Incorporating Learning in BDI Agents

Lin Padgham: RMIT University, Melbourne, Australia

lin.padgham@rmit.edu.au

Collaborators: **Dhirendra Singh**
Sebastian Sardina
Stéphane Airiau

RMIT UNIVERSITY

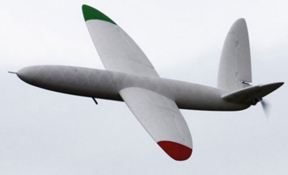
Outline

1. Belief Desire Intention (BDI) Agents
2. Learning Approach and Issues
3. Confidence and Experience
4. Case Study Evaluations
5. Planning and Learning - future work
6. Conclusion

Outline

1. Belief Desire Intention (BDI) Agents
2. Learning Approach and Issues
3. Confidence and Experience
4. Case Study Evaluations
5. Planning and Learning - future work
6. Conclusion

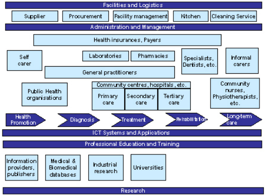
Agent Systems Useful in Many Applications



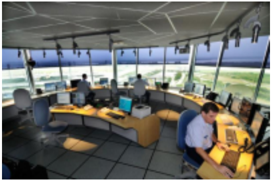
Unmanned (Aerial) Vehicles

Trading Agents

Logistics



E-Health



Air Traffic Control

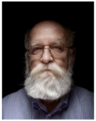
Agents have been used in many successful applications in complex environments.

Agent Oriented Programming

- The increasing complexity of computer programs benefits from more expressive abstractions. Microcode, Assembly, Structured, Object-Oriented, Agent-Oriented.
- Agent-Oriented programming provides abstraction at the level of mental attitudes to explain the operation of a system. Beliefs, Desires, Intentions.
- The modularity of plans makes it easy to develop complexity incrementally.
- The goal oriented approach makes it suitable for use in dynamic environments.
- Agent programs are fast to develop. A 2006 study showed:
  - Average gain in development time of 350%.
  - Gain compared to java programming 500%.

Belief Desire Intention Model of Agency

- BDI is a framework for describing the behaviour of rational agents.
- Based on work in the philosophy of mind:



Dennett

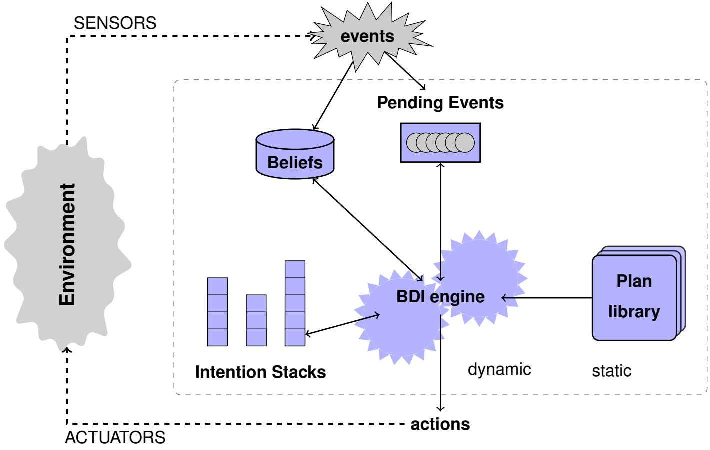
Intentional systems: "[..] whose behavior can be predicted by the method of attributing belief, desires and rational acumen."

Bratman

Practical reasoning: "[..] a matter of weighing conflicting considerations [..] provided by what the agent desires [and] believes."

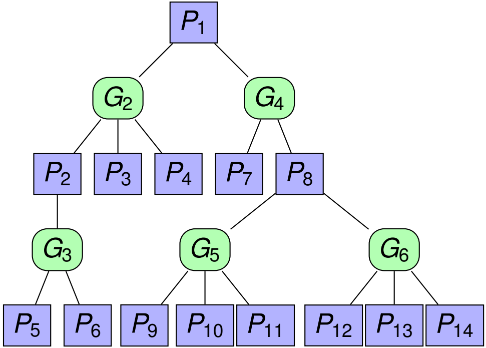
- Human practical reasoning consists of two activities:
  - Deliberation: deciding what to do i.e., form intentions.
  - Means-ends Reasoning: deciding how to do it i.e., form plans.

