Cognitive Robotics:
The science of building intelligent autonomous robots and software agents

Giuseppe De Giacomo

University of Roma “La Sapienza”, Italy
Miegunyah Fellow, University of Melbourne, Australia
Cognitive Robotics:
The science of building intelligent autonomous robots and software agents

INTRODUCTION
Introduction

• Cognitive Robotics is the science of building autonomous robots/agents that operate in changing, incompletely known, unpredictable environments.

• Cognitive robots are “smart”: have autonomy, including autonomous reasoning and planning capabilities.

• Cognitive robots are already a reality.
Example: “Curiosity” the Latest NASA Mars Rover

- “Cognitive robotics” in a NASA cartoon!
Example: “Curiosity” the Latest NASA Mars Rover

• The previous video is referring to Curiosity, the latest NASA Mars Rover (a large, mobile laboratory), which started its mission a year ago (landed on Mars on Aug. 6, 2012)
• Within the first 8 months of a planned 23-month primary mission, Curiosity met its major objective of finding evidence of a past environment well suited to supporting microbial life.
Robots Come in a Variety of Shapes

- Nowadays through increasing use of software (and hence CS) robots are increasingly acquiring forms of autonomy

**Rhex**: power - and computation - autonomous hexapod robot with compliant legs and only one actuator per leg, developed by a consortium of American universities, including University of Michigan, McGill University, Carnegie Mellon University, University of California, Berkeley, Princeton University, Cornell University, University of Pennsylvania, within the RHex DARPA project. (http://kodlab.seas.upenn.edu/RHex/Home)

**SuperBot**: modular, multifunctional and self-reconfigurable, autonomous robot developed at Polymorphic Robotics Laboratory, University of Southern California (http://www.isi.edu/robots/)

**BigDog**: developed by Boston Dynamics spin-off of MIT, is a rough-terrain robot that walks, runs, climbs and carries heavy loads. BigDog is powered by an engine that drives a hydraulic actuation system. BigDog has four legs that are articulated like an animal’s, with compliant elements to absorb shock and recycle energy from one step to the next. BigDog is the size of a large dog or small mule. (http://www.bostondynamics.com/robot_bigdog.html)
Australia is Strong in Robotics!

Hugh Durrant-Whyte, NICTA, Australia's ICT Research Centre of Excellence: “Australia is a great place to do robotics”

Brisbane freight terminal completely robotized and run remotely from Sydney.

Open sky mines in remote places can take enormous advantage from robots that run autonomously and are controlled remotely.

See Whyte’s talks at:
• 22nd International Joint Conference on Artificial Intelligence IJCAI, Barcelona 2011 (50min)
• TED Sydney 2012 (15min)
Cognitive Robotics:
The science of building intelligent autonomous robots and software agents

HISTORY (PERSONAL VIEW)
It All Started with Shakey the Robot

• Shakey the Robot was built in late 60’s as a proof of concept of what we now call Cognitive Robotics (1966 – 1972, Charles Rosen, Nil Nielsen, and many others)

• Shakey
  – stored its knowledge on the world as set of fact
  – planned autonomously to achieve goals given by humans in a block world (STRIPS was invented for it)
  – executed (and monitored) its plans
Shakey Was Too Ahead of its Times

• Though it sounds incredible now, Shakey was seen as a sort of failure from the general point of view (too slow, required too ad hoc solutions, too costly to be applied in practice)

• “Cognitive Robotics” went under-radar for almost 20 years

• But, in fact, Shakey was a big success from a scientific point of view!

• Shakey set the research agenda in AI, generating several subfields that have shaped AI research in the 70’s, 80’s, 90’s, and even today
  – Robotics
  – Knowledge representation
  – Planning (though the current golden age of planning started only at the end of the 90’)
  – Machine learning
  – Vision
  – …
AAAI 1998 Fall Symposium on Cognitive Robotics

• Connect directly to the Shakey heritage

• Develop theory and implementation of robots/agents that act in an informed way in changing, incompletely known, partially observable, and unpredictable environments

• Basic tools:
  – Representation: model of the world state and of how it changes when actions are performed
  – Reasoning: for making informed selection of actions to be performed
Cognitive Robotics

- Connect directly with the Shakey heritage
- Develop theory and implementation of robots/agents that act in an informed way in changing incompletely known partially observable unpredictable environments

