The Benefits of Explicit Ontological Knowledge-Bases for Robotic Systems

Zeyn Saigol[1] Minlue Wang[2] Bram Ridder[3] David M Lane[1]

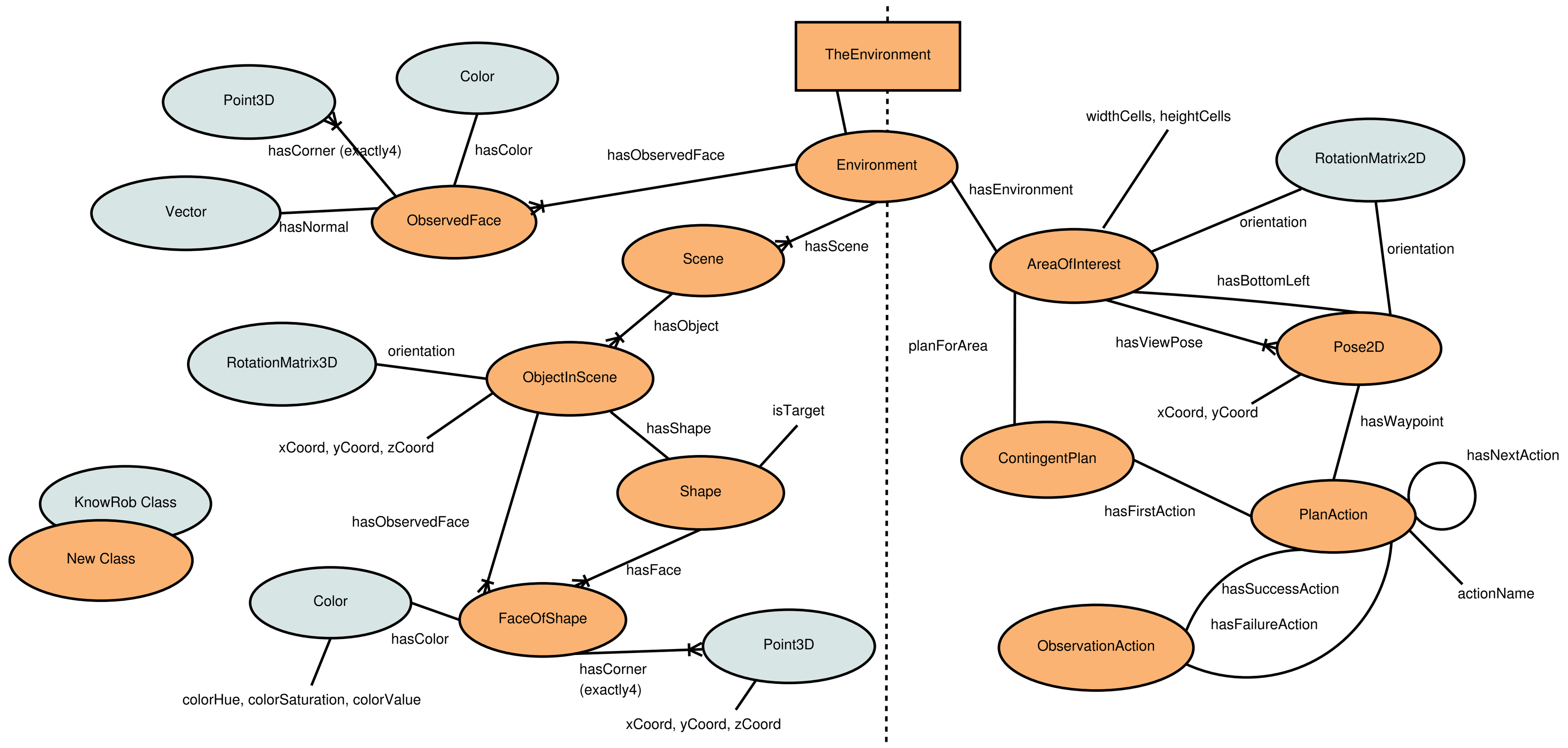
[1]Heriot-Watt, Edinburgh [2]University of Birmingham [3]King's College London

Introduction

With the increasing abilities of robots comes a corresponding increase in the complexity of creating the software enabling these abilities. We present a case study of a sophisticated robotic system that uses an ontology as the central data store for all information processing. We show how this makes for a system that is easier to develop, understand, and modify.

System and Ontology

We consider a mobile robot trying to find an item of known shape in an unmapped area. This work focuses on ontology; the full system and results are described in [2]. The ontology stores a set of possible worlds (`Scenes`), all of which are consistent with observations made so far. Each `Scene` contains `ObjectInScenes`, which have a known `Shape` and a pose derived from sensor data.