Belief-Desire-Intention (BDI) Agent Architecture



A plan is a *programmed* recipe for achieving a goal in some situation.
A BDI execution engine selects from a plan library, based on the situation.

Goal-Plan Tree: Decomposition and Selection



- A plan typically has a number of (sub)goal steps.
- Each sub-goal generates an (internal) event which has some relevant plans.
- So the plan library can be seen as a set of goal-plan trees.
- At each goal node a plan must be selected (OR).
- At each plan node the goals must be accomplished (AND).

Adaptation in BDI Agents

- BDI programs are an efficient way to provide many different ways to achieve a top level goal. (A goal plan tree with 2 plans/goal, 4 sub-goals/plan and 3 levels can give over 2 million ways to achieve the top goal.)
- If a plan fails (often because something changed), another plan can be tried. So BDI programs are very robust.
- But the specification of when a plan will work - its context condition is critical. If the environment changes so that a plan which worked previously in a certain sitiuation, no longer does so, the agent is UNABLE TO LEARN this.
  E.g. If a plan for taking the train has a context condition that you have at least $5, but the price of a train ticket goes up, the agent will not learn that the situation in which this plan succeeds has changed.

What We Would Like...

- Agents that start with the best knowledge we can give them, but then learn from experience.
- Agents that learn on-line, after they are deployed and functioning.
- Agents that learn continuously to adapt to changing characteristics of the environment. (Ability to “unlearn” previously learned rules.)

Outline

1. Belief Desire Intention (BDI) Agents
2. Learning Approach and Issues
3. Confidence and Experience
4. Case Study Evaluations
5. Planning and Learning - future work
6. Conclusion

Machine Learning

What do we mean by learning?

- A computer program is said to learn from experience E with respect to some class of tasks T (e.g., chess games) and performance measure P (e.g., winning), if its performance at tasks in T, as measured by P, improves with experience E (e.g., obtained by playing).
- In our BDI program the tasks T are the selection of appropriate plans for various situations, and P is success of the goal to be achieved.

For this we use Decision Tree learning:

- allows disjunction of conjunctive terms similar to context formulas
- robust against noisy training data
- well-developed technology
- may be converted to if-then rules for validation if needed

Decision Tree Learning

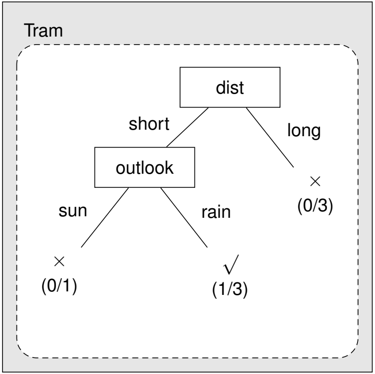
Example plan: Tram

dist	outlook	result
long	sun	$\times$
short	rain	$\surd$
short	sun	$\times$
short	rain	$\surd$
long	rain	$\times$
short	rain	$\times$
long	sun	$\times$
...	...	...

Decision Tree Learning

Example plan: Tram

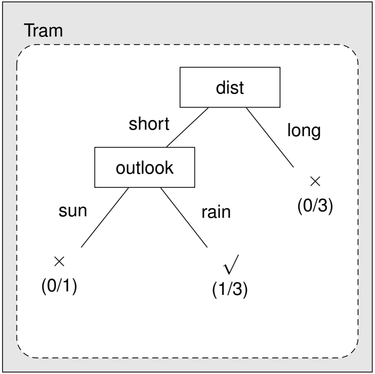
dist	outlook	result
long	sun	×
short	rain	√
short	sun	×
short	rain	√
long	rain	×
short	rain	×
long	sun	×
...	...	...



Decision Tree Learning

Example plan: Tram

dist	outlook	result
long	sun	$\times$
short	rain	$\surd$
short	sun	$\times$
short	rain	$\surd$
long	rain	$\times$
short	rain	$\times$
long	sun	$\times$
...	...	...



