Обзор LASER-2017

Ю.В. Литвинов
А.Н. Терехов
y.litvinov@spbu.ru
a.terekhov@spbu.ru

29.09.2017

LASER-2017

- Летняя школа, с 9 по 17 сентября, о. Эльба, Италия
- Тема школы этого года: **Software for Robotics**
- Формат: 7 лекций по 45 минут в день + студенческие презентации
- 7 докладчиков, примерно по 6 лекций на каждого:
  - Davide Brugali, University of Bergamo
  - Rodolphe Gelin, Softbank Robotics
  - Ashish Kapoor, Microsoft Research
  - Nenad Medvidovic, University of Southern California
  - Bertrand Meyer, Politecnico di Milano
  - Issa Nesnas, NASA Jet Propulsion Laboratory
  - Hiroshi “Gitchang” Okuno, Waseda University and Kyoto University

Robot variability

Tourist guide

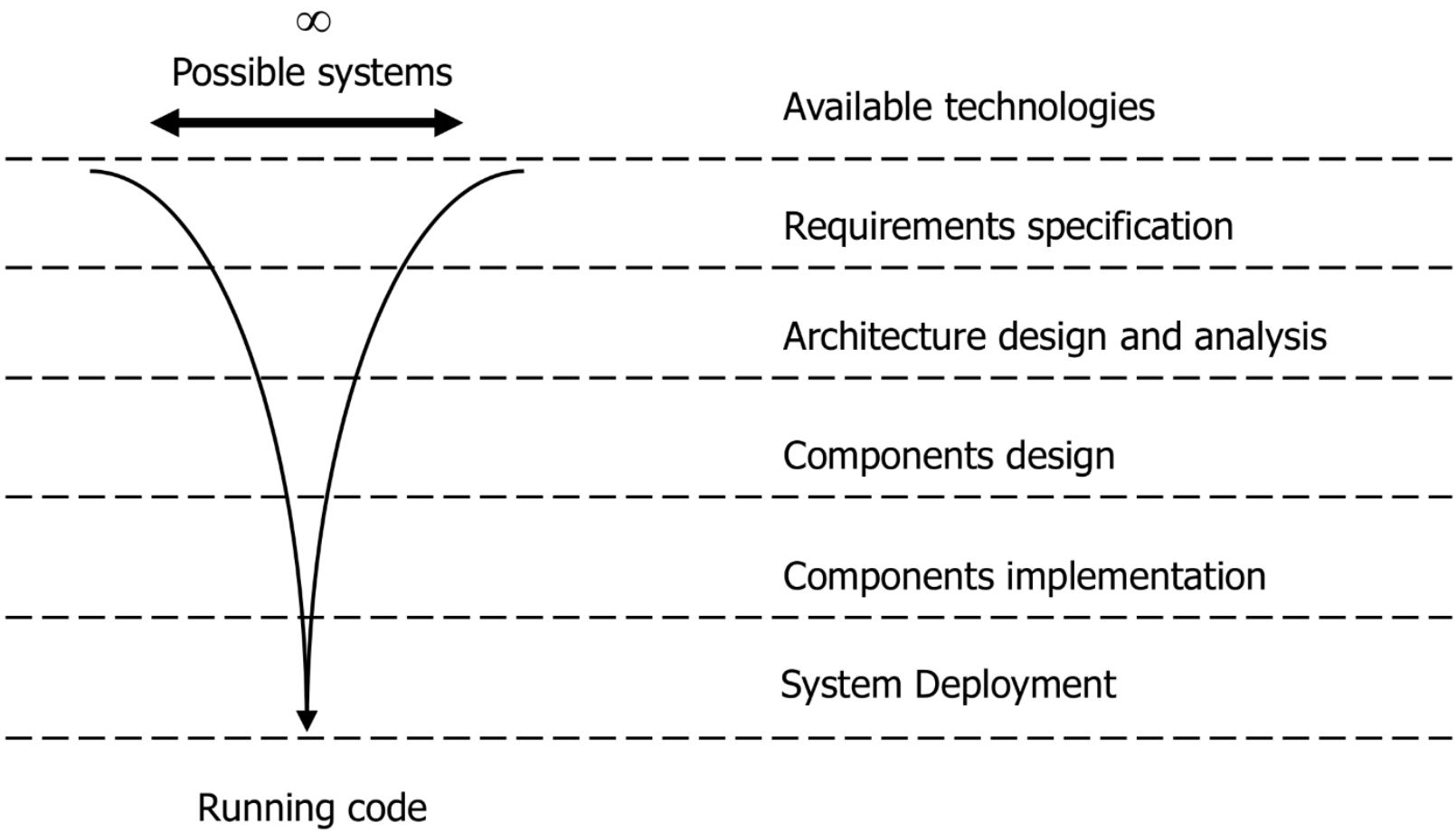
Store inventory

Autonomous cars

RoboCup

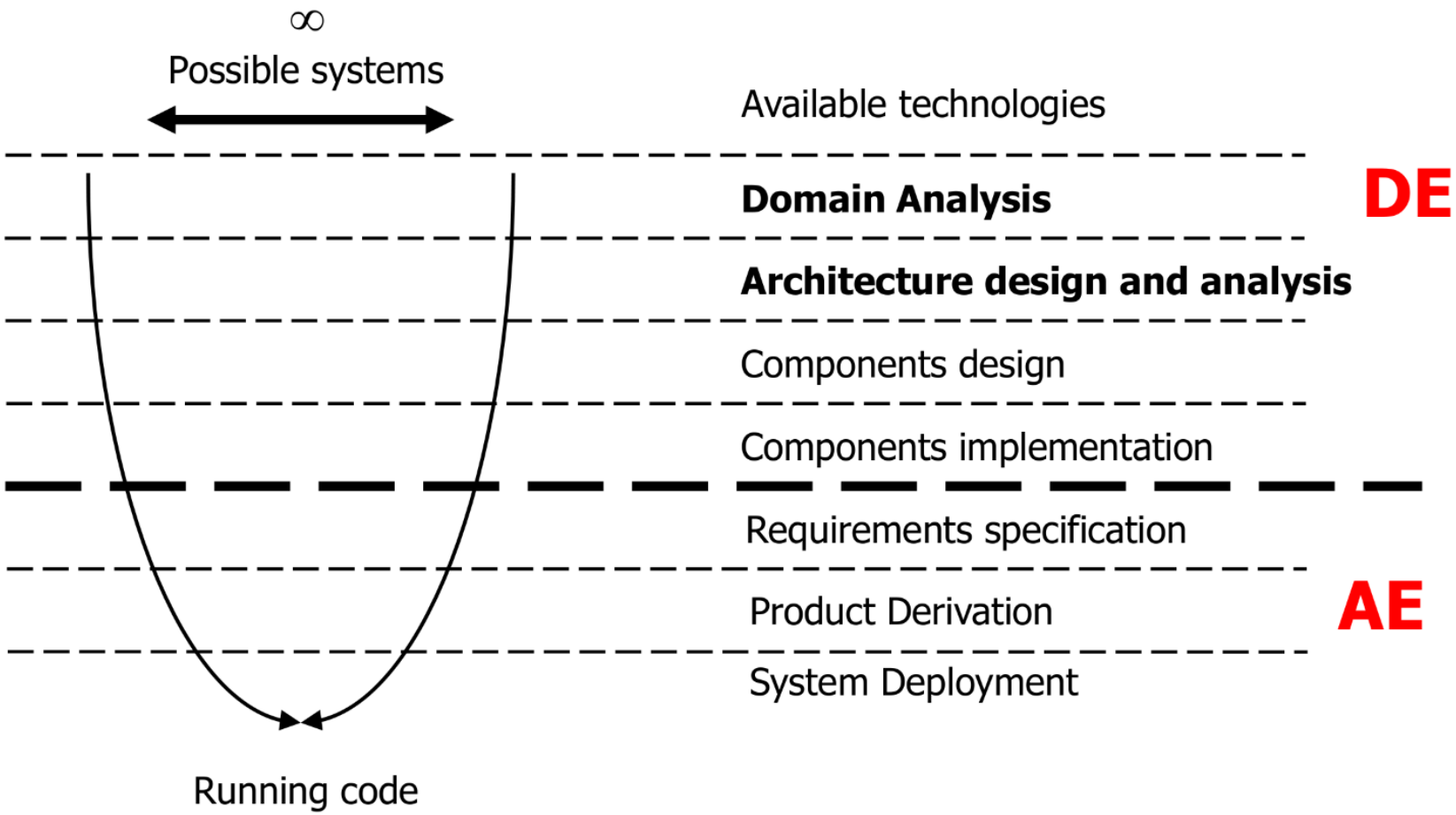
Shop Floor logistics

Traditional software development approach



Courtesy of Svahnberg, van Gurp, Bosch

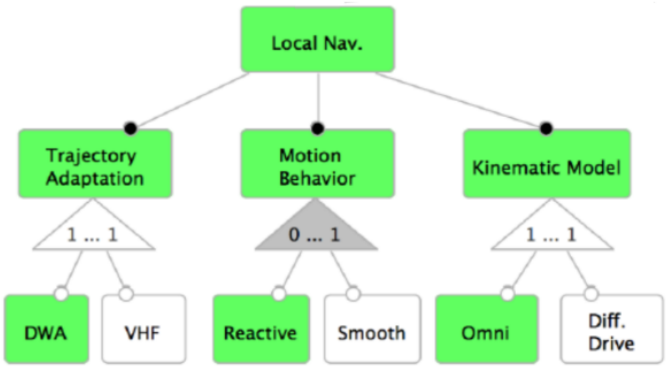
Software Product Lines



Courtesy of Svahnberg, van Gurp, Bosch

Product Line Engineering

The specification of
functional requirements

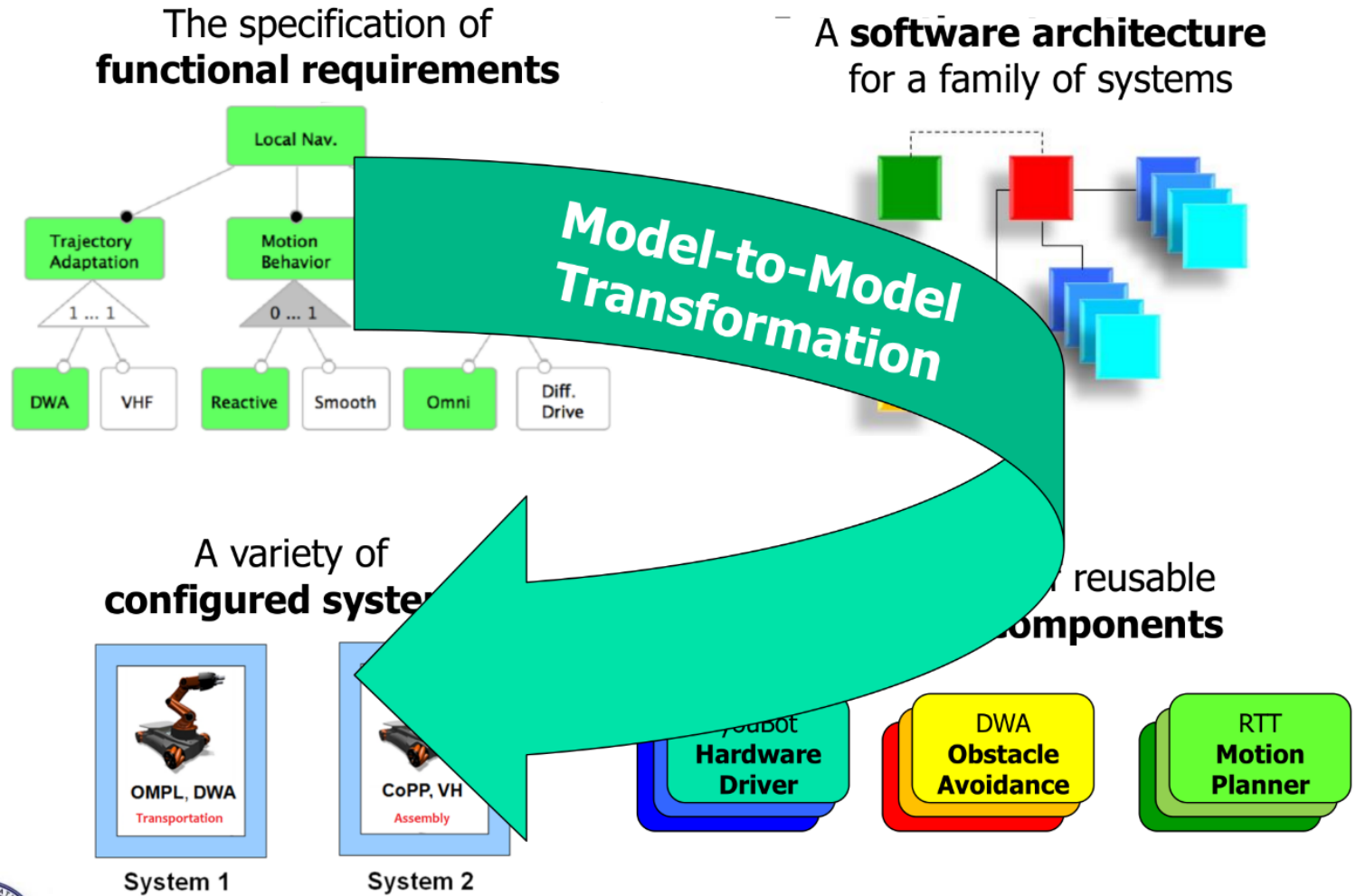


Integration structure

Products

Components

Product Line Engineering

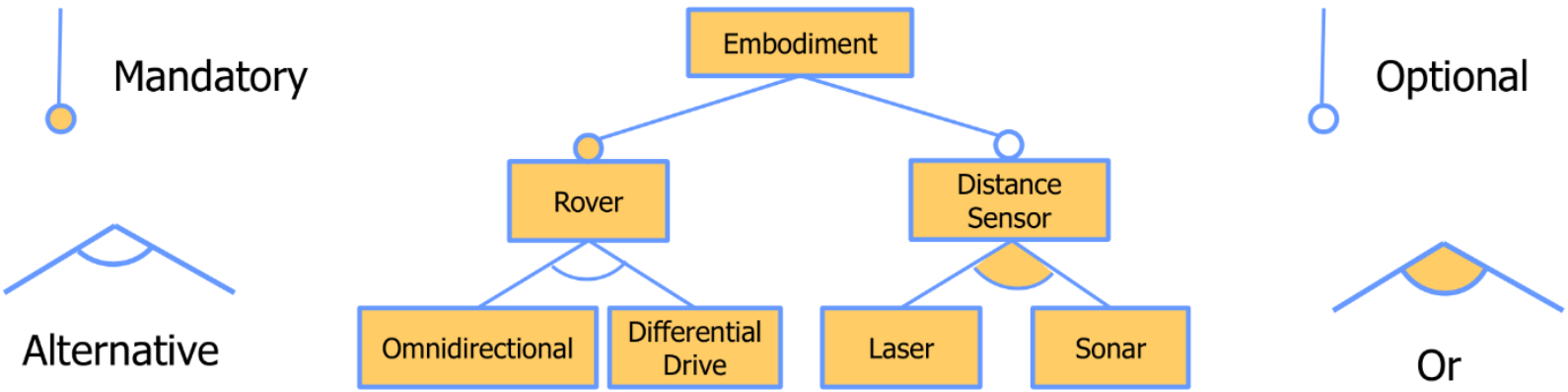


Domain Analysis

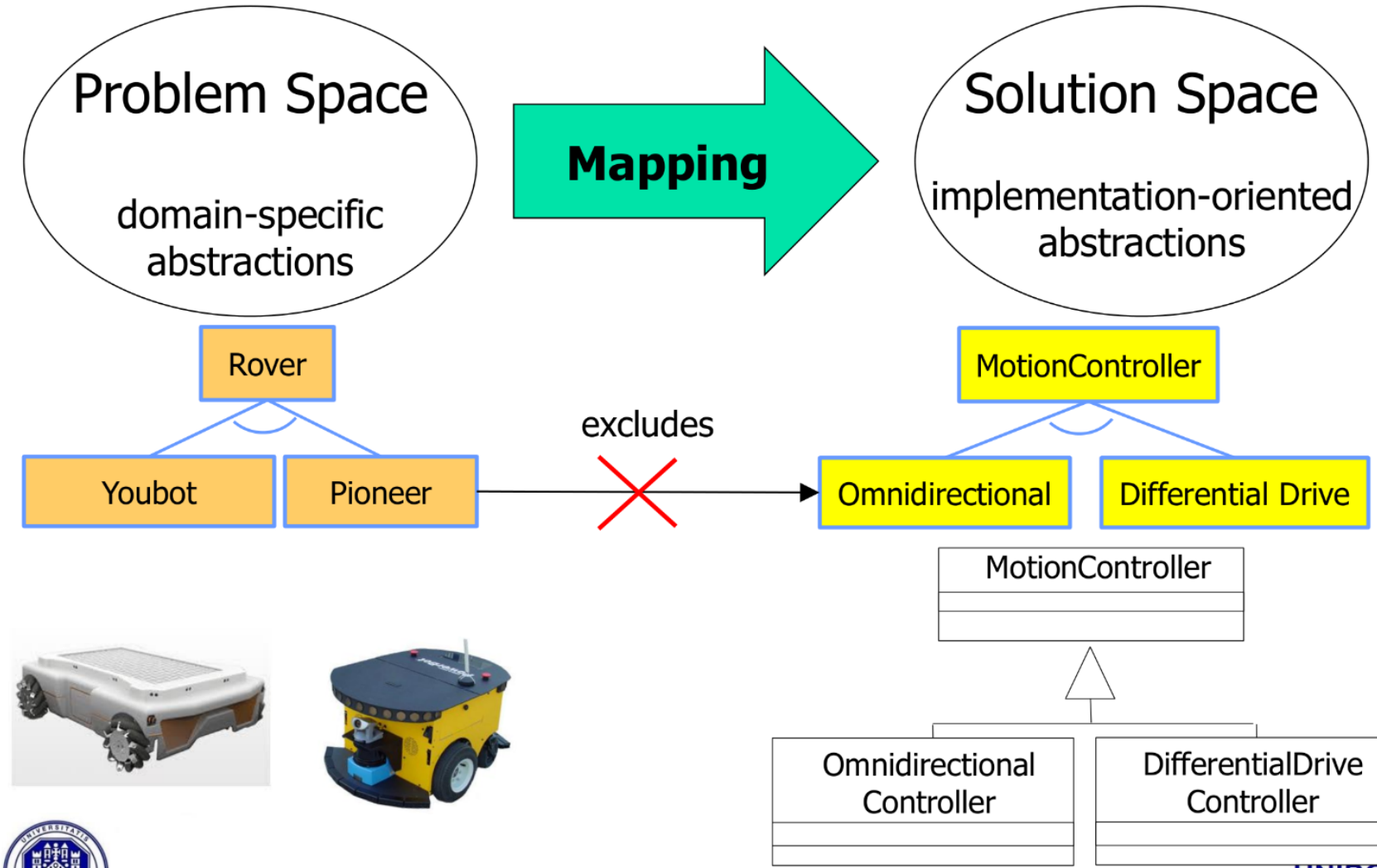
- Feature-oriented Domain Analysis (FODA)
  - Kang, K. et al. Feature-Oriented Domain Analysis (FODA) CMU/SEI 1990
  - Commonalities and variabilities in software systems
  - Feature: symbolic representation of a system characteristics
  - Feature Model: relationships among the features

- Stability Analysis
  - Software stability: a software system's resilience to changes in the original requirements specification.
  - Clien M. & Girou M. Enduring Business Themes. CACM, Vol 43(5), 2000
  - Fayad M.E. Accomplishing Software Stability. CACM 45(1), 2002

