Dexmart:
Programming by Demonstration

Rainer Jäkel, Andreas Hermann, Martin Lösch,
Steffen Rühl, Sven R. Schmidt-Rohr, Zhixing Xue, Rüdiger Dillmann

FZI - Forschungszentrum Informatik
Dexmart: Artificial Cognitive System

Execution → Planning, coordination and control

Observation

Activity observation

Activity execution

Action observation

Action execution

Motion/grasp multisensor observation

Dual arm/hand control

Planning, coordination and control

Symbolic representation of goals

Level of abstraction
Planning model generation

Motivation: Generation of planning models from observation and reasoning

- Planning enables acting in complex, variable environments
- Generation of models describing the planning problem is difficult
- Approach:
  - Different levels of abstraction with complementary planning methods
  - Generation of planning models from human demonstrations and reasoning on the scene
Planning model generation

Architecture for generation of planning models from observation and reasoning

- Lowest layer:
  Constrained manipulation strategy planning Programming by Demonstration
Planning model generation

Architecture for generation of planning models from observation and reasoning

- **Top layer:**
  Probabilistic mission planning Programming by Demonstration
Planning model generation

Architecture for generation of planning models from observation and reasoning

- **Intermediate layer:**
  Symbolic activity decomposition classical planning
  - Introduces additional actions such that a learnt strategy can be applied in complex scenes
Planning model generation

Architecture for generation of planning models from observation and reasoning

System overview:
Planning model generation

- Manipulation strategy Programming by Demonstration
Motivation: Manipulation in restricted workspaces
- Goal-directed motion planning

Aim: Generate a Strategy graph
- Planning model based on task constraints
- Nodes = Subgoals, e.g. where to place the bottle in the fridge door
- Arcs = Transitions, e.g. keep the bottle “upright” during the motion

Approach: Programming by Demonstration
**Motivation:** Manipulation in restricted workspaces
- Goal-directed motion planning

**Aim:** Generate a *Strategy graph*
- Planning model based on task constraints
- Nodes = Subgoals, e.g. where to place the bottle in the fridge door
- Arcs = Transitions, e.g. keep the bottle “upright” during the motion

**Approach:** Programming by Demonstration
- Learning models for planning of manipulation strategies

---

Strategy graph

Constraints (red):
- bottle position relative to fridge door
Manipulation Strategy Planning Model PbD

Human Demonstrations / Learning of the initial planning model

- **Demonstration:**
  - Multiple demonstrations of the same task in sensory environment
  - Recording of finger joints, wrist poses, object poses, tactile measurements, contacts

- **Learning the initial planning model:**
  - Graph structure: segmentation
  - Feature space defined by large set of automatically generated task constraints
  - Learning of constraint parameters
  - Relaxation of constraints to consider “correspondence problem”

→ Planning model is overspecialized
Manipulation Strategy Planning Model PbD

Generalization of initial planning model

- **Problem:**
  - Generalization to different environments and different objects is limited

- **Teaching:**
  - “Curriculum learning”: second set of more complex demonstrations
  - Pruning of inconsistent constraints

- **Automated robot tests in new scenes:**
  - Identify a maximal subset of task constraints, which admits a successful plan on a set of robot tests
  - Statistics about constraint inconsistencies
  - Parallelized optimization
Problem:
- Generalization to different environments and different objects is limited

Teaching:
- “Curriculum learning”: second set of more complex demonstrations
- Pruning of inconsistent constraints

Automated robot tests in new scenes:
- Identify a maximal subset of task constraints, which admits a successful plan on a set of robot tests
- Statistics about constraint inconsistencies
- Parallelization

Dexmart Workshop, Humanoids 2011
Bled, Slovenia • 26th October 2011
Specialization / Execution

- **Specialization:**
  - Learning of search heuristics based on past experience to speed up planning

- **Execution:**
  - Constrained Motion Planning using *Strategy graph*
  - Physics simulation to simulate non-rigid contacts
  - Mapping to different robots: Adero, Albert II, Armar (simulation), Justin (simulation), PR2 (simulation)
Planning model generation

- Mission model Programming by Demonstration
Motivation: Robotic mission with manipulation
- Proactive, coarse grained activity selection

Aim: Generate a *symbolic-probabilistic mission model*
- State, activity sets
- Causal effect probabilities (transitions)
- Goal rewards, activity costs
  → Execution time policy is computed

Approach: Programming by Demonstration for efficient generation
- Generates systems/planning model of mission as (PO)MDP from human demonstrations
  - Manipulation activities are elementary actions
Mission Model Planning Model PbD

Learning models for selection of activities from demonstrations

- **Demonstration: robot based observation**
  - Recording of human poses, object poses, classification of abstract manipulation activities, speech

- **State abstraction**
  - Autonomous state discretization by clustering, considering timing of abstract activities

- **Manipulation action abstraction**
  - Autonomous mapping of observed activities onto manipulation strategies, learned on the lower level of abstraction, by trajectory analysis
**Mission Model Planning Model PbD**