**Basic tools:**
- Representation: model of the world state and how it changes when actions are performed
- Reasoning: for making informed selection of actions to perform

**Participants:**
- Hector Levesque
- Yves Lesperance
- Eric Sandewall
- Martha Pollack
- John Pollock
- Sebastian Thrun
- Hector Geffner
- Chitta Baral
- Kurt Konolige
- G. De Giacomo
- Gerd Lakemayer
- Ray Reiter
- Murray Shanahan
The Cognitive Robotics Manifesto

As proposed at AAAI 1998 Fall Symposium on Cognitive Robotics

• Interesting agent behaviors require reasoning

• Knowledge Representation Hypothesis: Reasoning over explicitly represented facts leads to decisions about how to act

• And therefore mathematical logic (but not excluding probability and decision theory)

• Intellectual foundations within Knowledge Representation

• But maybe KR had become too concerned with “pure reason”

• CogRob is the natural reaction to purely deliberative KR in favor of principled theories of how deliberation relates to:
  – Real time computation and reactivity
  – Sensing
  – Deciding when and how to act
  – Execution monitoring
  – Online vs online computations
  – Belief revision
  – Etc

• In short CogRob is concerned with developing a unified theoretical and implementation framework for reasoning action and perception

• Robots implemented on these principles provide a criterion for success
The Toronto Approach

• In the 90’s two of the finest scientists in AI were both at University of Toronto:
  – Ray Reiter (1939 – 2002, IJCAI-93 Award for Research Excellence)
  – Hector Levesque (IJCAI-13 Award for Research Excellence)

• They were close friends

• Their interaction generated a beautiful quantum leap in research on Cognitive Robotics
The Toronto Approach

• **Logical foundation**: a variant of McCarthy’s Situation Calculus (1963, 1969)

• Distinguish two categories of actions
  – **Primitive actions**: simple, indecomposable, implementable
    Pick up a block, walk one step, open a door, perceive whether door is open.
  – **Complex actions**: composed of simple actions, using familiar programming language constructs:
    
    ```
    if (CarInDriveway) then drive else walk
    
    while (not Obstacle) do {
    walkOneStep;
    senseForObstacle
    }
    ```

• Develop **axiomatizations** for these in SitCalc

• Develop and implement SitCalc based **high-level programming languages** (*Golog, ConGolog, IndiGolog*) for defining complex actions

• **Extend classical SitCalc** to include
  – Time, processes, concurrency
  – Probability and decision theory
  – Knowledge to account for the effects of perceptual actions

• **Implement on autonomous robots/agents**
SitCalc-Based High-Level Programming Languages

• **Golog, ConGolog, IndiGolog, …**

• Generalize conventional imperative languages

• Normal imperative languages provide for procedure definitions that bottom out in simple operations on machine states, e.g., assignment statements

• In Golog (and variants) instead, the programmer specifies world states (via fluents), the primitive actions, and how these change states via successor state axioms

• Machine state becomes the world state

• Primitive operations on machine states become actions on the world state (their effect is axiomatized in SitCalc)

• Programs become executable specifications of complex behaviors at a high level of abstraction
The Toronto Approach is Still Very Active Today!

See proceedings of IJCAI, KR, AAAI, AAMAS

University of Toronto

H. Levesque  R. Reiter

Sapienza U, Rome

D. Nardi  G. De Giacomo

York U & UoT, Toronto

Y. Lesperance  S. McIlraith

UniMelb & RMIT, Melbourne

A. Pearce  S. Sardina

... and many others!!!
Cognitive Robotics:
The science of building intelligent autonomous robots and software agents

ROBOCUP
RoboCup (Robotic Soccer)

Rationale:
- Build cooperative autonomous multi-robot systems that play soccer
- Players in a team must cooperate to play well

Grand Challenge:
By 2050, build a team of fully autonomous humanoid robots that win against human world champion under the official regulation of FIFA

