25.3186s
Alpha=5, Beta=5 (Min)	92.9426s	33.2225s	34.8548s	24.3520s
Alpha=10, Beta=10 (Avg)	369.1806s	90.5079s	92.7427s	43.1175s
Alpha=10, Beta=10 (Min)	340.9618s	83.5594s	89.9693s	40.2057s

- Results for garden example where 9 zone moistures are plotted over 50 agent steps, goal is to keep zone moisture as high as possible.

Best run of Round Robin reasoning method.

Best run of online learning MCTS reasoning method.

Conclusion

- Implementation of MCTS in python is able to schedule next intention faster then MCTS implementation in prolog.
- The overhead of using the external architecture is negligible while parallel variant of MCTS is able to be faster then sequential variant.
- Online learning variant is able to achieve better results then other reasoning methods like Round robin.

Future work

- Faster intention scheduling by using different languages such as c++ or c.
- Integration of more variants of MCTS method for intention scheduling.
- Use of communication between prolog and different programming languages for more parts of FRAg system.