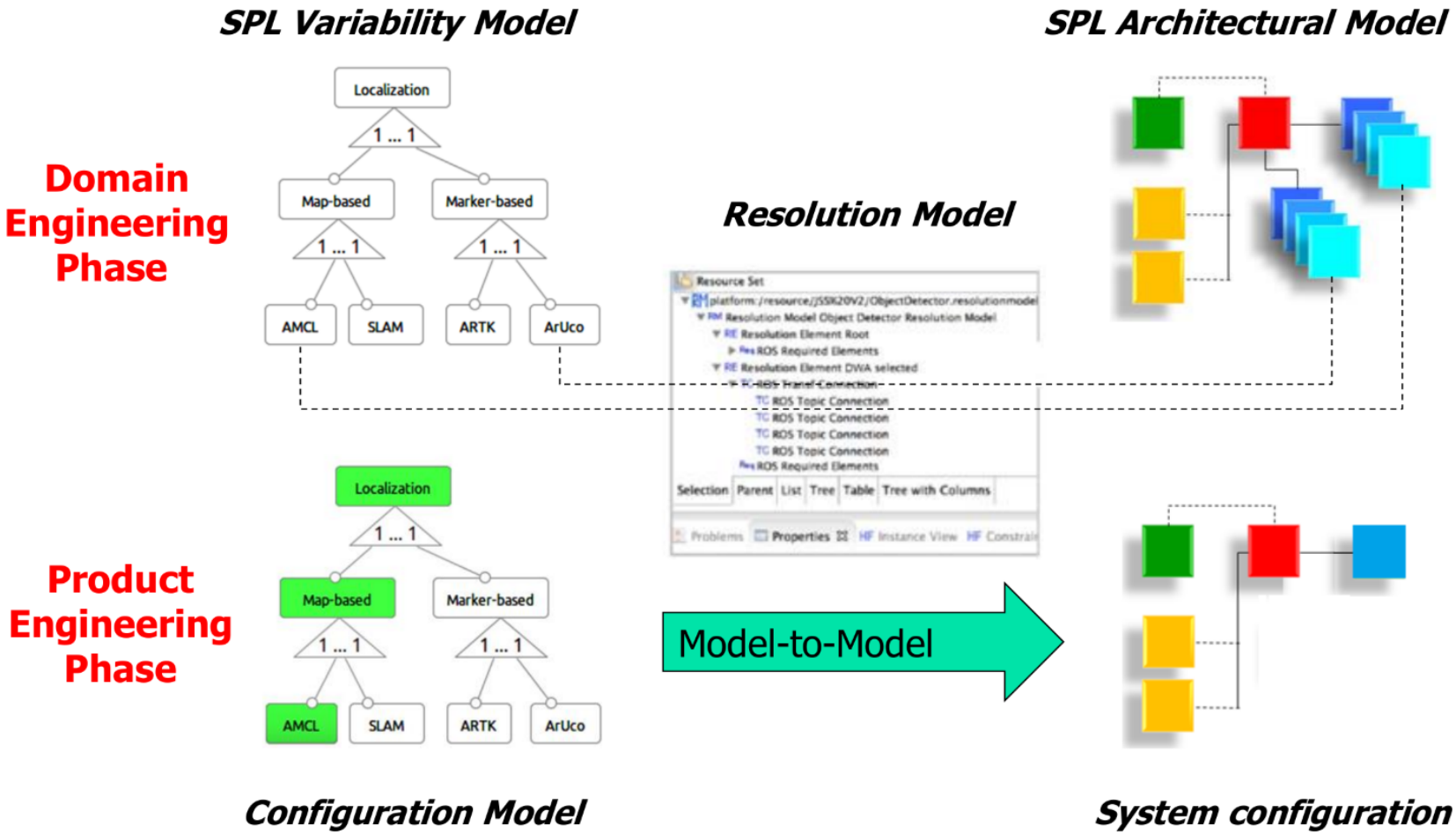
Feature Models



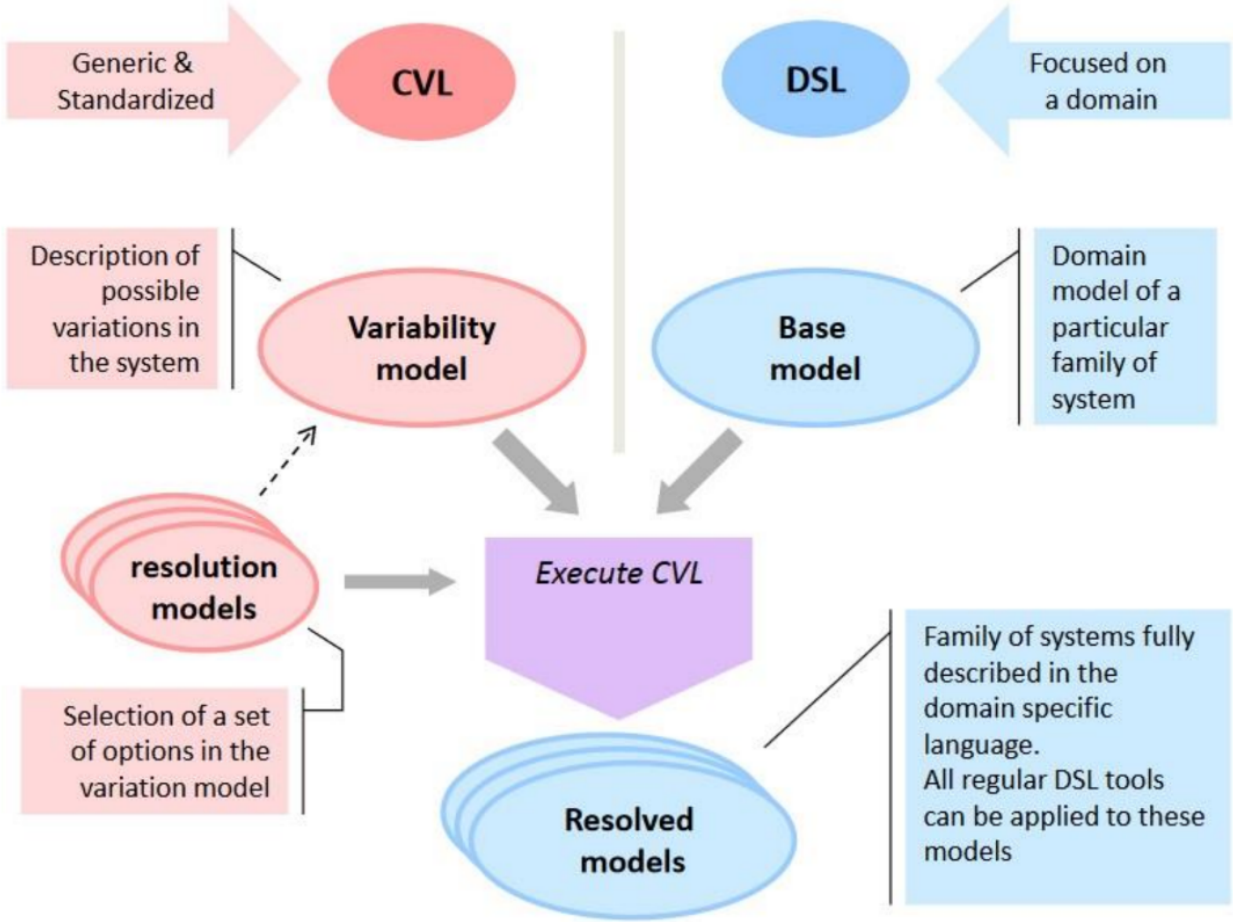
Modelling variability



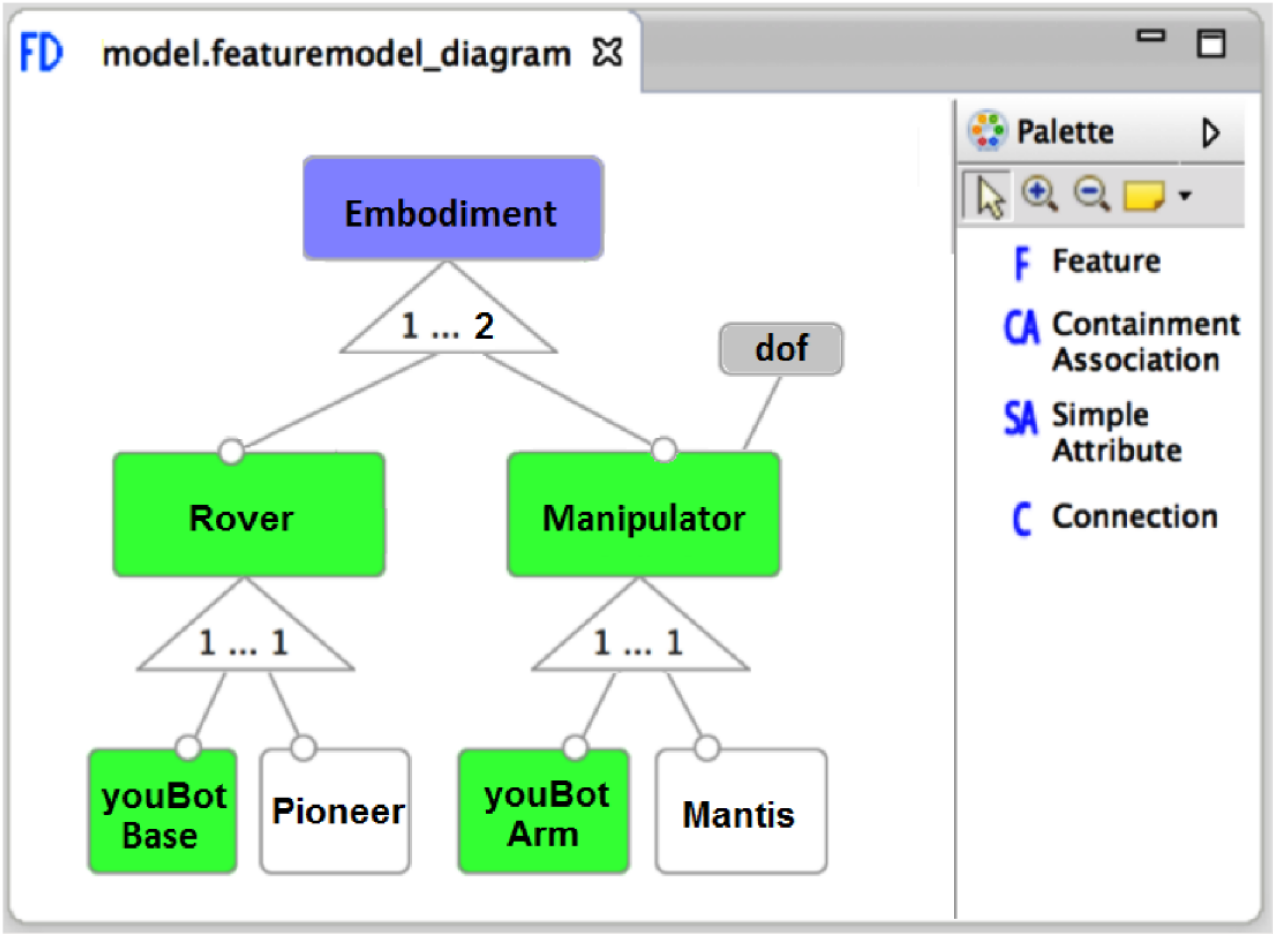
Orthogonal models



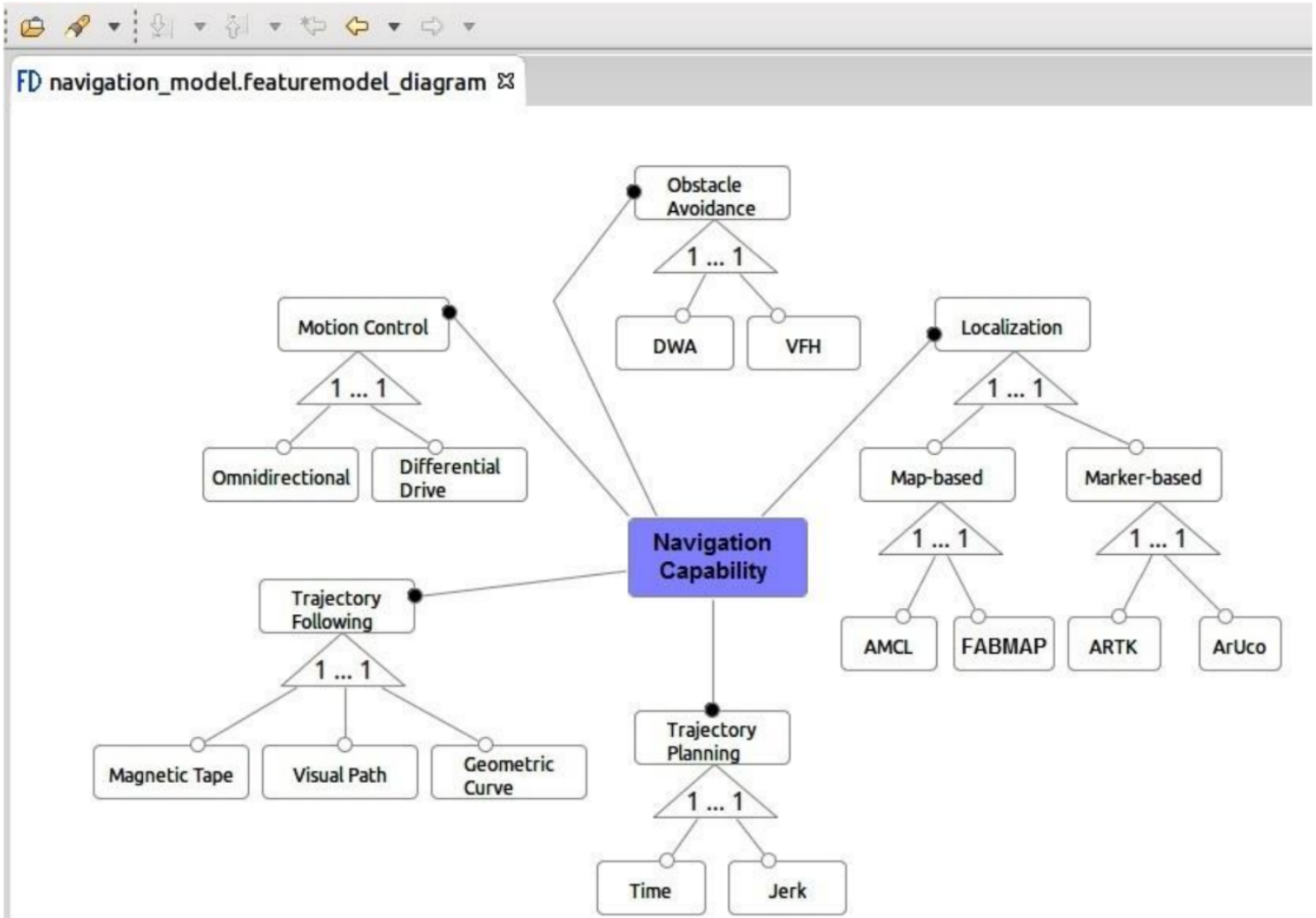