Currently RoboCup includes:
- Soccer
- Rescue + Home + Work
- Junior/Education

http://www.robocup.org/
<table>
<thead>
<tr>
<th>Year</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>1997</td>
<td>Nagoya</td>
</tr>
<tr>
<td>1998</td>
<td>Paris</td>
</tr>
<tr>
<td>1999</td>
<td>Stockholm</td>
</tr>
<tr>
<td>2000</td>
<td>Melbourne</td>
</tr>
<tr>
<td>2001</td>
<td>Seattle</td>
</tr>
<tr>
<td>2002</td>
<td>Fukuoka</td>
</tr>
<tr>
<td>2003</td>
<td>Padua</td>
</tr>
<tr>
<td>2004</td>
<td>Lisboa</td>
</tr>
<tr>
<td>2005</td>
<td>Osaka</td>
</tr>
<tr>
<td>2006</td>
<td>Bremen</td>
</tr>
<tr>
<td>2007</td>
<td>Atlanta</td>
</tr>
<tr>
<td>2008</td>
<td>Suzhou</td>
</tr>
<tr>
<td>2009</td>
<td>Graz</td>
</tr>
<tr>
<td>2010</td>
<td>Singapur</td>
</tr>
<tr>
<td>2011</td>
<td>Istanbul</td>
</tr>
<tr>
<td>2012</td>
<td>Mexico City</td>
</tr>
<tr>
<td>2013</td>
<td>Eindhoven</td>
</tr>
</tbody>
</table>

RoboCups
RoboCup-97 Nagoya

35 teams from 12 countries
RoboCup-13 Eindhoven
(http://www.robocup2013.org/)

Daniele Nardi
Current President of RoboCup

More than 40,000 visitors

Including Queen Maxima of the Netherlands!
RoboCup

The complexity comes from the dynamics of the environment (opponent):

- Complex perception-action
- Learning
- Cooperation/coordination
- Planning and game strategy

Soccer Leagues

- Simulated
- Small size
- Mid size
- Standard platform
- Humanoid
- RoboCup Rescue
- RoboCup@Home
Small-Size League

- Small and fast own built robots
- Global vision system
- Controlled by a remote computer
- Centralized system
  - Perception and action
  - Coordinated action
  - Teamwork
  - Learning the opponent model

2009 - Forward pass, moving ball interception
Standard Platform League

• Information on the game acquired through onboard sensors

• All computation Onboard
  ✓ Communication
  ✓ Articulated motion
  ✓ Robust perception
  ✓ Localization
  ✓ Teamwork
Standard Platform League

Standard robot NAO’s perspective on the world
RoboCup Rescue

- Goal: find victims in a disaster scenario

- Two categories of robots
  - Autonomous
  - Tele-operated

- Realistic rescue arenas & field exercises
  - Unknown
  - Unstructured
  - Dangerous

- Some robots used in
  - Fukushima 2012
  - L’Aquila 2008
  - …

- Rescue Robotics Camps
RoboCup Has Big Impact

• Research
  – Symposium
  – Publications
  – PhDs
  – Standard problems: Keepaway, …
  – Standard solutions: CMU Vision, …
  – Standard test methods: Rescue Arena, …
  – New platforms: NAO, Darwin, …

• Education
  – Beyond conventional curricula: complex, integrated system design, teamwork, competitiveness
    • Courses: multi-agents, multi-robot, software development, perception
    • Master theses
    • Camps
    • +Junior (high school)

• Citizen awareness
  – Simple and appealing message
  – Outreach (promoted by cities as science events for the general public)
  – High interest of media
Cognitive Robotics:
The science of building intelligent autonomous robots and software agents

SMART SPACES & ENVIRONMENTS:
BEHAVIOR COMPOSITION
Smart Environment as Cognitive Robots

- Currently communication among devices is easy
- Adding computing capabilities to devices is cheap
- Hard computation can be delegated to the Cloud

Spaces/environments themselves can become active and smart
- Even a common space such as a house may become a robot, or better a regulated community of simple devices.
Behavior Composition

Behavior composition in a nutshell

- **Common ground**: shared actions
- **Client** formulates target behavior using actions of common ground
- **Available behaviors**: described using actions of common ground
- **Composition system** realizes target behavior by “reversing” mapping and hence using fragments of available behaviors
Behavior Composition

- **Given:**
  - available behaviors B1, ..., Bn
  - target behavior T

  over the same *common ground*

- Check whether there exists an *orchestrator* that realizes T by *delegating* actions to B1, ..., Bn so as to *mimic* T over time (*forever!*)
Behaviors as Transition Systems

- Transition systems are mathematical devices to represent what an agent can do at each time point.

- They represent infinite behaviors in a compact way, as a sort of graphs.

- Technically similar to FSM, but equivalence based on bisimulation instead of mutual trace inclusion.

