Social Robots and Human-Robot Interaction
Ana Paiva
Lecture 5. Building Social Robots: generic architectures, tools and frameworks
Scenarios we are interested in..

“Interactions between humans and robots are inherently present in all of robotics”

Focus on the Interaction
How to “Build” Social behaviour in a Robot?
Characteristics of a “Social Robot”

Adapted from the characteristics proposed by Fong et al. we will consider that a social robot should have a set of “human-like social” characteristics that lead to specific:

1. S. establish/maintain Social relationships;
2. P. exhibit distinctive Personality and character;
3. L. Learn/develop social competencies.
4. E. Express and/or perceive emotions;
5. N. use of Natural cues (gaze +, gestures, etc);
6. D. communicate with high level Dialogue;
7. O. Other agents perception and modeling,
8. R. Recognition and understanding of the social situations;
A framework

1. Identify Patterns of interaction (through user studies, etc)
2. Build specific (start with simple) behaviours to the robots taking into account those identified patterns
3. Integrate those behaviours in a more generic architecture for social interaction and task execution
4. Conceptualize the social behaviours (developed in the architecture) for a specific scenario
5. Test the resulting system
ARCHITECTURES
Identifying modules for generating behaviour

What modules should we need?

Processing

Object Recognition
Memory
Decision Making
Planning

Sensors

Perceptions (sensory stimulation)

Effectors

Actions (motor actions)

Identifying modules for generating behaviour

What about **SOCIAL** actions?

**Processing**
- Object Recognition
- Memory
- Decision Making
- Planning

**Sensors**

**Effectors**

**Perceptions** (sensory stimulation)

**Actions** (motor actions)

*Pfeifer, Rolf, and Christian Scheier. From perception to action: The right direction?. IFI, 1995.*
According to Duffy

Figure 1. A social interface creates a social robot.
Identifying modules for generating behaviour

What about **SOCIAL** actions?

The Typical Approach

- **Processing**
  - Object Recognition
  - Memory
  - Decision Making
  - Planning

- **Sensors**
- **Effectors**

- **Actions** (motor actions)

- **Perceptions** (sensory stimulation)

- **Social behaviours**
Different types of architectures

- Layered-based architectures
- Competence-based architectures
- Functional module-based architectures
- Hybrid architectures
Layer based architectures

- The “behaviours” of the robot emerge from different “layers”
- Each “layer” should link Perception to Action
- There is a degree of complexity in the layers.
- Often for social robotics there is a “social layer”
Inspiration: R. Brooks – and the Subsumption Architecture

Main theses:

– Intelligent behavior can be generated *without explicit representations* of the kind that symbolic AI proposes

– Intelligent behavior can be generated *without explicit abstract reasoning* of the kind that symbolic AI proposes

– *Intelligence is an emergent property* of certain complex systems
Brooks - Subsumption Architecture

Decision-making: task-accomplishing behaviors

Behavior:
- Maps perceptions to actions
- Intended to achieve some task
- No complex symbolic representation
- No symbolic reasoning

```
Sensors →
    reason about behavior of objects
    plan changes to the world
    identify objects
    monitor changes
    build maps
    explore
    wander
→ Actuators
avoid objects
```
Yet, it is not so easy for social behaviours....
According to Duffy a robot whose architecture can be constructed on four consecutive layers:

(a) The **physical layer**: The robot has a form in an environment and this is steered by primitive motor activities.

(b) The **reactive layer** treats fundamental sensorical reflexes

(c) **deliberative layer** which is a Beliefs-Desires-Intentions (BDI) architecture.

(d) Finally, the **social layer** is calculating the communication via an Agent Communication Language.
Example: Cognitive-based layered Architecture *

(a) The **physical layer**: The robot interface to the world (its body) including its sensors and motor controllers.

(b) The **reactive layer** treats fundamental sensorical reflexes.

(c) The **deliberative layer** which is a Beliefs-Desires-Intentions (BDI) architecture.

(d) Finally, the **social layer** is calculating the communication via an Agent Communication Language.

Different types of architectures

- Layered-based architectures
- Competence-based architectures
- Functional module-based architectures
- Hybrid architectures
Competence based architectures

• Similar to layered based
• But, each “competence” (or even behaviour) is implemented as a module
• Competencies may be built by a combination of other competencies
• There are shared “modules” that competencies may rely upon
Example: Competency-based architecture

Figure 2: Overview of the different sub systems’ place within the three conceptual levels and of the connections between sub systems.
Different types of architectures

- Layered-based architectures
- Competence-based architectures
- Functional module-based architectures
- Hybrid architectures
Functional based architectures

• Each module of the architecture has an internal “function” (e.g. natural language understanding, ….)
• Modules may communicate with each other and share common information
• Typical modules (some examples)
  – Navigation
  – Decision-making
  – Memory
  – Gaze
  – Object identification
  – Dialogue manager
  – Speech recognition
  – Theory of mind
Example: ACT-R/E

At a high level, ACT-R/E is a hybrid symbolic/subsymbolic production-based system.