OMG Common Variability Language (CVL) (Late 2010 initial submission)



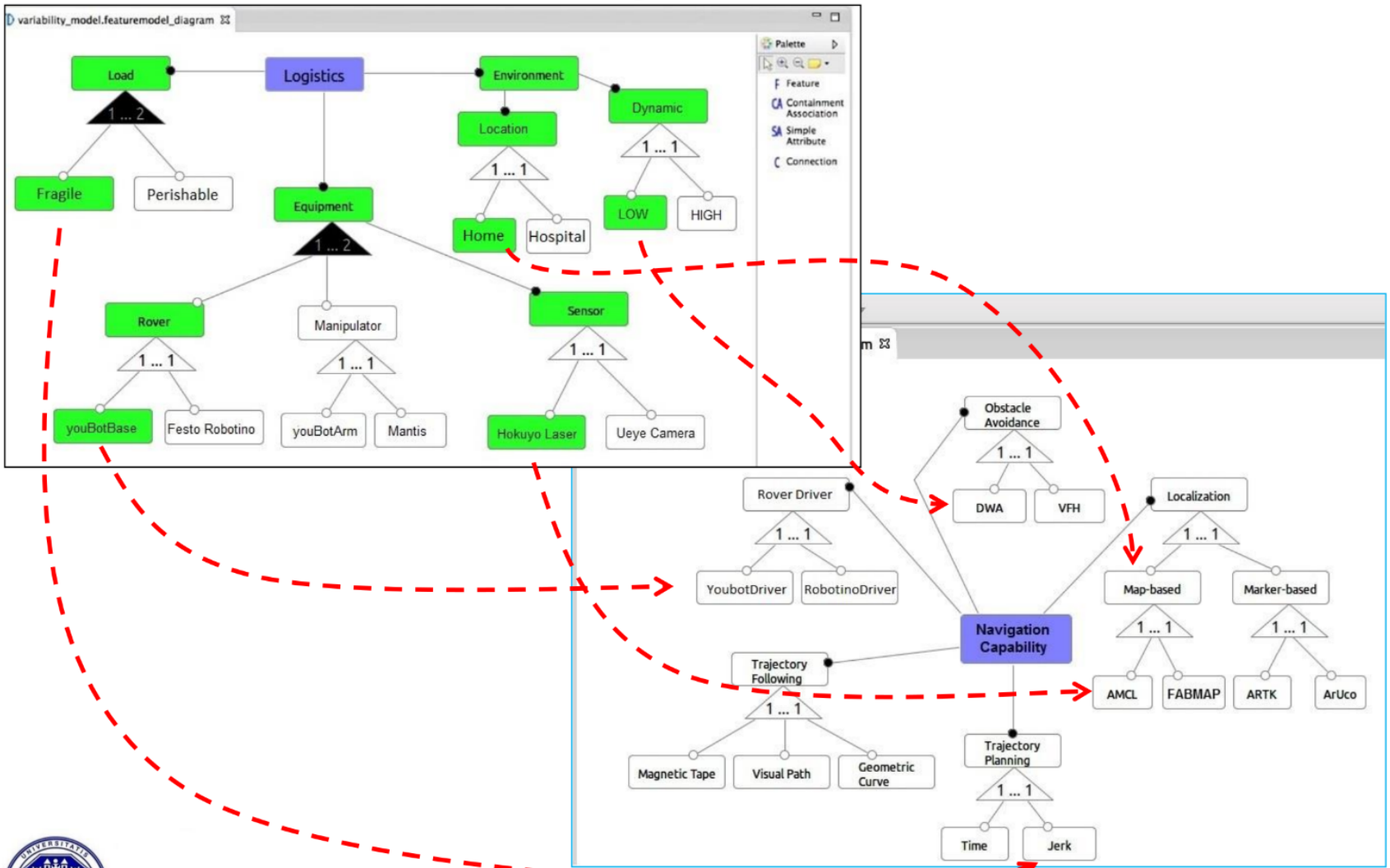
HyperFlex (BRICS project 2009-2012)



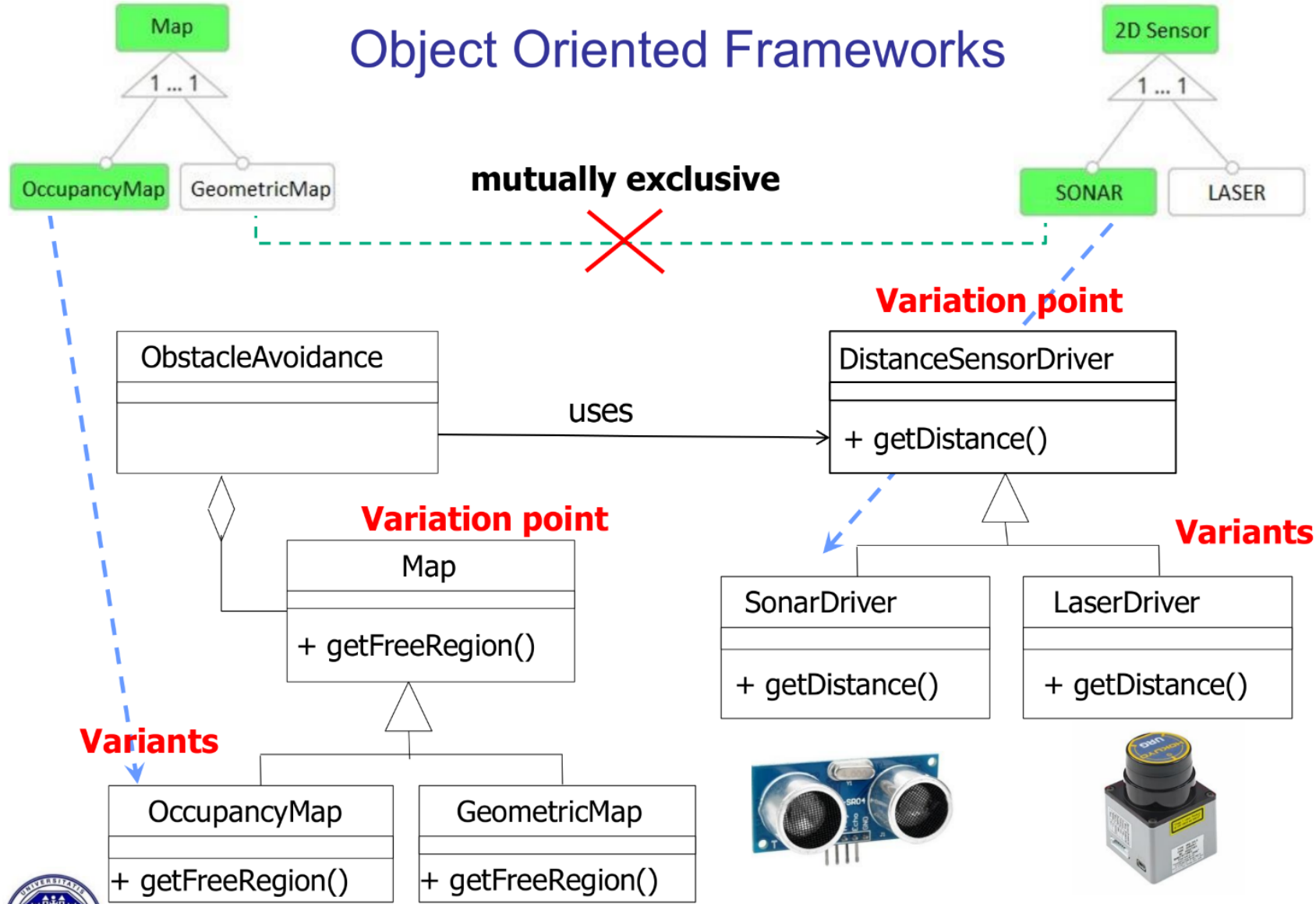
Feature Diagram of Navigation Capability