**Example:**

- $a$: "do action a"
- $b$: "do action b"
- $c$: "do action c"

![Diagram of a transition system with states $S_0$, $a$, $b$, and $c$.]
Example of composition

target behavior *(virtual!)*

available behavior 1

available behavior 2
Example of composition

target behavior (virtual!)

available behavior 1

available behavior 2

orchestrator
Example of composition

**target behavior (virtual!)**

**available behavior 1**

**available behavior 2**

**orchestrator**
Example of composition

target behavior (*virtual*)

available behavior 1

available behavior 2

orchestrator
Example of composition

target behavior *(virtual!)*

available behavior 1

available behavior 2

orchestrator
Example of composition

target behavior (virtual!)

available behavior 1

available behavior 2

orchestrator
Example of composition

**target behavior (virtual!)**

```
states: s1, s2, s3
initially: current state = s1
behavior:
if (current state == s1) {
        if (target request a) then {
            send a to available 1;
            current state = s2
        }
        if (target request b) then {
            send b to available 2;
            current state = s3
        }
    }
if (current state == s2) then
    if (target request c) {
        send a to available 1
        current state = s0
    }
if (current state == s3) then
    if (target request c) {
        send a to available 2;
        current state = s0
    }
end behavior
```
Composition Technique (Hint)

- **Problem:** Find orchestrator that realizes target $T$ by **delegating** actions to $B_1, \ldots, B_n$ so as to mimic $T$ over time (forever!)

- **Crux:** “mimic” captured by the formal notion of “simulation”

- **Solution:** Compute simulation, extract orchestrator program from computed simulation

- A binary relation $R$ is a **simulation** iff:

  $(s, t) \in R$ implies that
  - $s$ is final implies that $t$ is final
  - for all actions $a$
    - if $s \xrightarrow{a} s'$ then $\exists t'. t \xrightarrow{a} t'$ and $(s', t') \in R$

- A state $s_0$ of transition system $S$ is **simulated by** a state $t_0$ of transition system $T$ iff there exists a simulation between the initial states $s_0$ and $t_0$.

- Notably
  - **simulated-by** is a simulation
  - **simulated-by** is the largest simulation

*Note it is a co-inductive definition!*
Automated Program Synthesis

- Behavior composition is a different form of **planning**, or more generally a form of **automated program synthesis** (from components)
  - Very interesting results coming out of Rome, Toronto, Melbourne, and others

- Autonomous agents need to **program themselves automatically**, to act and react appropriately to the world around them

- **Automated program synthesis** is crucial
  - In general automated program synthesis is undecidable
  - However, planning in AI, behavior composition, service composition, discrete-event controlling, and several other areas are showing that automated program synthesis is feasible in significant cases!
Sophisticated Smart Spaces

• Behavior composition is just scratching the surface
  – Several on-the-fly configurable devices
  – Single client target behavior

• Much more sophisticated smart spaces/environments are conceivable:
  – Several on-the-fly configurable devices
  – Several client target behaviors acting simultaneously
  – Same device can be part of two different virtual agents at the same time (e.g., environmental camera shared by two virtual robots)
  – Devices that form the embodiment of a virtual agent change over time
  – We lose sense of embodiment!
  – We lose sense of unity!
  – How can a human interact comfortably with such kinds of systems?
  – HCI/psychological research is required!
Cognitive Robotics:
The science of building intelligent autonomous robots and software agents

VIRTUAL ENVIRONMENTS AND VIDEO GAMES
Non-Player Characters in Video Games

NPC: Non-Player Characters

- NPCs live and act in the game world as **autonomous** entities
- NPCs must look like acting intelligently within the video game
- The **behavior** of NPCs determines the success of a video game
Video Games Development Tools & Engines

Amazing tools available for game development

ShiVa3D

UDK

VALVE

Unity
Video Game Engine

How a video game engine works:

• Generally programmed in C++

• Creates game-world objects with (x, y, z) coordinates and calculates what happens to them on every frame

• E.g., a crate is up in the air on frame 1. On frame 2 the game engine will calculate the new position, etc

• Naïve Physics built in

• NPCs have a pre-programmed behavior
NPC Behavior Explicitly Represented

NPC behavior is typically based on

- Finite State Machines
- Behavior Trees
- Sometimes integrated with limited form of planning
NPC behavior based on high-level states

- **On Guard**
  - See small enemy
    - Fight
  - See big enemy
    - Escaped
  - Losing fight
    - Run away
- **Fight**
  - Energy OK
- **Run away**
Behavior Trees (BTs)

