Benchmarking Smart Spaces Through Autonomous Virtual Agents

Mario Caruso, Francesco Leotta, Massimo Mecella and Stavros Vassos

Dipartimento di Ingegneria Informatica, Automatica e Gestionale

In the recent years there has been a growing interest in the design and implementation of smart homes. The evaluation of related techniques requires massive datasets of measurements from deployed sensors in real prototypes.

CHALLENGE: Freely available smart home datasets are not sufficient for comparing different approaches and techniques in a variety of configurations.

SOLUTION: A smart home dataset generation strategy based on a simulated environment populated with virtual autonomous agents, sensors and devices.

### 1. Modeling daily life

The sensor log produced by a smart home can be considered as a snapshot of the inhabitants performing their habits (or activities of daily life) [1].

**Habit**

- A loosely specified sequence of high-level actions aiming at a particular goal, e.g., cleaning the house.
- The way a habit is performed may portray a high degree of variability between different users or even between the same user in different time frames.
- We adopt a declarative approach for the specification of habits, called DECLARE [2].

DECLARE

A declarative modeling language for business processes:
- Focuses on the (minimal) set of rules which must be satisfied in order to correctly carry out a habit.
- Each model (habit) is defined by a set of tasks (h-actions) and by a set of constraints, whose semantics is based on Linear Temporal Logic - LTL.

Each h-action in the habit traces is turned into a planning goal.

- The Planner Log is obtained by planning for each of them. A single h-action generates multiple p-actions.
- The final state of one of a planner execution is used as initial state for the successive one.

### 2. Habit Interleaving

**Input**

Habit trace parameters

The habit trace is composed by a fixed number N of model sets. Each model set is made up by a maximum of maxH habits and maxD distractions.

From each chosen model, a random instance [3] satisfying the constraints is extracted.

- The LTL semantics of DECLARE allow a set of templates to be combined by a simple logical AND.
- In the case of overlapping h-actions between templates, consistency is guaranteed.

### 3. Continuous planning

**Input**

The planning domain introduces predicates and actions related to different components of the virtual environment.

- The position of the character expressed by the charAt predicate. A character can move to either a specific room or device.
- The topology expressed using the adjacentChar predicate.
- The devices available into the environment:
  - Stateful devices have an associated state that can be changed.
  - Stateless devices can be only used.

The initial state of the world assigns the fixed predicate whose value will not change during the planning.

Each h-action in the habit traces is turned into a planning goal.

- The Planner Log is obtained by planning for each of them. A single h-action generates multiple p-actions.
- The final state of one of a planner execution is used as initial state for the successive one.

### 4. Sensing

**Input**

The set of sensors available in the house. Each sensor is in charge of monitoring a specific attached device/room. The type of the sensor denotes its ability to sense a certain kind of event (e.g., movement).

**Output**

P-actions that relate to sensors are automatically translated into sensor measurements while the others are filtered out. The result is the Sensor Log that is the output of the system.

The result sensor log strongly depends on the kind and number of sensors that are part of the environment.

- The performance of algorithms for smart homes can be tested over different sets of sensors.
- Robustness of algorithms can be evaluated introducing errors in sensor measurements.
- Sensors installed into real home can be selected maximizing the performance measure.