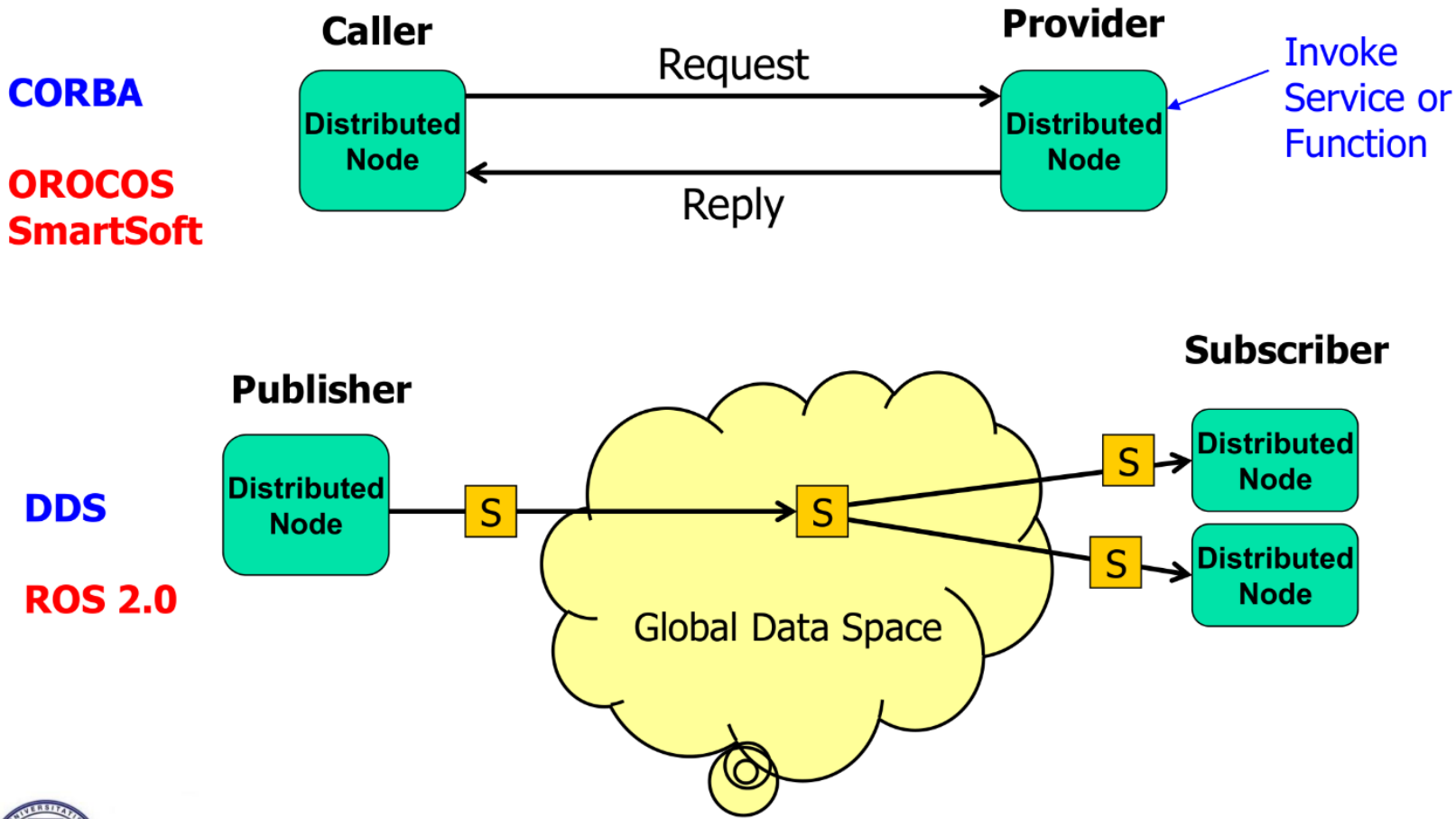
Connecting Feature Models

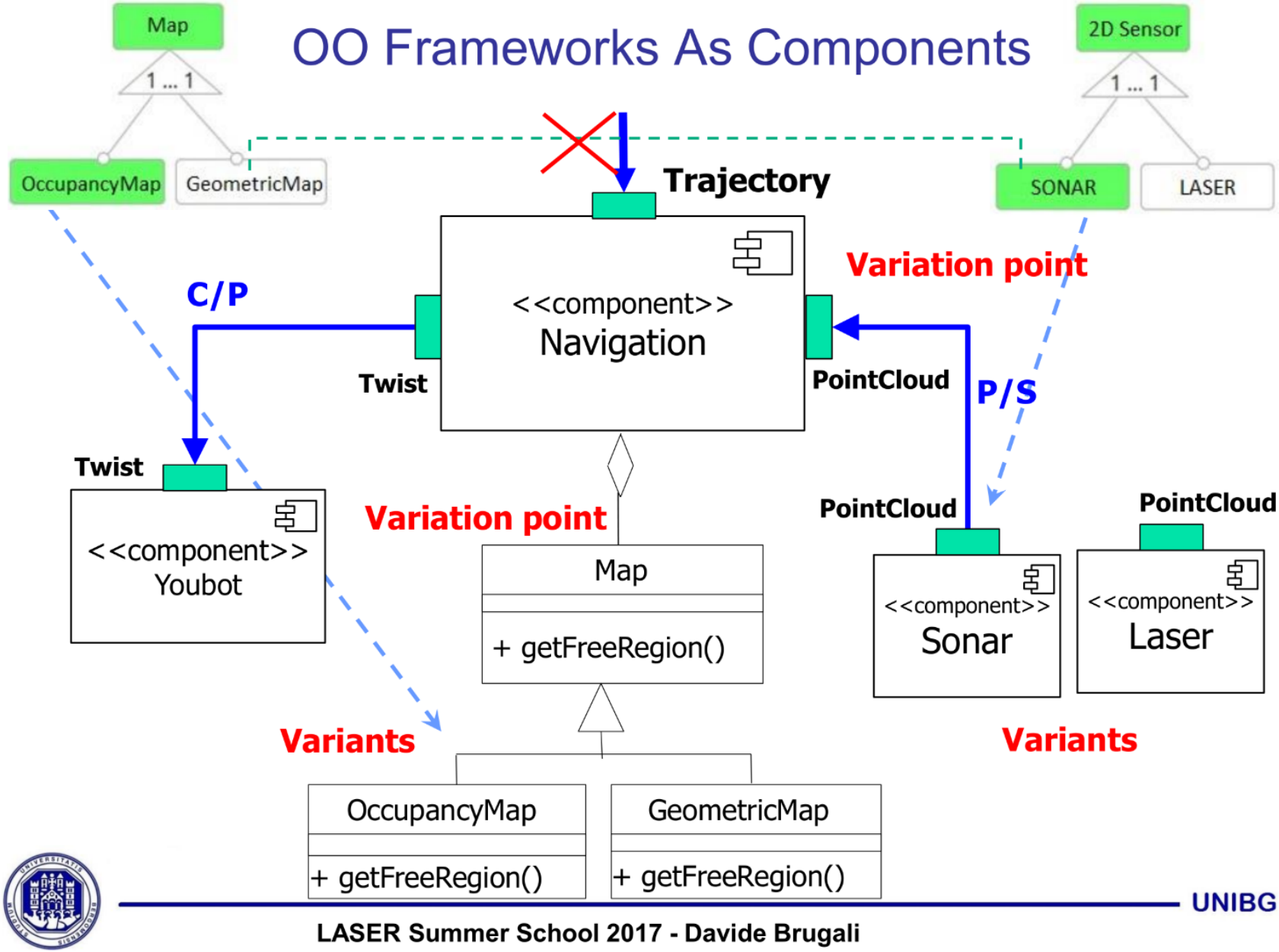


Object Oriented Frameworks



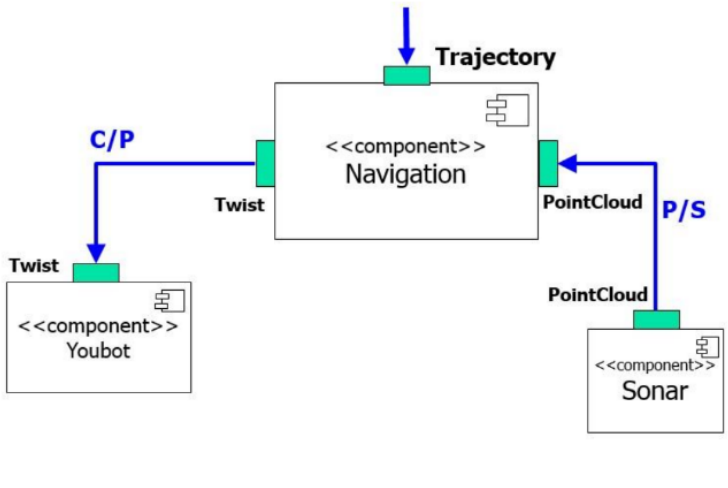
Component-based Distributed Middleware




OO Frameworks As Components
Map
1 ... 1
OccupancyMap
GeometricMap
2D Sensor
1 ... 1
SONAR
LASER
Trajectory
<<component>>
Navigation
C/P
Twist
PointCloud
Variation point
P/S
Twist
<<component>>
Youbot
Variation point
Map
+ getFreeRegion()
PointCloud
PointCloud
<<component>>
Sonar
<<component>>
Laser
Variants
Variants
OccupancyMap
+ getFreeRegion()
GeometricMap
+ getFreeRegion()
UNIBG
LASER Summer School 2017 - Davide Brugali

Architecture Design and Analysis

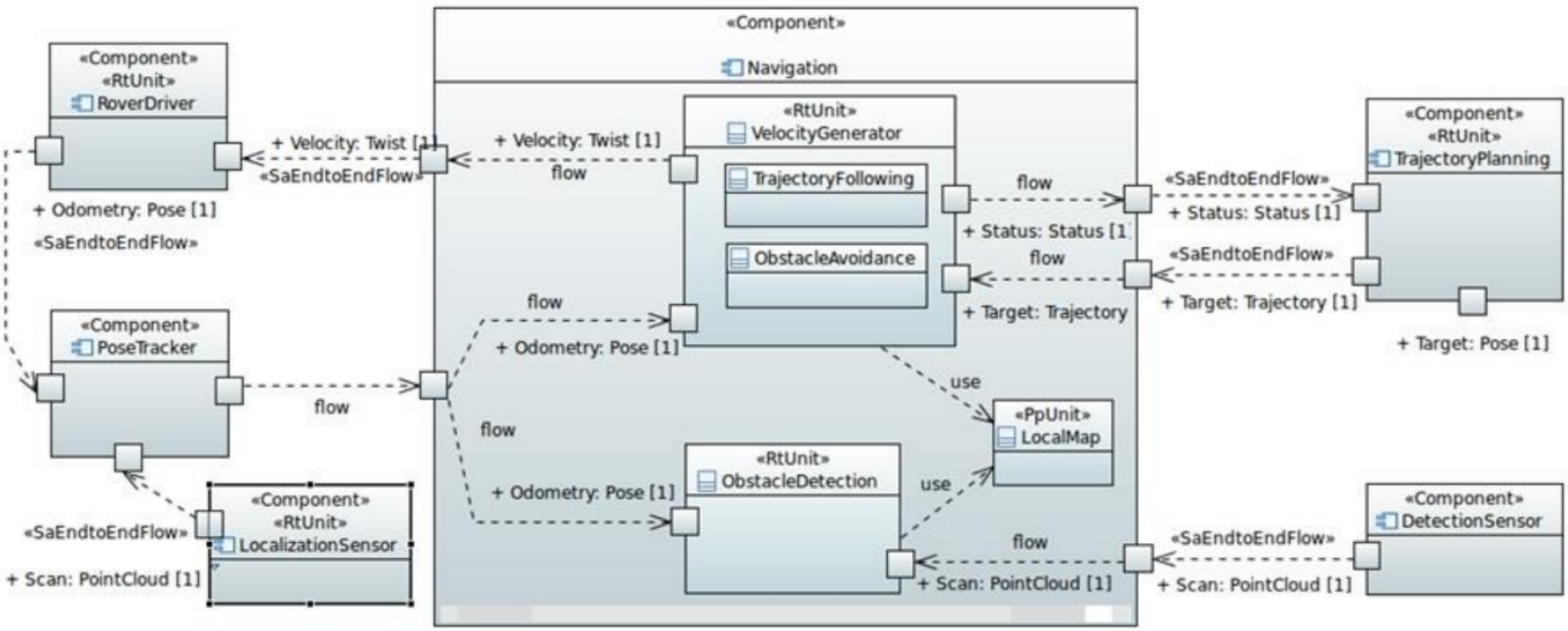
- Architectural Model



- Non Functional Properties
  - Timing constraints
  - Resources capabilities
  - Allocation of functionalities to resources

- Analysis Models
  - Timed Automata
  - Queueing Networks
- System properties
  - Schedulability
  - Response Time
  - Safety
- Analysis Methods
  - Model checking
  - Theorem Proving
  - Rate Monotonic Analysis

UML Modeling and Analysis of RTE Systems (MARTE)



Architecture Analysis and Design Language (AADL)

```
device laser_scanner
  features
    point_cloud: out data port;
  flows
    on_flow_src:flow source point_cloud
                {latency => 5 ms .. 5 ms;};
end laser_scanner;
```

```
process adapt_trajectory
  features
    point_cloud: in data port;
    trajectory: out data port;
  flows
    on_flow_path:flow path point_cloud->traje
                 {latency => 40 ms .. 60 ms;};
  properties
    Period => 100 ms;
end adapt_trajectory;
```

```
process compute_twist
  features
    odometry: in data port;
    trajectory: in data port;
    twist: out data port;
  flows
    on_flow_path:flow path trajectory->twist
                 {latency => 20 ms .. 30 ms;};
  properties
    Period => 50 ms;
end compute_twist;
```

```
device rover
  features
    twist: in data port;
  flows
    on_flow_snk:flow sink twist
                {latency => 10 ms .. 10 ms;};
end rover;
```

SoftBank Robotics

- Funded in 2005 as Aldebaran
- More than 400 employees
- 3 branches : Europe, US, Japan
- Nao since 2007
  - More than 10 000 NAO sold in 70 countries
  - 1 000 labs, universities and high schools
- Pepper since 2014
  - 10000 Peppers sold
  - B2B, B2D, B2A and B2C
- World Leader on the market of Humanoid Robotics

Why humanoid ?

- To have a better understanding of the way humans work
- To interact in a world designed for humans
- To communicate easily with humans through the body language
- To let the use cases possibilities wide open
- For a better acceptability

Research and Showcase in Japan

Toyota Partner

HRP2 - Kawada

HRP4 - Kawada

HRP4C - Kawada

Asimo - Honda

European humanoids

REEM - PAL

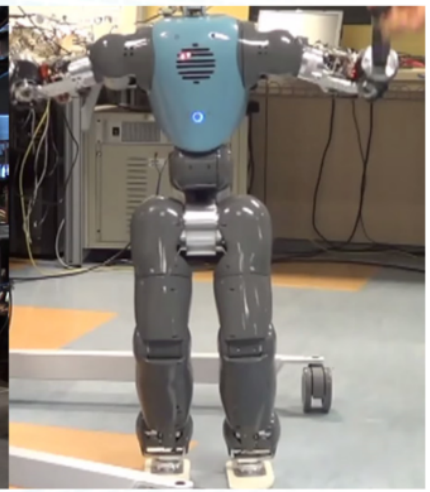
TORO - DLR

Romeo – SoftBank Robotics

iCub - IIT

US humanoids : Boston Dynamics

PETMAN



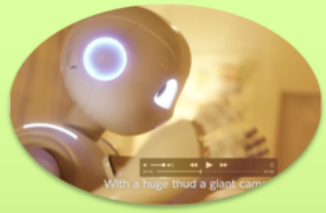
PETMAN Naked

ATLAS

ATLAS Next Generation

Why Robots and Humanoid robots in education ?