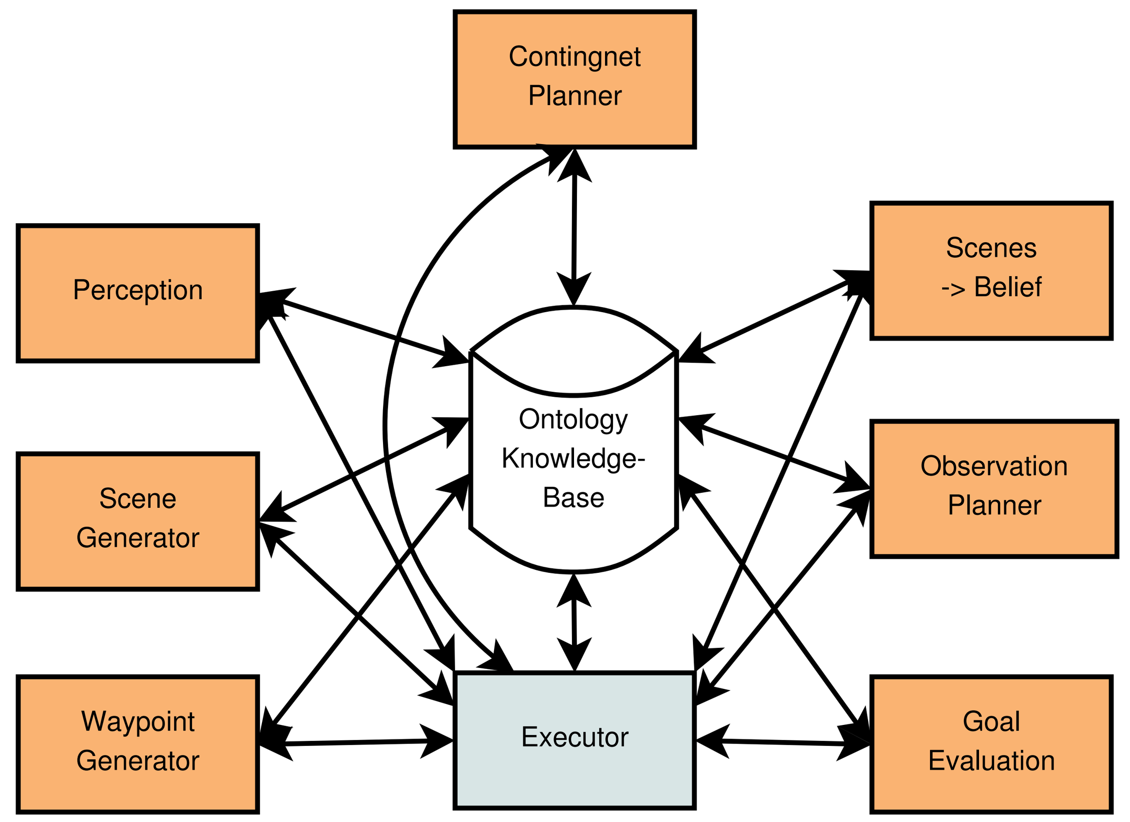


System components are:

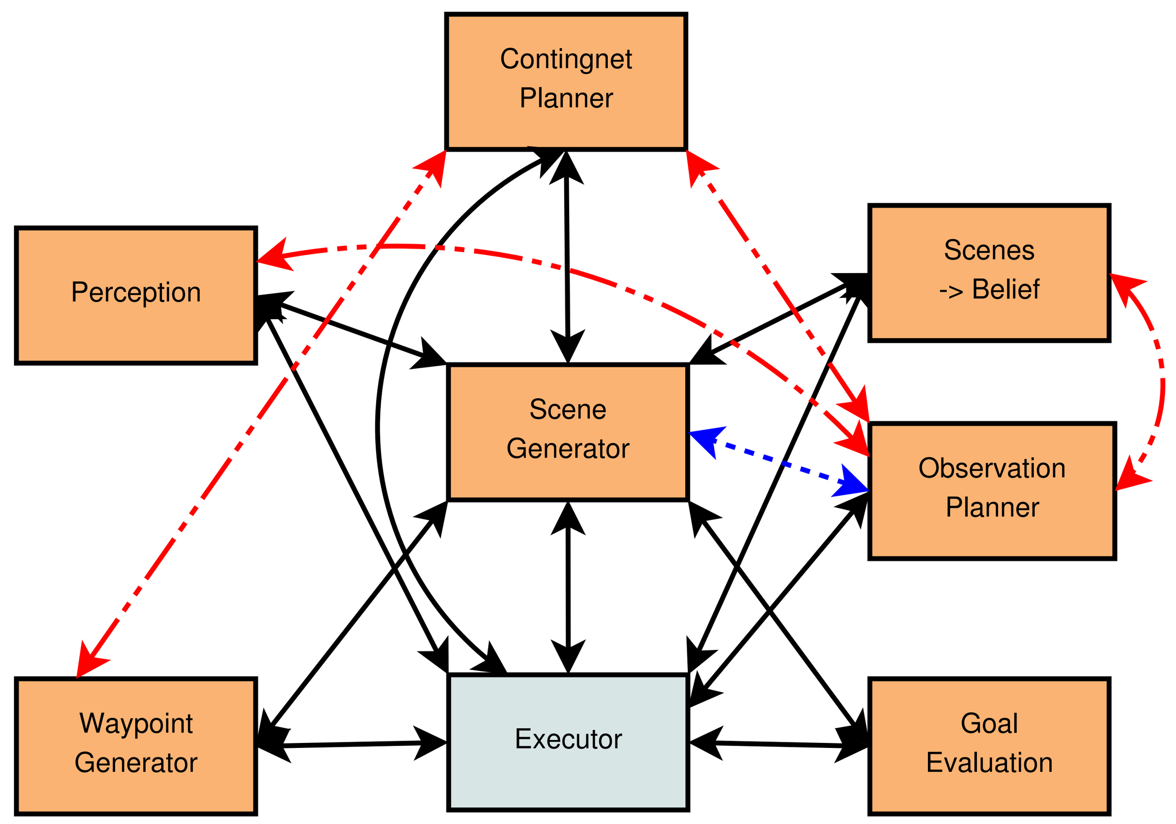
knowledge-base the central ontological database
perception extracts rectangular faces from RGB-D point clouds
scene-generator builds up a set of `Scenes`
waypoint-generator creates good observation poses
contingent-planner uses the ontology data to generate smart inspection paths
observation-action-planner handles sensor noise by re-running observation actions
scenes-to-belief-state converts the scenes into a probabilistic map
executor executes a mission by invoking other components in turn
goal-evaluator determines if the target has been found