Learning models for selection of activities from demonstrations

- **Model generalization with interactive verification**
  - Preliminary, abstract state-action transition model generated by counting frequencies in demonstrations
  - In a typical set of demonstrations, some causalities are missing
  - Preliminary model is analyzed for potentially important causalities, similar to observed ones
  - Demonstrating human is queried to perform additional demonstrations, including the most important causalities

<table>
<thead>
<tr>
<th>Observed: $p(s = S_{48} \mid u = U_3, s = S_{23}) = 0.7$</th>
<th>$p(s = S_{48} \mid u = U_3, s = S_{24}) = 0.57$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U_3 \quad \ldots \quad S_{47}' \quad S_{48}' \quad S_{49}' \quad S_{50}' \ldots$</td>
<td>$\ldots$</td>
</tr>
<tr>
<td>$S_{22}' \quad \ldots \quad 0 \quad 0 \quad 0 \quad 0 \ldots$</td>
<td>$\ldots$</td>
</tr>
<tr>
<td>$S_{23}' \quad \ldots \quad 0 \quad 0.7 \quad 0 \quad 0 \ldots$</td>
<td>$\ldots$</td>
</tr>
<tr>
<td>$S_{24}' \quad \ldots \quad 0 \quad 0 \quad 0 \quad 0 \ldots$</td>
<td>$\ldots$</td>
</tr>
<tr>
<td>$S_{25}' \quad \ldots \quad 0 \quad 0 \quad 0 \quad 0 \ldots$</td>
<td>$\ldots$</td>
</tr>
</tbody>
</table>

Generalizing over $s$, not $u$, not $s'$; $S_{24}'$ similar to $S_{23}'$

Computing confidence and transition estimate:

- Confidence: $p(s = S_{48} \mid u = U_3, s = S_{24}) > 0.83$
- Estimated $p(s = S_{48} \mid u = U_3, s = S_{24}) = 0.57$

Relevance ranking $p(s = S_{48} \mid u = U_3, s = S_{24}) : #7$

Computing sequence suggestion:

- $S_{73} \rightarrow U_{11} \rightarrow S_{24} \rightarrow U_3 \rightarrow S_{48} \rightarrow U_8 \rightarrow S_{11} \ldots$

Several human demonstrations:

- Demonstrated $p(s = S_{48} \mid u = U_3, s = S_{24}) = 0.4$

<table>
<thead>
<tr>
<th>Model verification</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U_3 \quad \ldots \quad S_{47}' \quad S_{48}' \quad S_{49}' \quad S_{50}' \ldots$</td>
</tr>
<tr>
<td>$\ldots$</td>
</tr>
<tr>
<td>$S_{22}' \quad \ldots \quad 0 \quad 0 \quad 0 \quad 0 \ldots$</td>
</tr>
<tr>
<td>$S_{23}' \quad \ldots \quad 0 \quad 0.7 \quad 0 \quad 0 \ldots$</td>
</tr>
<tr>
<td>$S_{24}' \quad \ldots \quad 0 \quad 0.4 \quad 0 \quad 0 \ldots$</td>
</tr>
<tr>
<td>$S_{25}' \quad \ldots \quad 0 \quad 0 \quad 0 \quad 0 \ldots$</td>
</tr>
</tbody>
</table>
Autonomous model refinement: Knowledge inference

- Robot specific skill errors cannot be learnt from observation of humans
- Error states, transitions and actions costs have to be deduced from background knowledge
- Preliminary model is completed using inference on an ontology

Autonomous model refinement: Dynamic simulation

- Transition probabilities for manipulation activities deduced from knowledge are not very precise
- Thus, simulation of deviations of situations observed in demonstrations
- Robot executes activities, effects evaluated in dynamics simulation
Planning model generation

- Decomposition, scheduling and monitoring
Motivation

- How can we create action sequences which fulfill a task in a given scene?
  - E.g. “move the tea box”
- Requires interaction between
  - The continuous real world and
  - A symbolic representation
Decomposition of activities

Generating a symbolic scene description

- **Physical simulation**
  - Translates a geometric scene model into a symbolic
  - Generated by simulation of motion primitives

- **Described aspects**
  - Effects of the motion of one object to another
  - Predicts, if an object can be manipulated without undesired effects

Dexmart Workshop, Humanoids 2011

Bled, Slovenia • 26th October 2011
Decomposition of activities

Graspability of objects

- Discretization of the table plane
- Graspability: Amount of graspable orientations in a 2D cell
  - How good can an object be grasped at that location
- Offline calculation by simulation of kinematics
- Application:
  - Find placement area
- Use only locations which are likely to be useful for planning
A symbolic planer adapts an action sequence based on the symbolic scene.

Actions are modeled using an occupancy grid:
- Freespace and
- Mechanical scene description as precondition.

Online queries to manipulation planner ensure feasibility of the execution.

Possible result: remove obstacles before manipulation.
Planning model generation

Example: Manipulation Strategy Planning Model PbD and Decomposition Planning

Artificial Cognitive System:
Learning manipulation strategies from human demonstrations for autonomous execution combined with decomposition planning and reachability analysis

– Progress and Integration Year 3 –
(Part of Brick A)
Planning model generation

Architecture for generation of planning models from observation and reasoning

Summary: