

ProDevSSR

Process Model for the Development of Semi-Autonomous Service Robots



Figure 1: Commanding a service robot on high abstraction level (here with brain computer interface)

Motivation

- **Task planning** is required for controlling a service robot on a high level (e. g. via command “Fetch meal”, etc.)
- The MASSiVE framework includes a task planner that is based on two paradigms:
- **Process structures** are used to flexibly pre-structure a task (reasonable limitation of task knowledge and enables offline verification)
- The user’s cognitive capabilities are included into task execution to achieve a system with manageable complexity (**semi-autonomous system**)
- The system development based on MASSiVE is a complex procedure that is entirely described and structured with the help of the **process-model ProDevSSR**

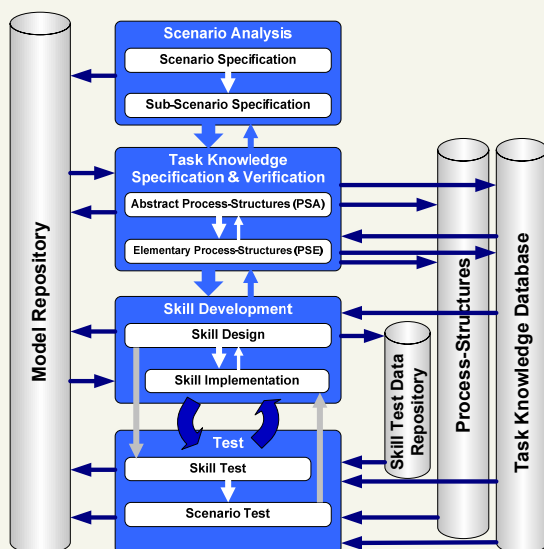


Figure 2: Process model ProDevSSR

Scenario + Model + Tool Driven Development

- **Scenario-driven** due to pre-structuring paradigm
- **UML model-driven** to provide an integrated view into a complex software system
- ... to have executable models in early development stages and thus detect defects early
- ... to provide an effective communication technique for developers
- **Tool-driven**, since complex processes cannot be successfully managed without tool support

Process Model ProDevSSR

- A process model defines the process steps; the tools, methods and repositories to be used; the in/output and feedback products throughout the development steps as well as the action flow.
- Eight development steps are defined and four repositories are used (see Figure 2)
- (a/b) **Scenario Analysis**: The system items and environmental items involved in the task are specified and added to the item ontology; the task is split into sub-scenarios
- (c/d) **Task Knowledge Specification and Verification**: Abstract and elementary process structures are defined for each sub-scenario. Formal verification based on Petri-Nets takes place.
- (e~h) **Skill Development and Test**: Based on the task knowledge specification the elementary functionalities of the system (skills) are developed and tested. This is done in a model-centric approach, supported by tools (UML CASE tool, test and visualization tools). Due to executable models the development is done in an iterative process, with skill simulation and execution from the initial skill design on.

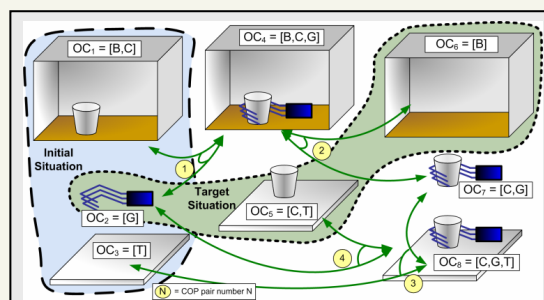


Figure 3: Abstract process structure (PS_A) for sub-scenario “Fetch cup”

Scenario + Model + Tool Driven Process Model to Develop Semi-Autonomous Service Robots

Aim

- Manage complexity of development process
- Consistency throughout the whole process
- Achieve uniform and reusable implementations
- Improve maintainability and all-time up-to-date documentation

Research

- Semi-autonomous systems
- Model-driven development, process models
- Task knowledge engineering (ontologies, specification and verification methods)
- Task planning and execution
- System testing

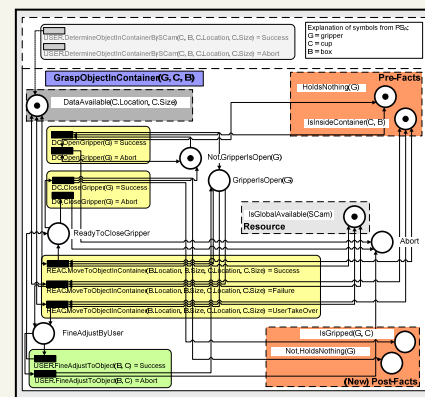


Figure 4: Elementary process structure (PS_E) “GraspObjectInContainer” as Petri-Net

Contact Person
Dipl.-Ing. Oliver Prenzel

Otto-Hahn-Allee
Gebäude NW1
D-28359 Bremen
Tel: +49 421 218-3594/-7523
Fax: +49 421 218-4596
prenzel@iat.uni-bremen.de
www.iat.uni-bremen.de