Home robot assisted teaching better in ***promoting and improving children's concentration*** (vs Traditional media e.g. books with audiotape-assisted learning, Web-based learning)***. [1]***



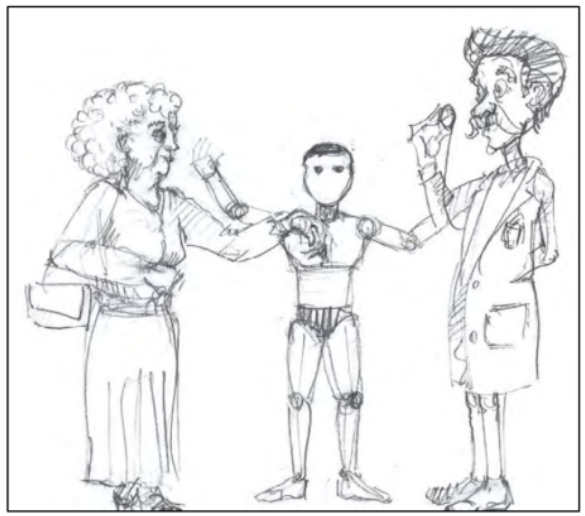
Body language and abilities to provide more human-like social signals; such robots are capable of being more engaging [2]

Children expect the robot to contain human-like features (e.g. a rectangular human like head) [3]

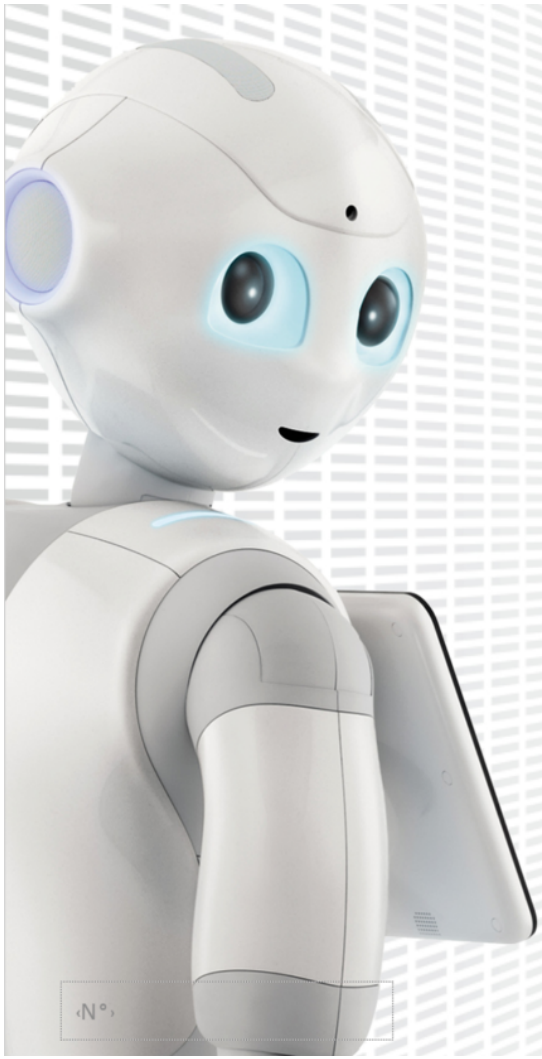
[1] Han, Jeong-Hye et al. "Comparative study on the educational use of home robots for children." Journal of Information Processing Systems 4.4 (2008): 159-168.
[2] Chang, Chih-Wei, et al. (2010). Exploring the Possibility of Using Humanoid Robots as Instructional Tools for Teaching a Second Language in Primary School. Educational Technology & Society 13, no. 2: 13-24.
[3] Obaid, M, et al. 2015. Designing Robotic Teaching Assistants: Interaction Design Students' and Children's Views. International Conference on Social Robotics.

Assistance to Elderly People

- **Improve the safety of the person**
  - Fall detection
  - Unusual behavior detection
  - Assistance for locomotion
  - Detection of objects in the path
  - Medicine
- **Communication and social interaction**
  - Reminder of the meeting of the day
  - Contact with family and friends
  - Connection with new people
  - Cognitive games
  - Dialog
- **Every day life**
  - Assistance to make up
  - Assistance to dress
  - Getting objects from an other room
  - Grasping objects on the ground
  - Physical coaching



PROJET ROMEO2

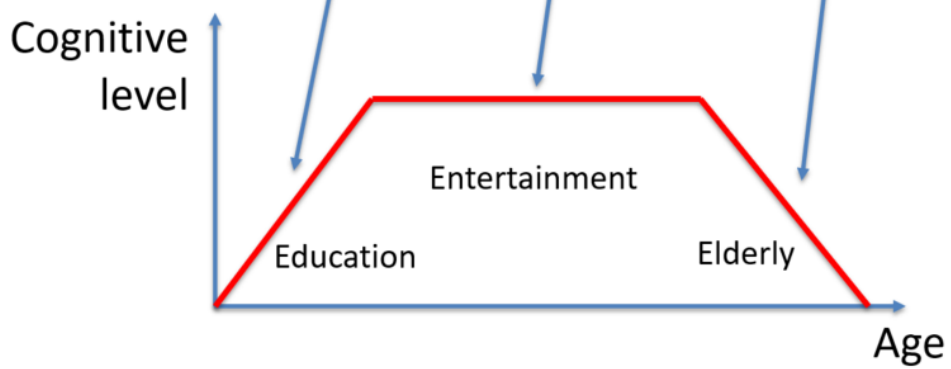


Entertainment

Companion for the family

- **Why do you need a humanoid at home ?**
  - Entertainment
  - Education
  - Care
  - Companionship
  - House keeping
  - Cooking

  } Touchy
- **How much are you ready to pay for this ?**



B2B

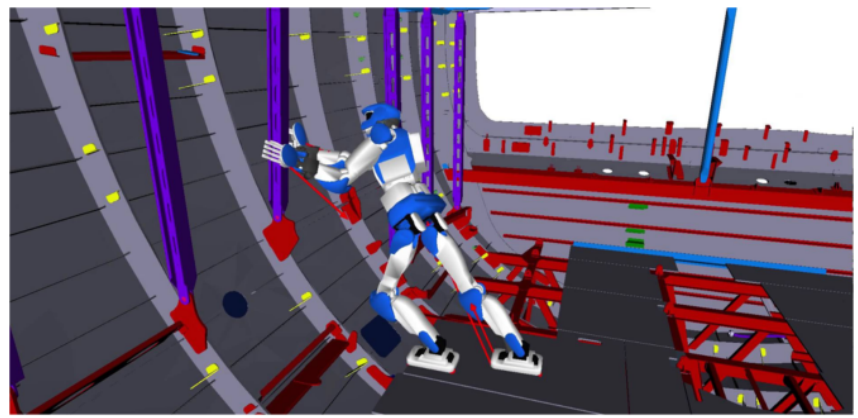
- **Simple applications**
  - Developed by partners
- **Challenges in**
  - Robustness in HRI (cognitive and physical)
  - Connectivity (WIFI, devices, data base,...)
  - Return on investment
- **First results with Pepper**
  - Good acceptability (Japan, Europe mainly)
  - +38% customer visits in SoftBank shops
  - +15% sales in Nescafe shops
  - Second advertising campaign with Renault (+130 Pepper in car dealerships)
- **Go beyond novelty**
- **Careful with social aspects**
  - Replacement of human worker ?
  - Support for human worker ?

Industry

- **Industrial robots**
  - Are not really robots but automaton
  - Are not humanoids
- **They work far away from humans**
- **The workplace is designed for them**
- **Things are changing**
  - More versatile robots
  - More interaction with humans
  - In constrained workspace

Next stage

- **Use fully the humanoid shape**
  - Not only for friendliness
  - But for mobility performances
- Only a man can do the task, the robot has to be humanoid
- **Comanoid project**
  - H2020 project
  - In collaboration with AIST
  - Deploy humanoid robots in Airbus assembly operation
  - Inaccessible for wheeled platforms
  - Management of multi-contact



Exoskeleton

- **Applications**
  - Super soldier
  - Assistance for carrying heavy tools
  - Rehabilitation
- **Man Machine Interface**
  - Muscle activity sensing
  - Force control : intention detection
- Several projects all over the world

RB 3D
we multiply your capacity

“What is it for ?” : an open question for longtime

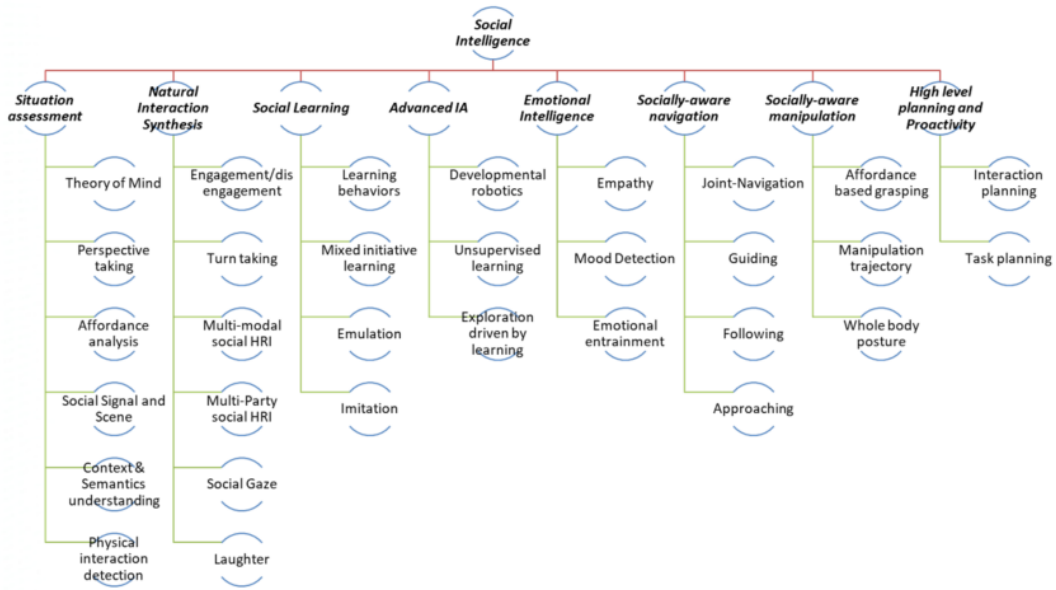
- **The ultimate humanoid robot will (perhaps) be application-independent**
- **In the meantime, think about the use case before designing**
  - **Consider the robot as a development platform**
  - **Consider the connection to external devices and external services**
- **We probably can’t imagine today the future usages of humanoid robots**
  - **Remember personal computers and mobile phones**

Сложности физического взаимодействия гуманоидных роботов

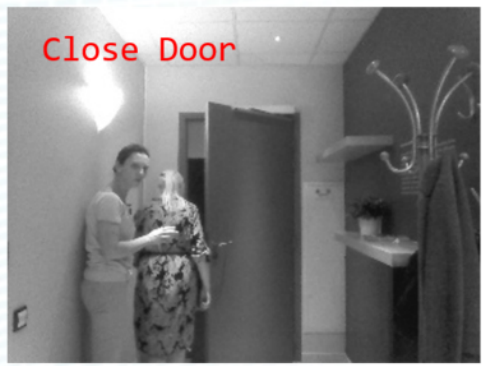
- Избежание самостолкновений
- Проблема приоритетов параллельно исполняемых задач
- Динамический выбор регуляторов для движения
- Избежание препятствий
- Навигация
- SLAM
- Захват объектов
- Управление силой, обратная связь

Cognitive Interaction

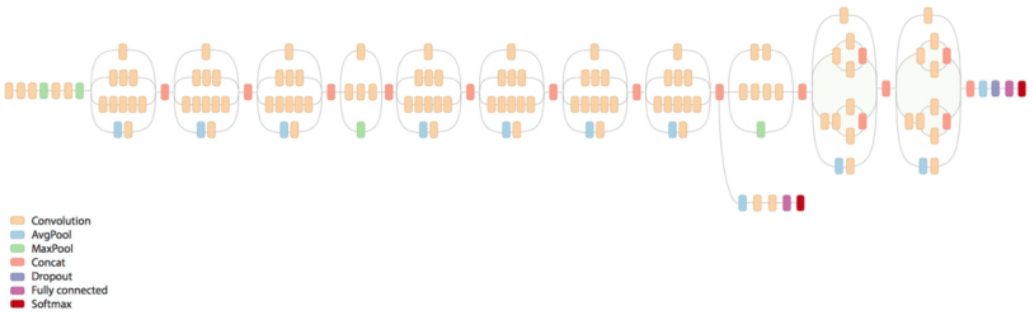
- Exchange of information between the robot and the user
- Also called Social Interaction
- Based on multimodal communication
  - Speech
    - Verbal
    - non verbal
  - Body language
    - Gesture
    - Position
- Objective
  - Fulfill a task
  - Improve the acceptability of the robot
  - Provide the right service at the right time

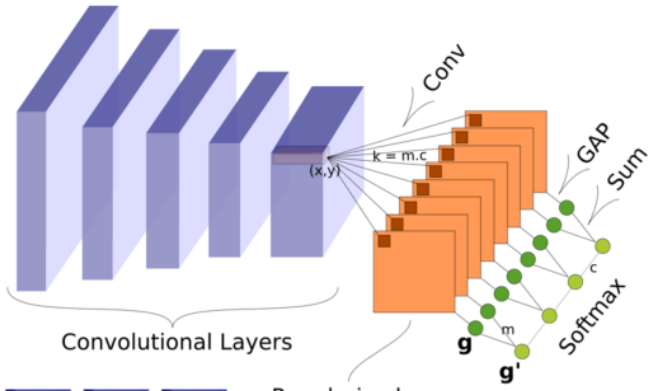


Activity recognition on video

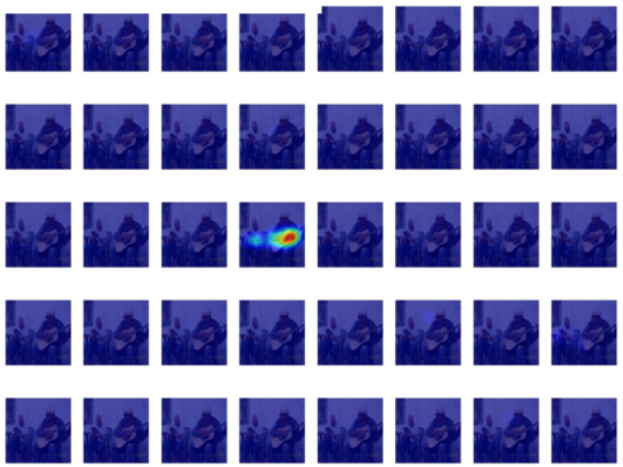


Object recognition for activity recognition





- Based on Inception V3 (Google)
  - Recognition of 1000 classes of objects
  - Trained on Imagenet (1M images)
- Replacement of 3 higher levels
- Training on Action 40 (Stanford)
  - 100 images for each class for training
  - Labels: drink, read, cook, run, ...