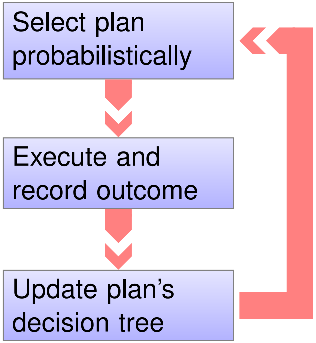
Prediction: Will succeed when dist $=$ short $\wedge$ outlook $=$ rain.
Likelihood of success is $\left[1 - \frac{1}{3}\right]$ or 67%.

The Learning Framework

Augment one decision tree per plan. State representation includes world features, event parameters, and context variables.

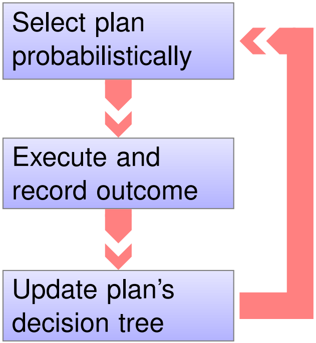
The Learning Framework

Augment one decision tree per plan. State representation includes world features, event parameters, and context variables.



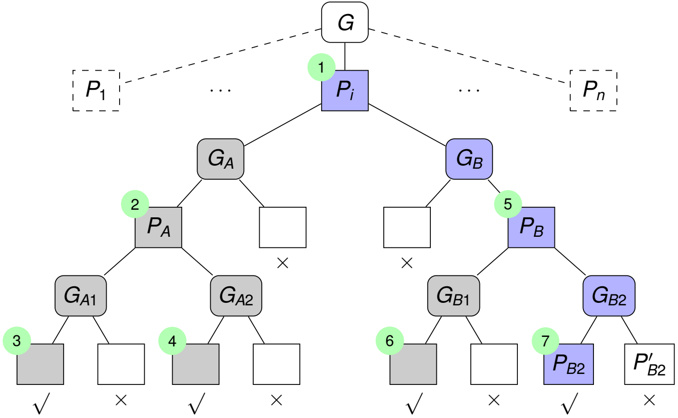
The Learning Framework

Augment one decision tree per plan. State representation includes world features, event parameters, and context variables.



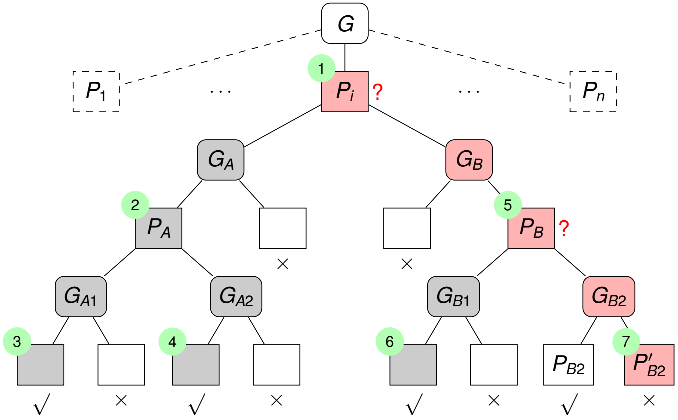
1. For every plan whose context() holds, calculate a selection weight based on perceived likelihood of success.
2. Select a plan probabilistically using selection weights.
3. Execute the plan (hierarchy) and record the outcome(s).
4. Update the plan’s decision tree.
5. Repeat.

Learning From Plan Choices



Execution trace for successful resolution of goal G given world state w.
Success means that all 7 correct choices were made.

Learning From Plan Choices

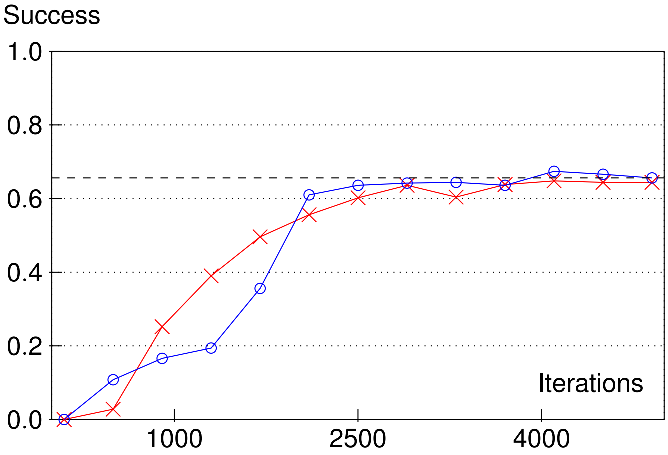


Possible execution trace where goal G is not resolved for w.
Should non-leaf plans consider this failure meaningful?

