The Ents

Students’ Software Project

Ondřej Bojar, Cyril Brom, Milan Hladík, Vojtěch Toman, Mikuláš Vejlúpek, David Voňka
Goal: World Simulator

- Key aspects of the simulation should be similar to the world where *people* live:
  - People are beings with independent thoughts.
  - People can observe their closest neighbourhood.
  - People can change their neighbourhood in a simple way.
  - People do not do random things, they have specific goals.
  - People talk to one another.
What *not to Simulate*

- No faithful simulation of physical properties of our world.
- No evolution of the world.
- No unreliability of human senses.
- No learning skills, adaptability.
- Basic limitations:
  - Discrete space (rooms+tiles+containment) and objects.
  - Discrete time flow, turn based.
Liberté, Egalité, Fraternité (LEF)

- Human users take part in the world simulation.
- Strict equality of rights for humans and ents – LEF.
- LEF was implemented.
- LEF consequences:
  - Platform for Turing test.
  - Complexity of human world made obvious.
Project Architecture

- Independent processes on Unix.
- Socket communication.
- Different programming languages used.
Common Dialect for All the Components: The Handles

- Universal numbering of all the ents, objects, properties, numbers etc.
- Four-byte constants hard-wired in the source code (by means of *symbolic constants* and preprocessing):
  - HOnchairH, HPcolour-redH, HEn GardnerH, ...
- Used for *atomic instructions* as well:
  - Full description of simple actions of ents.
  - Exactly one atomic instruction of every ent in every turn. Can succeed or fail.
  - Approx. 50 types of instructions: HXwalkH, HXpick_upH, HXsayH, ...
Behaviour of Ents

- Short and long term memory.
- Scripting language E.
  - Hierarchical description of actions (goals); syntax similar to Prolog.
  - Variants of backtracking.
  - Interrupts and handlers. Priorities.
  - Evaluation functions.
  - Modularization, libraries.
- Simple understanding of a natural language (Czech).
- Debugger.
The Memory of Ents

- Main usage:
  - Looking up particular objects with requested properties.
  - Common “blackboard” for all the modules.

- Short- and long-term memory not distinguished:
  - Structurally less complex.
  - Within one turn, the ent knows “all at once”.
  - Behaviour not driven by attention.
  - Difficulties with formulating algorithms based on “current changes”.
  - Difficulties with natural language generation.
The Linguistic Module

- Translates strings of NL to sequences of handles (internal representation of meaning).
- Simplistic sentence patterns, focus on identifying objects by NL expressions (“concretization”):
  - A blue book, the blue book, it, <this book>.
- A black box for all other modules.
  - Easier to implement.
  - In fact unnatural ⇒
    - Problems of full concretization.
    - Problems of semantic feasibility.
      (Pick up the bed.)
Programming of Ents

- A new language E.
- Syntax similar to Prolog:
  - Hierarchical description of actions (scripts).
  - Emitted atomic instructions succeed or fail.
  - Different variants of backtracking.
  - Utility functions assigned to different variants of a script choose the best variant.
- Modularization, libraries.
  Walking, manipulating objects, “cooperation”, …
- External functions in C/C++/Mercury:
  - Used for path searching etc.
  - More efficient but need recompilation.
Custom scripts to be run immediately under certain circumstances.

After every atomic instruction sent, the interpreter checks for interrupts.

- *Global interrupts* to check for physical needs.
- *Local interrupts* to check if the required conditions still hold.
  
  Is the watering-can still full enough? If not, refill.

After handling the interrupt, the control goes back to the interrupted script.

Attention! Your script wakes up in different conditions!

Different priorities for different interrupts.
Configuration: World Package

- A directory with all the configuration files needed:
  - Rooms and furniture, initial object locations, number of participants (ents/browsers), ...
  - Sentence patterns.
  - All the scripts for ents.
- All the components should be run in the World Package directory.
- The World Package is the “playground” for an AI student or researcher.
The Ents: Summary

- Program components:
  - World Server, Ent, Browser.
- Language E to describe actions of ents.
  - Including a GUI debugger.
- A library of basic actions and scripts:
  - Walking..., a simple scheduler, eating, drinking, toilet, ...
- A sample world package
  - A gardener and a musician.
- Four books (~500 pages) of documentation.