PROJET ROMEO2

LAAS-CNRS

Dialogue engine

- **Static tree structure**
  - Different choices for a question
  - Different expressions for the same meaning (yes, yeah, ok,…)
- **Dynamic parameters**
  - Information collected during the dialog or from vision (Mrs Smith, Mr Jones,..)
- **Connection with actions**
  - Orders : Stand-up, look at me, raise you arm
  - Body language : waving, …

Ethical, legal and social issues of robotics

- **Very trendy topic**
- **Roboethics**
  - Salvini, P., Laschi, C., & Dario, P. (2010). Design for acceptability: improving robots' coexistence in human society. *International journal of social robotics*, 2(4), 451-460.
  - Veruggio, G., Operto, F., & Bekey, G. (2016). Roboethics: Social and ethical implications. In *Springer handbook of robotics* (pp. 2135-2160). Springer International Publishing.
  - Sharkey, A., & Sharkey, N. (2012). Granny and the robots: ethical issues in robot care for the elderly. *Ethics and Information Technology*, 14(1), 27-40.
- **Has to be mentioned in any proposal**

Emotion expression

- Based on body language
- Everything is scripted by an animator
- It changes the level of communication
- Risks
  - I can forget it is a machine
  - I can develop excessive attachment
  - Loss of the social links
- Prevention
  - The robot looks like a machine
  - It can prevent too long periods of interaction
  - The robot can solicit social connections

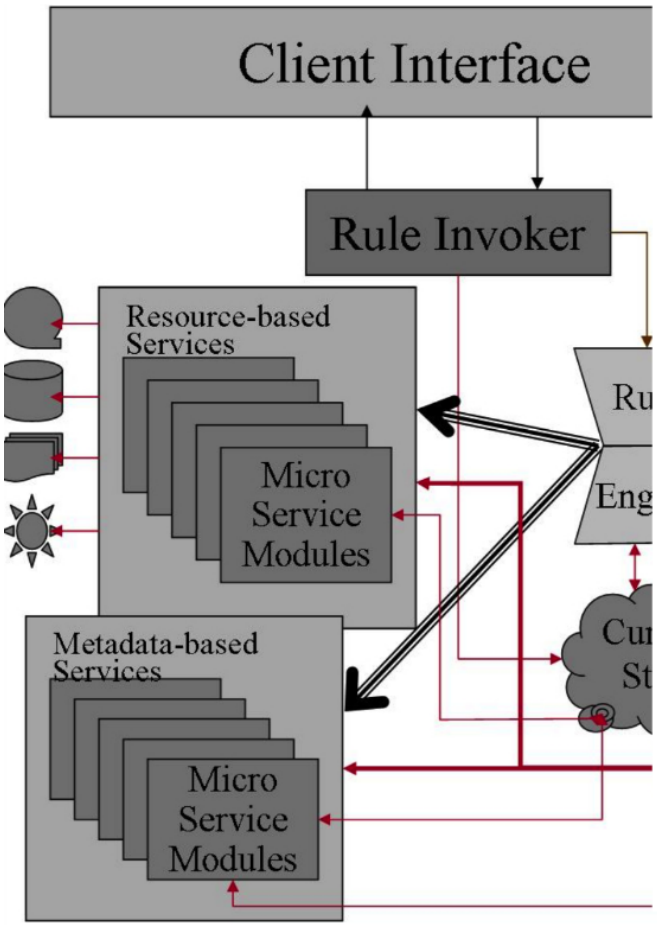
Responsibility of the robot manufacturer

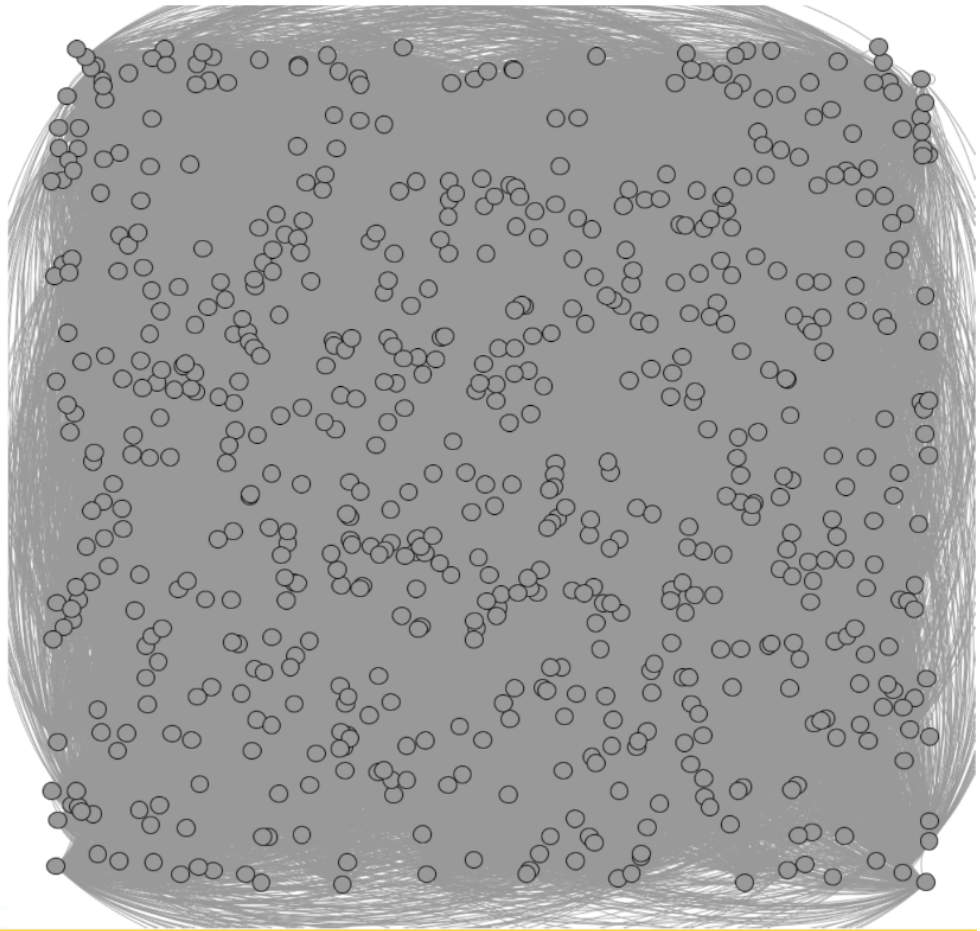
- **Guarantee of the good functioning of the robot**
  - Specifications of the basic functions (not the applications)
  - Warnings for the user
  - Test, test, test...
- **Technical solutions**
  - Safety features (force control, auto-diagnosis...)
  - Code certification
  - Protection against hacking
- **The manufacturer cannot be responsible**
  - For the applications downloaded on its robot
  - For the training by the final user

Ashish Kapoor

- Не выложил слайды!
- AirSim (`https://github.com/microsoft/airsim`)
- Рассказывал в основном про обеспечение безопасности полёта квадрокоптера путём задания ограничений в виде вероятностных темпоральных логик

Blueprint *vs. Product*



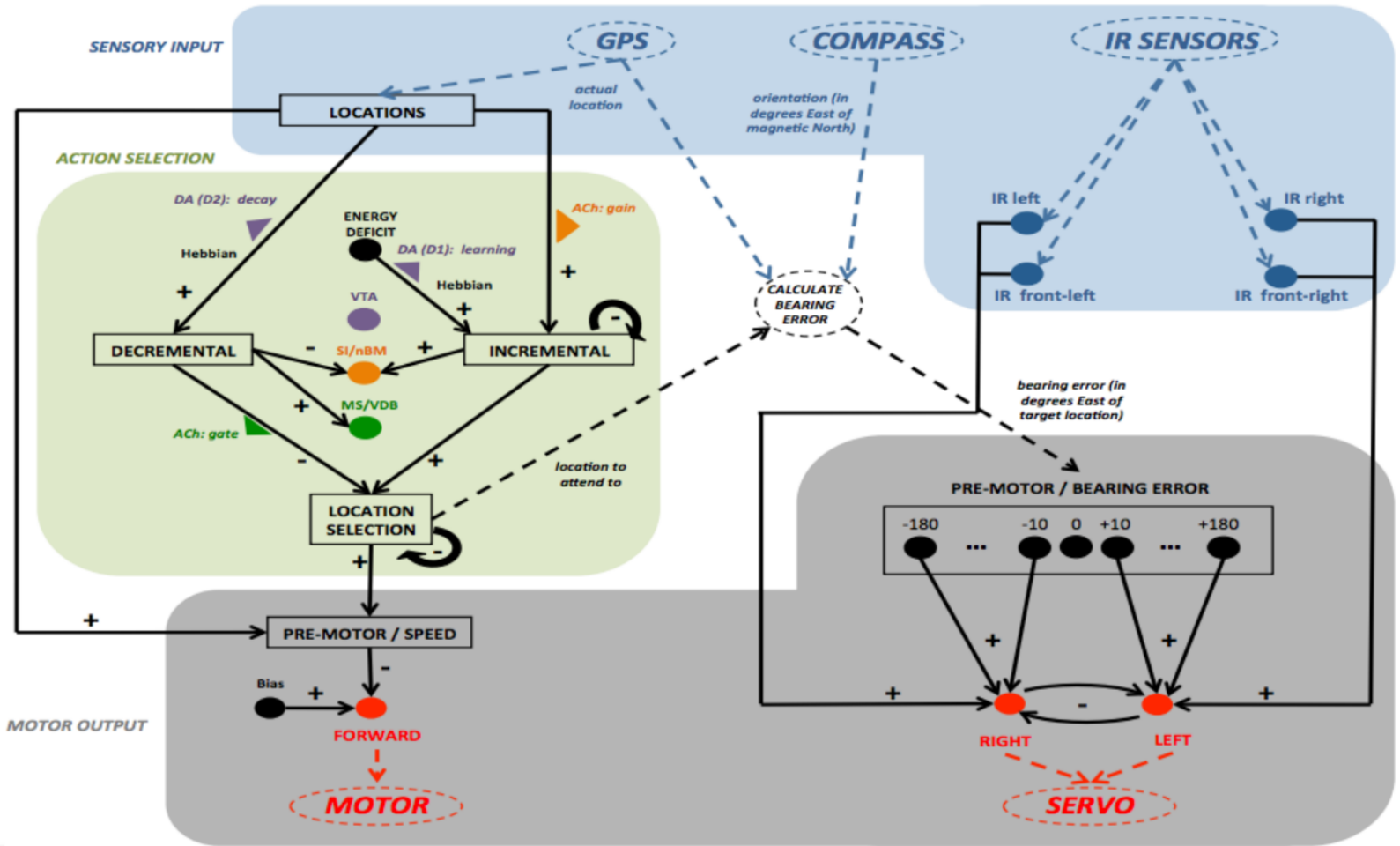


USC Viterbi School of Engineering — University of Southern California

Example: Android S/W Stack

- "Android" is only the top of the stack
- Below Android is
  - HAL
  - Linux kernel
  - Device drivers

Neural Architecture for Android Robots



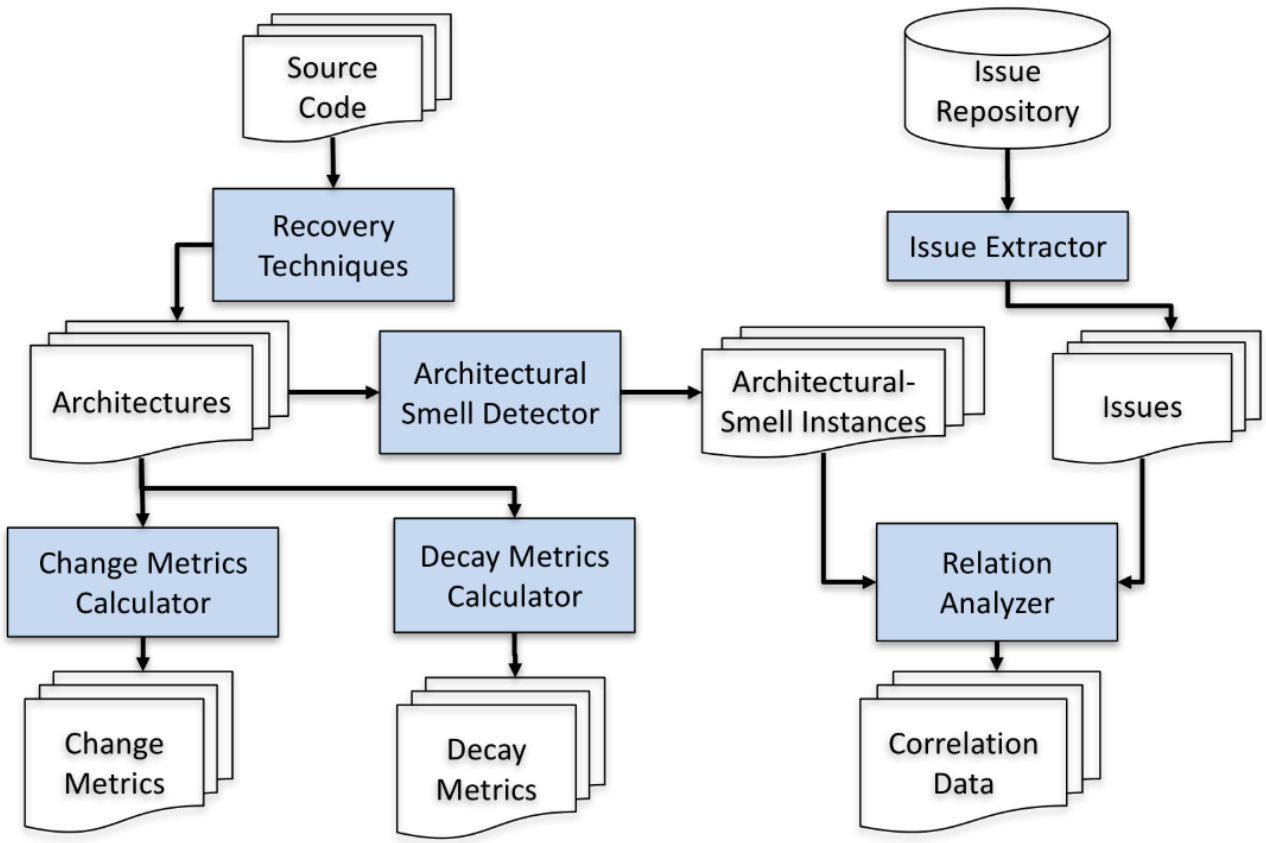
USC Viterbi School of Engineering — University of Southern California