- It allows the representation of declarative knowledge (fact-based memories) and procedural knowledge (rule-based memories), as well as input from the world (visual, aural, etc.)
- ACT-R decides what productions to fire next; these productions can change either its internal state (e.g., by creating new knowledge) or its physical one (e.g., by pressing a key on a keyboard).
Example: ACT-R and ACT-R/E Architectures

Figure 1. Architectural diagram for ACT-R/E.
Decision making is associated with the decision of what production to fire next.

a) *symbolic* knowledge, such as who was where at what time; and

b) *subsymbolic* knowledge, such as how relevant a fact is to the current situation, or how useful a production is expected to be when fired.
ACT-R is made up of several major components.

- First, it has several limited-capacity buffers which, together, comprise its context.
- Each buffer is backed by one (or more) theoretically motivated modules (e.g., declarative, visual, aural, etc.);
- There is the procedural module, which does not have any associated buffers.
- Each module represents a specific cognitive faculty and has been shown to have anatomical correspondences in the brain.
ACT-R/E Modules

- **Intentional and Imaginal Modules** provide support for task-oriented cognition. The goal buffer (intentional module) is intended to contain chunks that are used to identify the model’s current goal and provide place-keeping within the task. The imaginal buffer, in meant to provide support for intermediate problem state representations.

- **Declarative Module** manages the creation and storage of factual knowledge, or *chunks*. In addition, it manages memories’ subsymbolic information by constantly updating their activation values. Activation values are a function of how frequently and recently

- **Procedural Module** creates and stores procedural knowledge, or productions. Subsymbolic information for production rules is represented by an *expected utility*. Expected utilities are learned over time according to a temporally discounted reinforcement learning function.

- **Visual and Aural Modules** enables the architecture to see elements in the model’s world.

- **Spatial Perception** Spatial cognition is a core component of embodied cognitive models because it is a critical aspect of embodiment. Picking up objects, remembering where one’s keys are, tracking people as they move around, and searching for a light switch all require fundamental spatial competency.

- **Temporal Module** The temporal module allows the architecture to keep track of time in a psychologically plausible manner.

- **Manual and Speech Modules** is concerned with providing robot-related effectors to the model. The *Manipulative* module, in turn, combines visual and motor information to represent objects in the immediately graspable space.
Example
<table>
<thead>
<tr>
<th>Task</th>
<th>Components of ACT-R/E</th>
<th>Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaze following</td>
<td>Manipulative module</td>
<td>Corkum &amp; Moore (1998)</td>
</tr>
<tr>
<td></td>
<td>Configural module</td>
<td>Moll &amp; Tomasello (2006)</td>
</tr>
<tr>
<td></td>
<td>Utility learning</td>
<td></td>
</tr>
<tr>
<td>Hide and seek</td>
<td>Imaginal module</td>
<td>Trafton, Schultz, Perzanowski, et al. (2006)</td>
</tr>
<tr>
<td></td>
<td>Visual module</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Vocal module</td>
<td></td>
</tr>
<tr>
<td>Interruption and resumption</td>
<td>Declarative module</td>
<td>Trafton et al. (2012)</td>
</tr>
<tr>
<td></td>
<td>Intentional module</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Imaginal module</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Procedural module</td>
<td></td>
</tr>
<tr>
<td>Theory of mind</td>
<td>Declarative module</td>
<td>Leslie, German, &amp; Polizzi (2005)</td>
</tr>
<tr>
<td></td>
<td>Architecture as a whole</td>
<td>Wellman, Cross, &amp; Watson (2001)</td>
</tr>
</tbody>
</table>

Table 1: Human-robot interaction tasks to which the ACT-R/E cognitive architecture has been applied.
TOOLKITS
Robot Behavior Toolkit

Robot Behavior Toolkit*

- *Robot Behavior Toolkit*, an open-source implementation of a social robot framework as a Robot Operating System (ROS) module;
- Has a community-based repository for behavioral specifications, and an evaluation of the effectiveness of the Toolkit in using these specifications to generate social behavior in a human-robot interaction study, focusing particularly on gaze behavior.

Robot Behavior Toolkit*  

Figure 1: Our framework that integrates (1) social-scientific specifications of human social behavior and (2) an interaction model inspired by Activity Theory to guide the generation of humanlike behavior for robots.

Robot Behavior Toolkit*

Robot Behavior Toolkit*


Figure 2: The Robot Behavior Toolkit consists of three subsystems—the perceptual, cognitive, and behavior systems; two memories—the working memory and long term memory; and supporting components—the activity model and knowledge base.
Robot Behavior Toolkit*

Cognitive System
The cognitive system takes external and internal information and generates a set of triggers that form a context model of the current situation. This sub-system consists of three main components: the world manager, activity manager, and context generator.