NPC behavior based on more refined conditions and strategies

- Door open?
  - Move into room
- Door locked?
  - Unlock door
  - Kick door
  - Door open?
Reactive NPC Behavior

A game level from the eyes of an NPC
Reactive NPC Behavior

Historically, virtually all video games with NPCs use FSMs and BTs for NPC decision making

• Very simple to understand and implement

• Separation between programmers and game designers

• Any extension needed can be handled effectively using programming tricks

• The behavior is enhanced by extra information in the game world that is ad-hoc prepared for NPCs
Reactive NPC Behavior

Both FSMs and BTs are reactive techniques

- The NPC follows a pre-programmed strategy that specifies how the NPC should react in the game depending on the current state and conditions that currently hold in the game-world

- A sequence of actions that may be executed in the game, e.g., [move to door, kick door, move into room], need to be represented explicitly in the structure of the FSMs or BTs
Reactive NPC Behavior

The situation today

• Open worlds with **increasing available interactions**
• NPCs need to be **autonomous**, with their **own agenda**, goals, personality
• Under these circumstances, maintaining the possible and applicable interactions using reactive techniques becomes **complex** and **difficult**
• The need for more **flexible** techniques arises
Situation Emerging Today

The “Basket Exploit”

The Elder Scrolls
Cognitive Robotics in Video Games

• What can AI research offer?
  – Knowledge Representation
  – Logic-based formalisms
  – Reasoning about actions
  – Automated behavior synthesis/refinement, reactivity …
  – Belief-Desire-Intention architecture, agent-based programming, …
  – Probabilistic reasoning, Bayesian networks, Utility theory, Markov Decision Processes, …
  – Machine learning, …
  – …

• In a nutshell: consider NPCs as a sort of virtual Cognitive Robots!
Interest not only in entertainment

But especially in so-called “serious games” i.e., virtual environments where people can act and interact among themselves and virtual agents

Humans are particularly engaged by video-game like interaction with software

But the more the interaction becomes free, the more NPCs need to gain a sort of understanding of the world they are in (through representation & reasoning)
Cognitive Robotics:
The science of building intelligent autonomous robots and software agents

CONCLUSION
What to Bring Home

• It all started with Shakey

• We owe a lot to Ray Reiter & Hector Levesque

• Forms of cognitive robotics are already around us …

• … and will be increasingly so in the future
What to Bring Home

Robots come in very different forms:

• **Traditional robots:**
  – Unity with fixed embodiment
  – On board computation
  – Incredible progresses (but more to be done)
  – They are meant to live in our environments as companions
  – Serious ethical issues may arise in the future as these robots become more and more capable

• **Non-traditional robots:**
  – No unity, no fixed embodiment
  – Sensors/actuators/devices distributed
  – Cloud based computation, i.e., “even intelligence distributed!”
  – Still in its infancy
  – But progresses may become extremely quick very soon!
    • Look for work in smart spaces, smart environments, service ecologies.
  – How to interact with these creatures? Big open question

• **Virtual robots / agents**
  – They are going to become common as the interaction with software becomes more free
  – Video games give us evidence already now
Thanks

Hummingbird Star Trek logo demo for the Earth Hour - March 23, 2013

- Adrian Pearce (UniMelb)
- Sebastian Sardina (RMIT)
- Daniele Nardi (Sapienza)
- Stavros Vassos (Sapienza)
- Massimo Mecella (Sapienza)
- Diego Calvanese (U Bolzano)
- Luca Iocchi (Sapienza)
- Hector Geffner (U Pompea Fabra, Barcelona)
- Alfonso Gerevini (U Brescia)
- Yves Lesperance (York U)
- Alessio Lomuscio (Imperial College London)
- Morri Pagnucco (U NSW)
- Fabio Patrizi (Sapienza)
- Alessandro Saffiotti (U Orebro)
- Moshe Vardi (Rice U)
- Hector Levesque (UoT)
- Ray Reiter (UoT)
- And many others!

Videos courtesy of:
NASA, Curiosity
UPenn, Rhex
USC, SuperBot
Boston Dynamics, BigDog
Hugh Durrant-Whyte, IJCAI, TED
SRI & Computer History Museum, Shakey
RoboCup.org
Anonymous, Elder Scroll Basket Esploit
Ascend Technologies, Hummingbird Star Trek logo demo