Automated Recovery Solutions

- ACDC – Algorithm for Comprehension-Driven Clustering
  - Structural pattern-based clustering
- ARC – Architecture Recovery Using Concerns
  - Concern-based hierarchical clustering based on similarity measure
- Bunch-NAHC & Bunch-SAHC
  - Hill-climbing algorithm for maximizing *Modularization Quality*
- LIMBO – scaLable InforMation BOttleneck
  - Probabilistic hierarchical clustering
- WCA-UE & WCA-UENM – Weigted Combined Algorithm
  - Dependency-based hierarchical clustering
- ZBR – Zone-Based Recovery
  - Hierarchical clustering based on textual information
- PKG – Implementation Package Structure

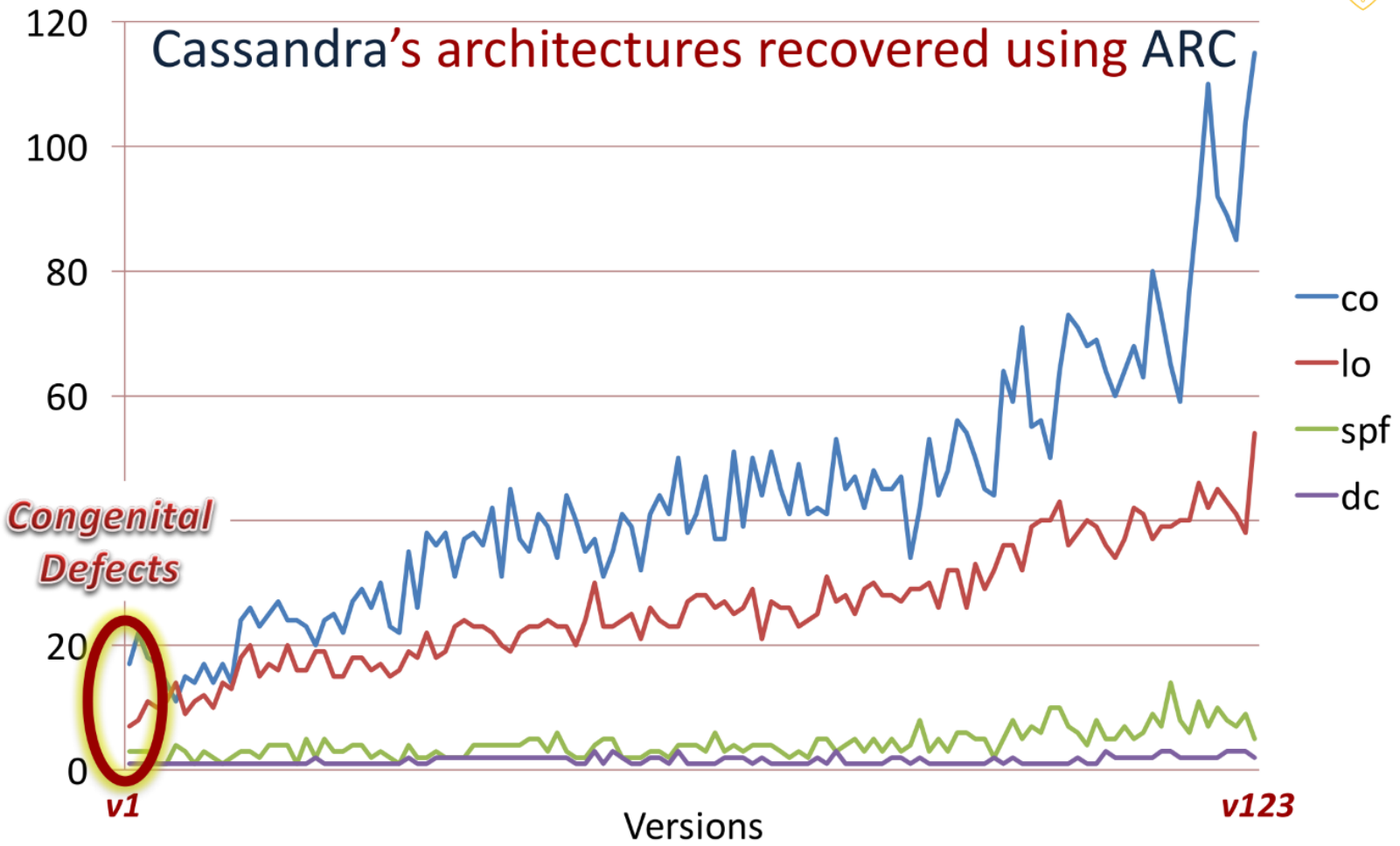
Software Architecture ARCADE

Architecture Recovery, Change, and Decay Evaluator

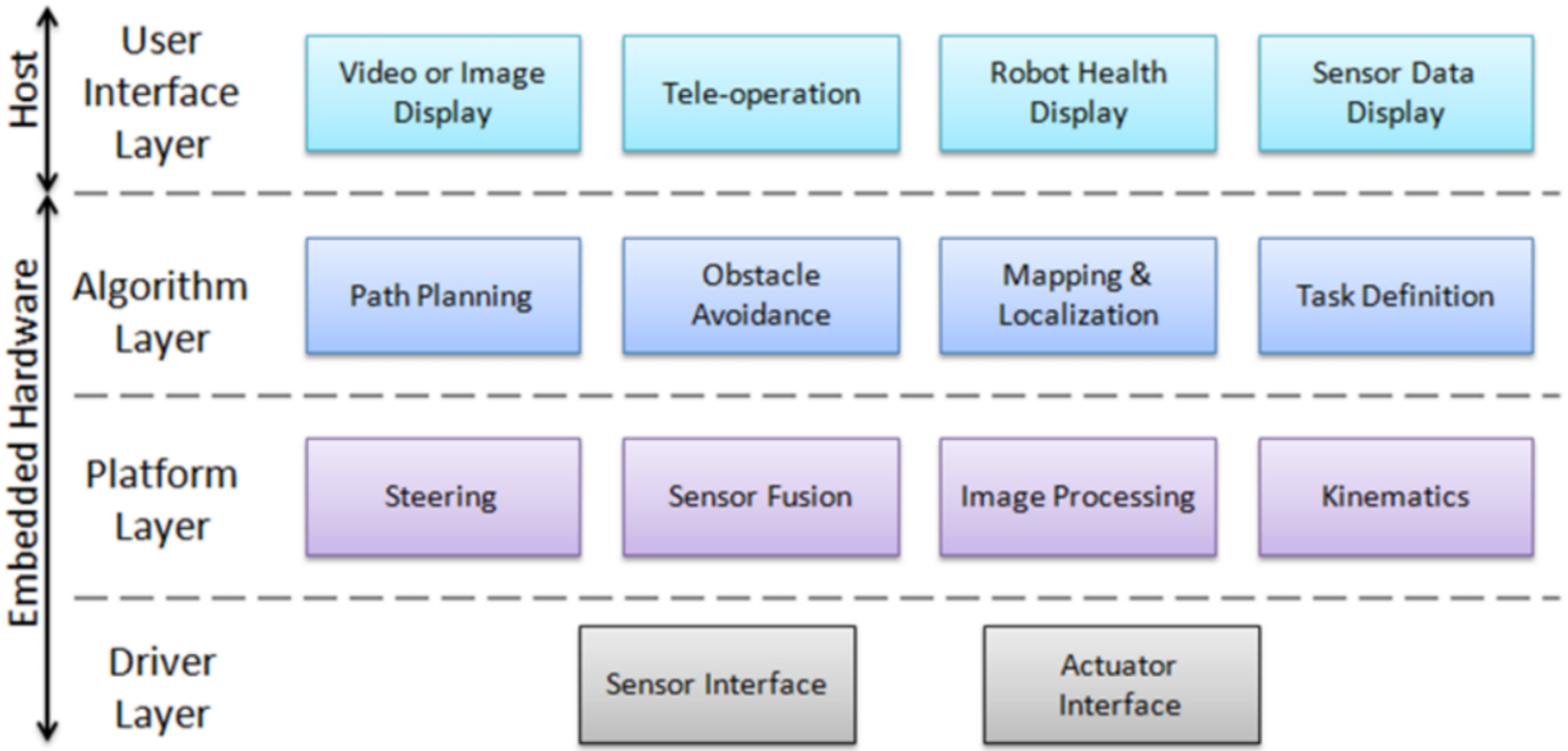


USC Viterbi School of Engineering — University of Southern California

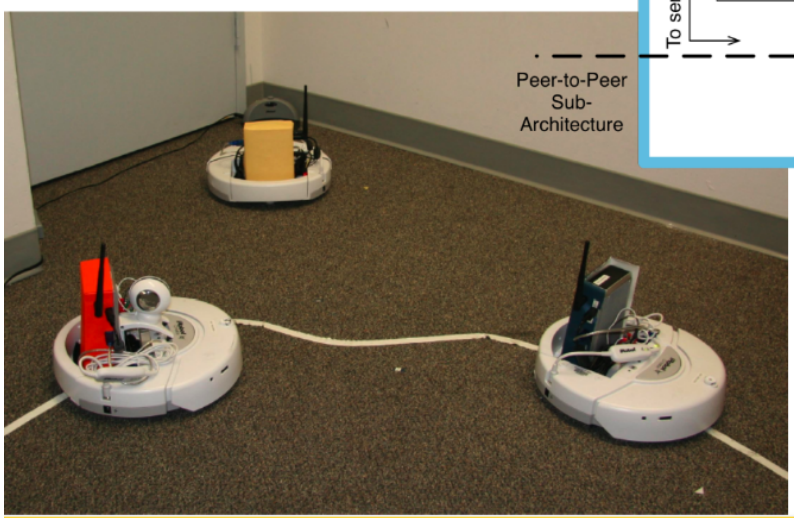
Architectural Decay

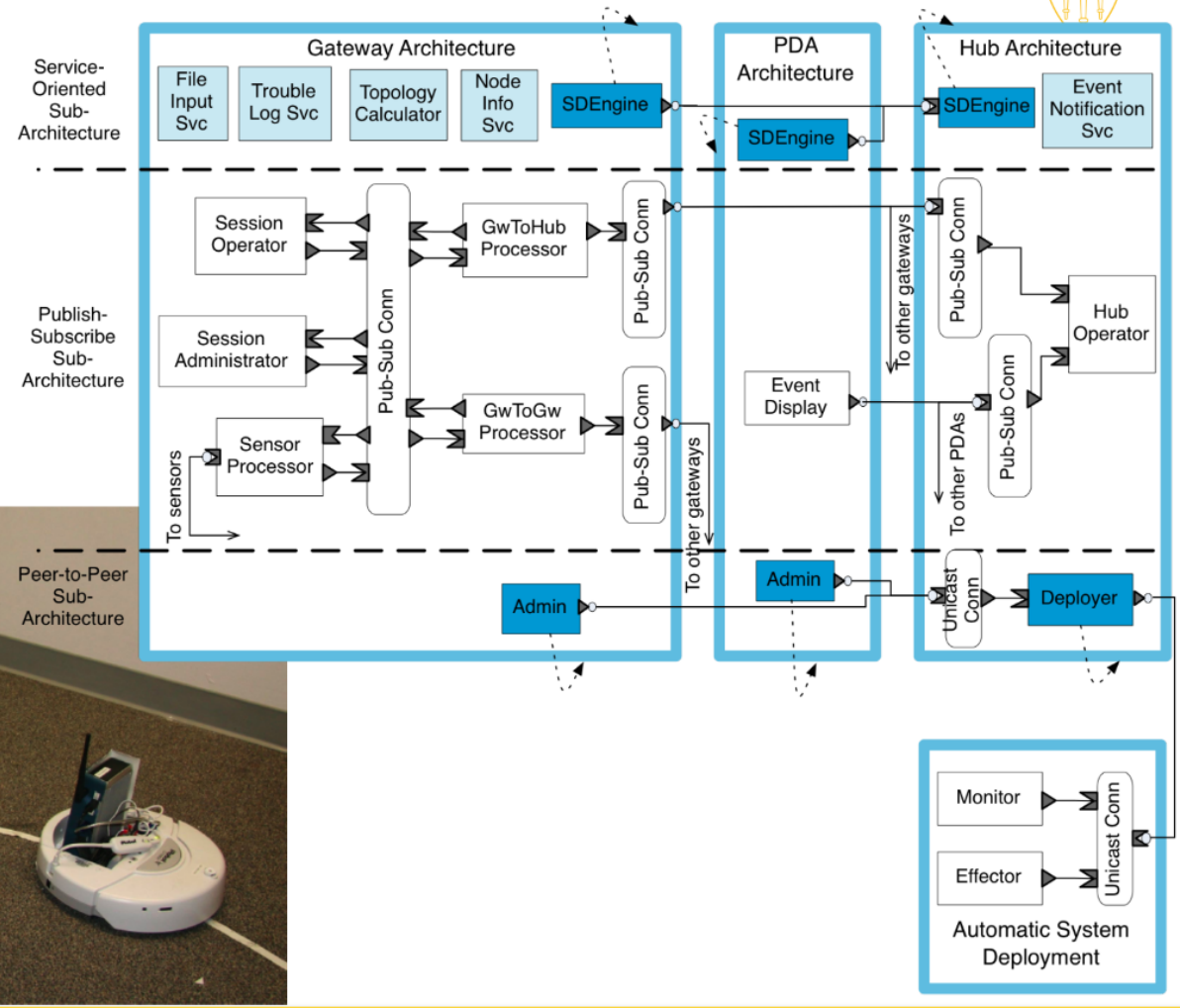


Robotics DSSA



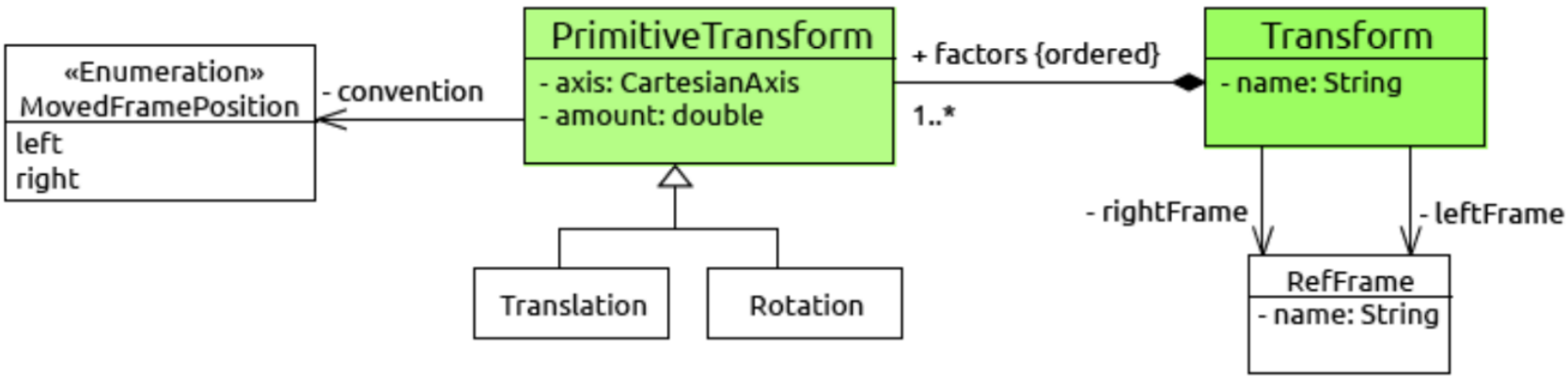
Styles in Action - MIDAS



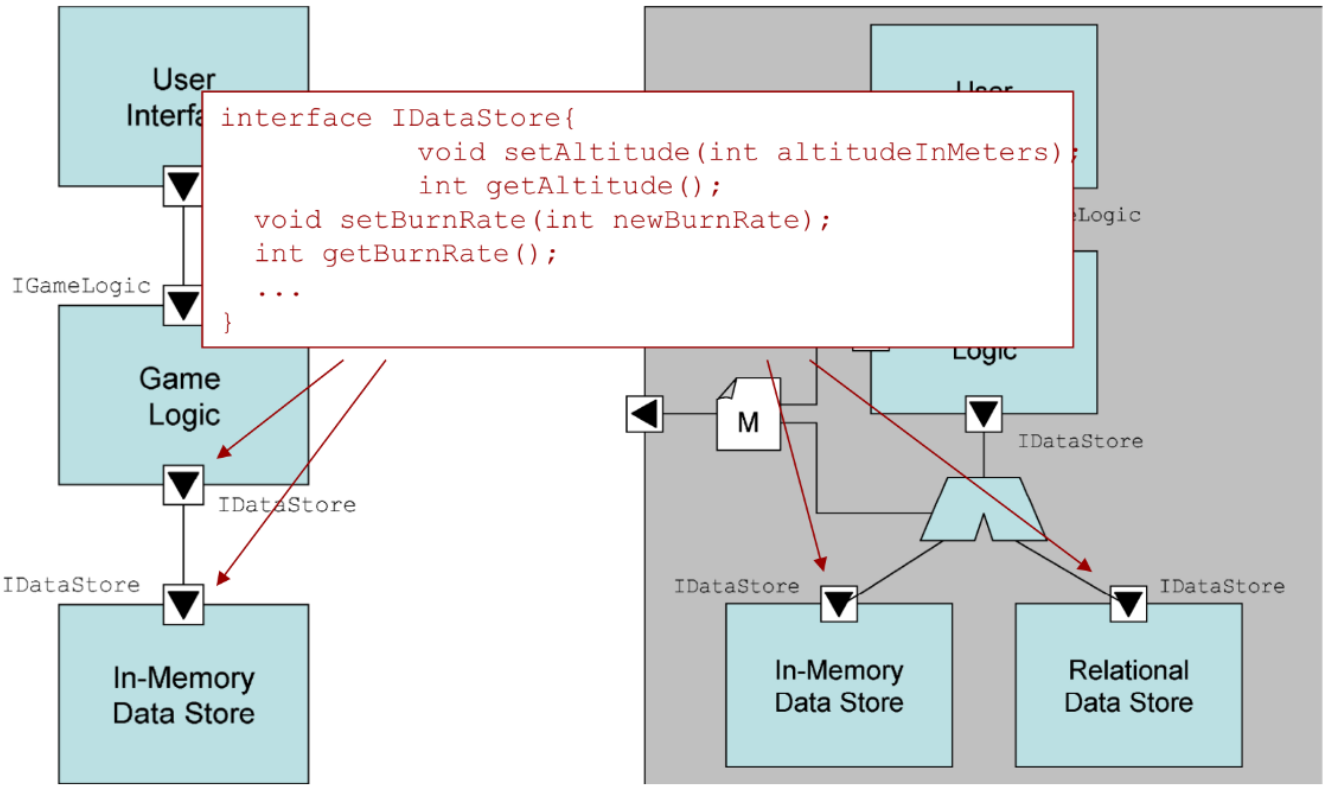


Use of UML in Robotics

RobCoGen Coordinate Transforms



Koala Example



Single system

Product line of two systems

AADL Example

```
data lander_state_dat
end lander_state_data
bus lan_bus_type
end lan_bus_type;

bus implementation la
properties
  Transmission_Time =
  Allowed_Message_Siz
end lan_bus_type.ethe
system calculation_ty
features
  network : requires
            lan_bus.
  request_get     : ou
  response_get    : in
  request_store   : ou
  response_store  : in
end calculation_type;

system implementation
subcomponents
  the_calculation_pro
                process
  the_calculation_pro
                calcula
```

```
connections
  bus access network -> the_calculation_processor.network;
  event d
  event p
  event d
propertie
  Actual_
end calcu
processor
features
  network
end calcu
process c
features
  request
  respons
  request
  response_store : in event port;
end calculation_process_type;
```

```
thread calculation_thread_type
features
  request_get    : out event port;
  response_get   : in event data port lander_state_data;
  request_store  : out event data port lander_state_data;
  response_store : in event port;
properties
  Dispatch_Protocol => periodic;
end calculation_thread_type;
process implementation calculation_process_type.one_thread
subcomponents
  calculation_thread : thread client_thread_type;
connections
  event data port response_get ->
              calculation_thread.response_get;