Knowledge Base
The knowledge base stores a collection of behavioral specifications from the social-scientific literature on human social behavior.

Figure 2: The Robot Behavior Toolkit consists of three subsystems—the perceptual, cognitive, and behavior systems; two memories—the working memory and long-term memory; and supporting components—the activity model and knowledge base.
Figure 2: The perceptual, cognitive, and behavior systems; two memories—the working memory and long term memory; and supporting components—the activity model and knowledge base.
Behavior System
The behavior system generates humanlike behaviors based on the current context and behavioral specifications and includes three components: behavior selector, behavior coordinator, and behavior generator.

- The behavior coordinator resolves conflicts and/or overlaps among specifications by prioritizing them.
- Finally, the behavior generator organizes coordinated behaviors in an XML format for execution.

Integration with the Robot Operating System (ROS) by implementing the Toolkit as a ROS node.


Figure 5: An example behavior output generated by the Toolkit in XML (top) and in visual representation (bottom).
Robot Behavior Toolkit*

TOOLS
CASE 2
Main goal:

1. Allow for easy construction of social behaviours in robots;
2. Seamlessly integrate agent’s logic and decision making with body animation, navigation and manipulation;
3. Support various types of embodiments (virtual & robotic)
4. Allow for mixed environments (virtual & physical)
SERA is a toolkit for building social robots, composed of both a model and associated tools for integrating a cognitive agent with a robotic embodiment, in human-robot interaction scenarios.
Components of the SERA model

There are several components developed in order to be generic and reusable across applications and robotic embodiments.

Major components:

- *Thalamus*, which is the backbone integration middleware,
- *Skene* which is a semi-autonomous behavior planner, and
- *Nutty Tracks* which is a symbolic animation engine
Thalamus: Component integration framework

C# - Simple and fast for prototypes
Works out of the box
Kinect, Unity..

High-level, symbolic messages
Strongly typed shared messages
Agent/Character isolation
Conceptually less technical than ROS
Example of use of Thalamus
Thalamus: Component integration framework: example
Components of the SERA model

There are several components developed in order to be generic and reusable across applications and robotic embodiments.

Major components:

- *Thalamus*, which is the backbone integration middleware,
- *Skene* which is a semi-autonomous behavior planner, and
- *Nutty Tracks* which is a symbolic animation engine.
Skene: Behaviour Planner

• Semi-autonomous behaviour planner
• Reused in different applications and embodiments
• Outputs BML actions
Skene: Behaviour Planner

- High-level utterances with NBV markup
- Schedules speech with
  - Gazing, Glancing, Pointing, Waving, Animations, Application Commands…
- Model of the Environment
- Gaze-state machine
- Questions/Backchanneling
Skene: Behaviour Planner

Skene

Decision-Making

FML/Utterances

Utterance Parser

Gaze/Glance

Behaviour Instructions

Gaze State Manager

Targets Manager

Physical Space Manager

Target Information (e.g. User tracking, Screen objects)

Coordinates Transformation

Target Angle Coordinates

Config file

Actions Feedback

BML and Application Actions

Character

Application
In order to have more space to draw

press

the eraser icon.

Just try it.

<pause=300>
Integration
Components of the SERA model

There are several components developed in order to be generic and reusable across applications and robotic embodiments.

Major components:

- *Thalamus*, which is the backbone integration middleware,
- *Skene* which is a semi-autonomous behavior planner, and
- *Nutty Tracks* which is a symbolic animation engine
Nutty Tracks

- Symbolic real-time animation system that can be used to design and run interactive animation on any robotic character using animation tools commonly used by professional animators.
  - Automated animation
  - Interactive parameters
  - Abstract, high-level degrees-of-freedom
  - Character-independent animation
  - Both Robotic and CG character animation
Nutty Tracks: Innovation

• 100 years of animation techniques…
• Robots:
  *Interactivity vs. Quality of the Animation*
• Borrowed some concepts from the CG pipeline
• Other tech tools for artists
Tools

Nutty Tracks - Inspiration

Unreal Development Kit (UDK)
Components to build in order to follow the design patterns

– Perception of the user/s
  • Body Posture, Gestures and Facial expressions
  • Gaze
  • Actions towards the robot (eg. Hand shake)
  • Speech and Natural Language understanding
  • Environment perception
Components to build in order to follow the design patterns

– Decision making
  • Reasoning about the actions of the user & ToM
  • Task based decision making
  • Emotional and Social behaviour
  • Monitoring execution and manage interruptions
  • Manage turn-taking and Dialogue generation

– Action execution
  • Navigation in the world
  • Manipulation of objects in the world
  • Speech and facial expressions
  • Manage internal constraints (different bodies)
Discussion