Does This Noisy Data Matter?

- Experimented with filtering training data, so we used it for a given plan, only when we were convinced that choices below were well informed. (We call this Filtered Training Data FTD).
- Compared FTD results with an approach using all training data (Unfiltered Training Data: UTD), using different kinds of program structures: many solutions vs few, mix of deep and broad sub-hierarchies, etc.
- Success always recorded for both approaches.
- On some structures FTD learnt much faster, on some UTD learnt much faster.
- However, if we introduce a probability threshold below which we don't try a plan at all, then UTD may fail to ever learn.

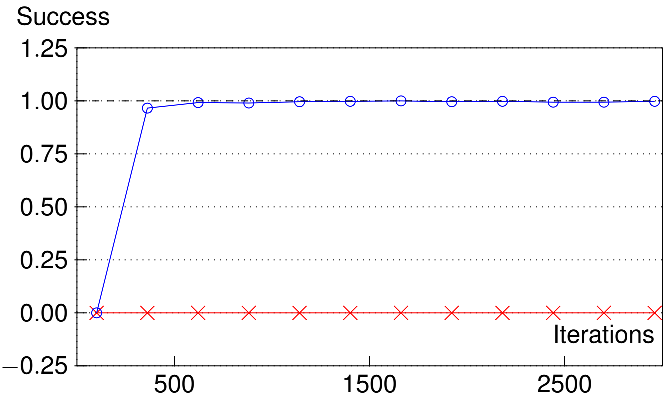
Results: Some Structures Equivalent Performance



Performance of UTD (crosses) vs. FTD (circles).
Dashed line shows optimal performance.

Results: With Applicability Threshold

Plan execution is generally not cost-free, so agent may fail a goal without even trying if all available plans seem unlikely to succeed.



Performance of UTD (crosses) vs. FTD (circles).
Dashed line shows optimal performance.

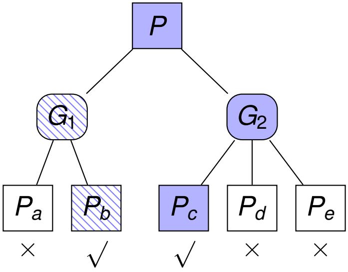
Outline

1. Belief Desire Intention (BDI) Agents
2. Learning Approach and Issues
3. Confidence and Experience
4. Case Study Evaluations
5. Planning and Learning - future work
6. Conclusion

A Dynamic Confidence Measure

- Instead of filtering training data, we can have a measure of how confident we are in the learning so far.
- If we are very confident we allow the DT estimates of success to strongly influence the probability of selecting specific plans, giving high exploitation. If we have low confidence, the DT estimates of success have only a small influence on the probability of selecting specific plans (and therefore there is more exploration.
- This also allows us to deal with other issues such as over-generalisation when we don't yet have sufficient experience of different world states, and cases where we should decrease our confidence if plans that previously succeeded start to fail.

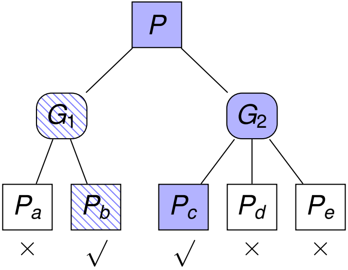
A Dynamic Confidence Measure



We build confidence from observed performance of a plan using:

- how well-informed were the recent decisions, or stability-based measure
- how well we know the worlds we are witnessing, or world-based measure
- an averaging window n and preference bias α

A Dynamic Confidence Measure

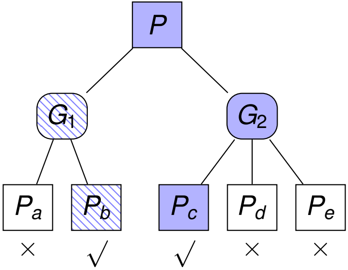


We build confidence from observed performance of a plan using:

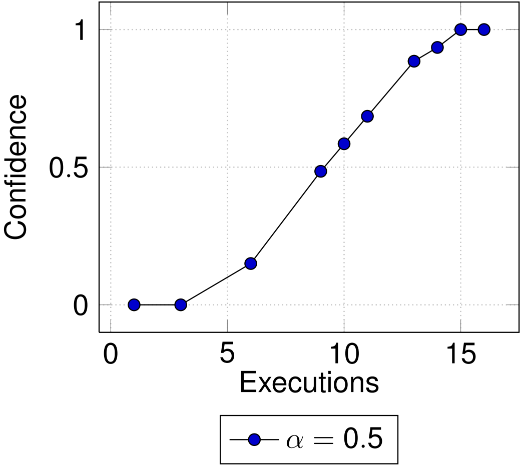
- how well-informed were the recent decisions, or stability-based measure
- how well we know the worlds we are witnessing, or world-based measure
- an averaging window *n* and preference bias α

Plan selection weight, that dictates *exploration*, is then calculated using the predicted likelihood of success and the dynamic confidence measure.

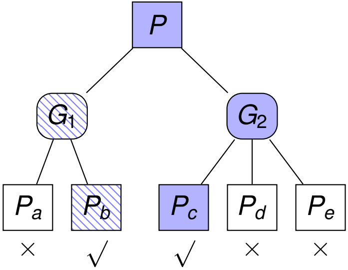
Example: Dynamic Confidence Measure



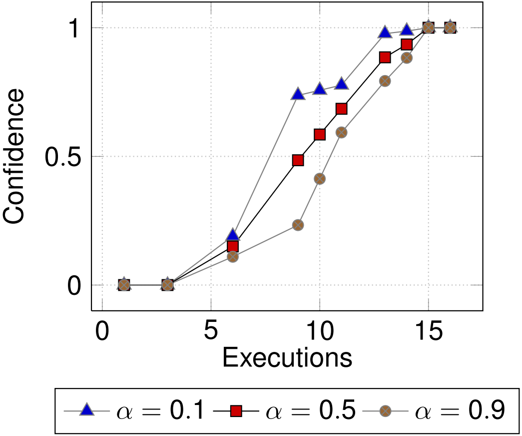
Solution found at E=10 and full confidence at E=15.



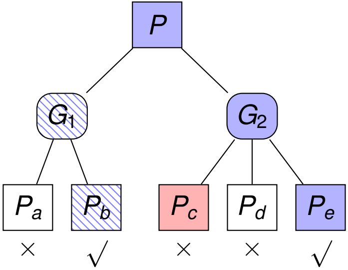
Example: Dynamic Confidence Measure



$\alpha = 0.0$: Only world-based
$\alpha = 1.0$: Only stability-based
$n = 5$: Averaging window



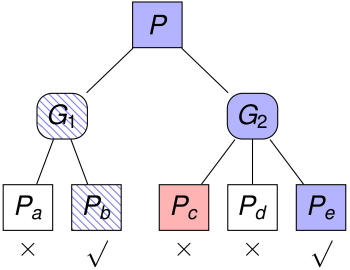
Example: Dynamic Confidence Measure



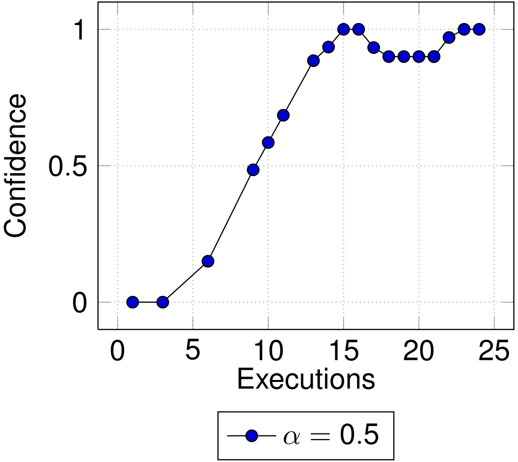
What if the environment changes after we have learnt the solution?

Say after execution 15, plan P_c no longer works for resolving goal G_2, but plan P_e does.

Example: Dynamic Confidence Measure



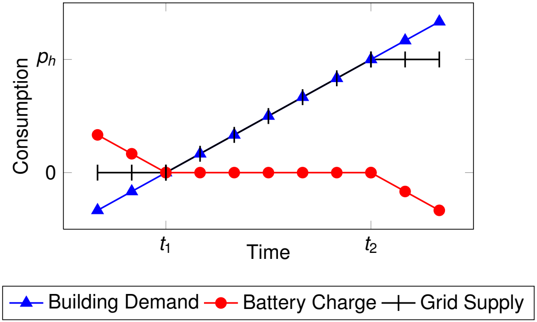
After E=15, P_c starts to fail. The confidence drops, promoting new exploration and re-learning.



Outline

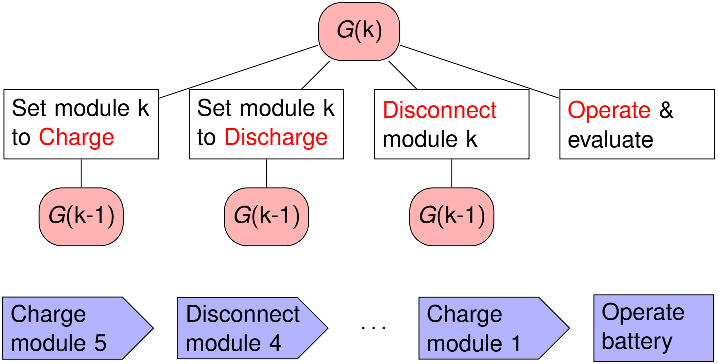
1. Belief Desire Intention (BDI) Agents
2. Learning Approach and Issues
3. Confidence and Experience
4. Case Study Evaluations
5. Planning and Learning - future work
6. Conclusion

A Battery Storage Application



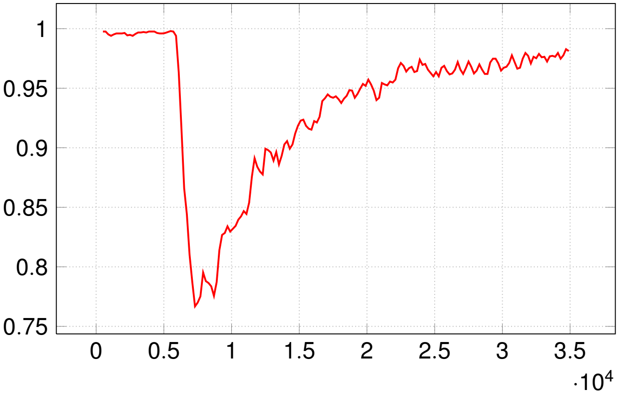
Given net building demand, calculate an appropriate battery response in order to constrain grid power consumption.

Design: A Battery Storage Application



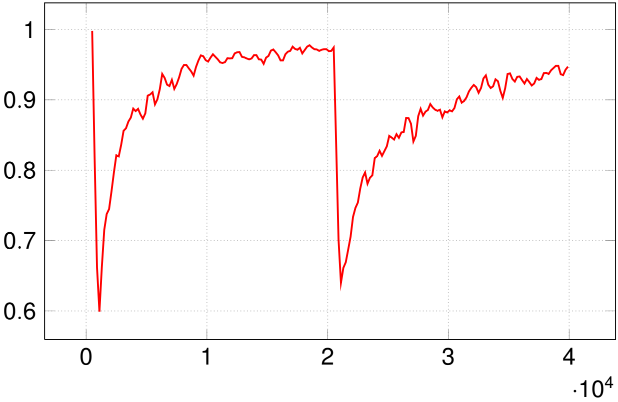
Aim: Learn appropriate plan selection to achieve a desired battery response rate, given the current battery state. Full state space for a battery with five modules is $\approx$13 million.