Evaluation

The evaluation of architectural aspects of robotic systems is extremely difficult, as the benefits are not easily quantified. A novelty of our approach is that we store all the knowledge of the robot in the knowledge-base, including the world model (on the left of the schema diagram) and the robot's plans and execution data (on the right). We illustrate how this simplifies the dependencies between components.



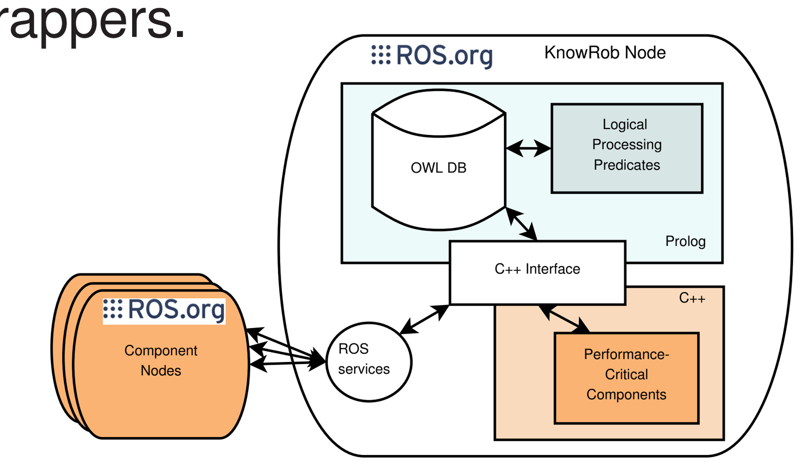
Compared to the actual system (above), a hypothesised system without an ontology (below) would have the extra dependencies shown in red, and would lose the dependency shown in blue.



With one component fewer, the non-ontology version still has slightly more inter-component connections. Also the *complexity* of the interactions is much higher; components have a varying number of interactions with unpredictable other components.

System Architecture

The system uses KnowRob, which is built on top of SWI-Prolog, to store the ontology. We developed a fast interface to the ontology using SWI-Prolog's C++ bindings, with ROS service wrappers.



Conclusions

Our ontological knowledge-base provides several benefits:

- New components can be added without needing changes to any existing components, as input/output is via the ontology
- A new developer does not need to guess at the interactions of a component: they are all invoked by the executor, perform some processing, and store results into the knowledge-base
- Domain engineering outputs can be re-used
- Editing, consistency checking and reasoning tools are readily available
- The format for storing and amalgamating knowledge can easily be understood by humans

These make the system more robust, simpler to build, and easier to modify and maintain. These advantages will become more important as robotic systems increase in complexity – according to Brooks [1]

> *The most pernicious and subtle bugs are system bugs arising from mismatched assumptions made by the authors of various components.*

References

[1] Brooks, F.P.: The Mythical Man-month: Essays on Software Engineering. Addison Wesley (1995)

[2] Saigol, Z., Ridder, B., Wang, M., Dearden, R., Fox, M., Hawes, N., Lane, D.M., Long, D.: Efficient search for known objects in unknown environments using autonomous indoor robots. In: IROS Workshop on Task Planning for Intelligent Robots in Service and Manufacturing (2015)

This work was supported by EPSRC