Experiment: Capacity Deterioration



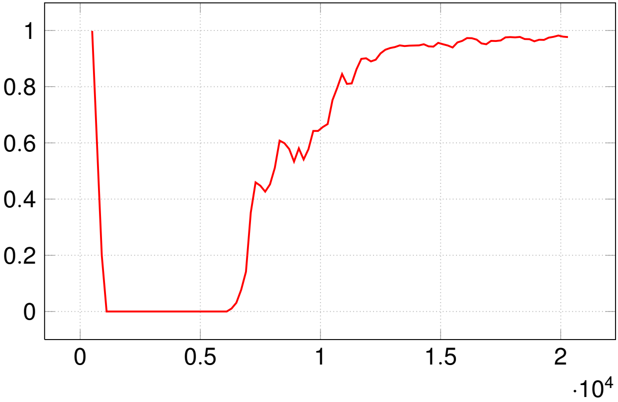
Battery performance during permanent deterioration in module capacities.

Experiment: Partial Failure with Restoration



Battery performance during various module malfunctions and restorations.

Experiment: Complete Failure with Restoration



Battery performance during complete failure followed by full restoration.

Outline

1. Belief Desire Intention (BDI) Agents
2. Learning Approach and Issues
3. Confidence and Experience
4. Case Study Evaluations
5. Planning and Learning - future work
6. Conclusion

Learning More Than Plan Selection?

- Learning the rules of plan selection is very useful - but ideally we would also like to learn new plans if needed.
- This is related to planning and some work we have done there adding planning to BDI systems.
- Similar to the learning work, we want to build on the BDI program, not start with an empty slate.
- Approach we used was to combine existing “chunks” of BDI program to achieve desired goal, if no existing applicable plan.
- The plan found could potentially be added to library, and its context condition learned.
- The approach retains more of the BDI structure (and encoded expertise) than just planning with actions.

Further use of Planning with Learning

- When there are interactions between sub-goals within a plan, the current approach cannot learn well.
- This is a fundamental aspect of BDI representation where goals/subgoals are modular independent chunks, reusable in multiple contexts.
  E.g. Booking a flight and a hotel in the context of a goal to attend a conference with limited budget, have interactions that do not apply to the sub-goals independently.
- What we will learn is that the top level goal (doing the conference arrangements) will always succeed in some situations (when there is plenty of money), will always fail in others (when there is no solution with given budget) and will succeed with some probability in between - depending on how the plans are selected.
- HTN planning could be used to find a successful combination of selections for the in-between space.

Outline

1. Belief Desire Intention (BDI) Agents
2. Learning Approach and Issues
3. Confidence and Experience
4. Case Study Evaluations
5. Planning and Learning - future work
6. Conclusion

Conclusions

- Successful mechanism to learn refinements to context conditions as an environment changes.
- If the environment changes with respect to what plans work in a situation, or with respect to what situations are seen, the confidence measure will adapt.
- One limitation is that context conditions cannot be weakened by learning.
- A further limitation is inability to learn about interactions. This can possibly be addressed using HTN planning
- Future work could look at planning to find new ways to achieve a goal when there are no good plans, and then learning to discover the context condition for the new plan.

Publications

- Dhirendra Singh, Sebastian Sardina, Lin Padgham, and Geoff James. Integrating learning into a BDI agent for environments with changing dynamics. IJCAI 2011, p 2525-2530.
- Dhirendra Singh, Sebastian Sardina, and Lin Padgham. Extending BDI plan selection to incorporate learning from experience. Journal of Robotics and Autonomous Systems, 58:1067-1075, 2010.
- Dhirendra Singh, Sebastian Sardina, Lin Padgham, and Stéphane Airiau. Learning context conditions for BDI plan selection. AAMAS 2010 pp. 325-332.
- Stéphane Airiau, Lin Padgham, Sebastian Sardina, and Sandip Sen. Enhancing adaptation in BDI agents using learning techniques. International Journal of Agent Technologies and Systems (IJATS), 1(2):1-18, January 2009.
- Lavindra P. de Silva, Sebastian Sardina, and Lin Padgham. First principles planning in BDI systems. AAMAS 2009, volume 2, pages 1001-1008.