  event port calculation_thread.request_get -> request_get;
  event port response_store ->
              calculation_thread.response_store;
  event data port request_store -> request_store;
properties
  Dispatch_Protocol => Periodic;
  Period => 20 ms;
end calculation_process_type.one_thread;
```

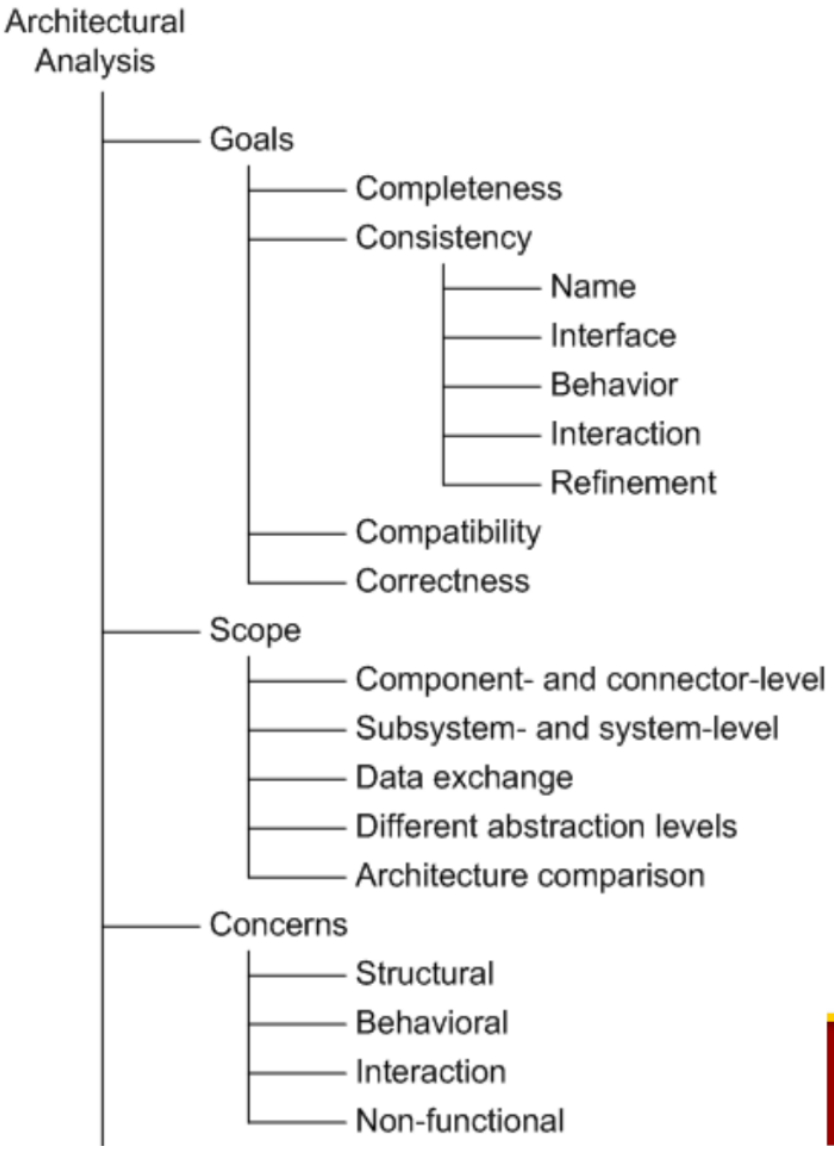
USC Viterbi School of Engineering — University of Southern California

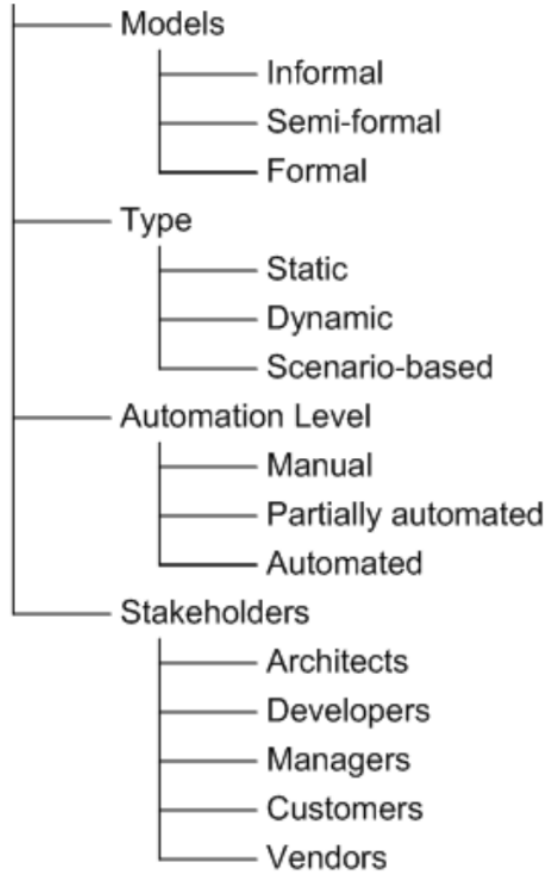
Formal Models and Analysis

```
thread calculation_thread_type
features
  request_get    : out event port;
  response_get   : in event data port lander_state_data;
  request_store  : out event data port lander_state_data;
  response_store : in event port;
properties
  Dispatch_Protocol => periodic;
end calculation_thread_type;
process implementation calculation_process_type.one_thread
subcomponents
  calculation_thread : thread client_thread_type;
connections
  event data port response_get ->
              calculation_thread.response_get;
  event port calculation_thread.request_get -> request_get;
  event port response_store ->
              calculation_thread.response_store;
  event data port request_store -> request_store;
properties
  Dispatch_Protocol => Periodic;
  Period => 20 ms;
end calculation_process_type.one_thread;
```

- Helps architects determine component composability
- Helps developers with implementation-level decisions
- Helps with locating and selecting OTS components
- Helps with automated code generation
- Not as useful to non-technical stakeholders

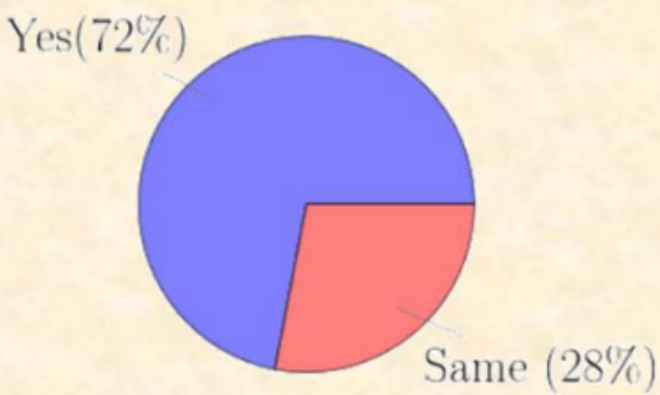
In a Nutshell



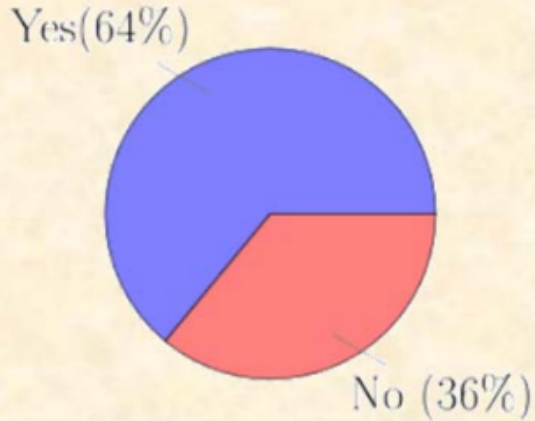


University of Southern California

Concurrency in robotics (Rusakov, 2016)



(a) Do you find concurrent programming more difficult than sequential programming?



(b) Have you come across any robotics application where concurrency would have been beneficial but was not implemented?

4

SCOOP background

Simple Concurrent Object-Oriented Programming

First version described in CACM article (1993) and chapter 32 of *Object-Oriented Software Construction*, 2nd edition, 1997

Prototype implementation at ETH (2005-2010)

Production implementation at Eiffel Software now part of EiffelStudio

Recent descriptions:

- Nienaltowski's ETH PhD, 2007
- Morandi, West, Nanz, Meyer, ETH PhD theses and papers, 2011-2014

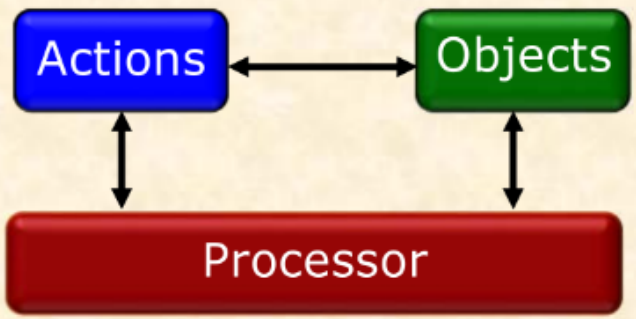
20

Processors in SCOOP

Processor: Thread of control supporting sequential execution of instructions on one or more objects

Can be implemented as:

- Computer CPU
- Process
- Thread



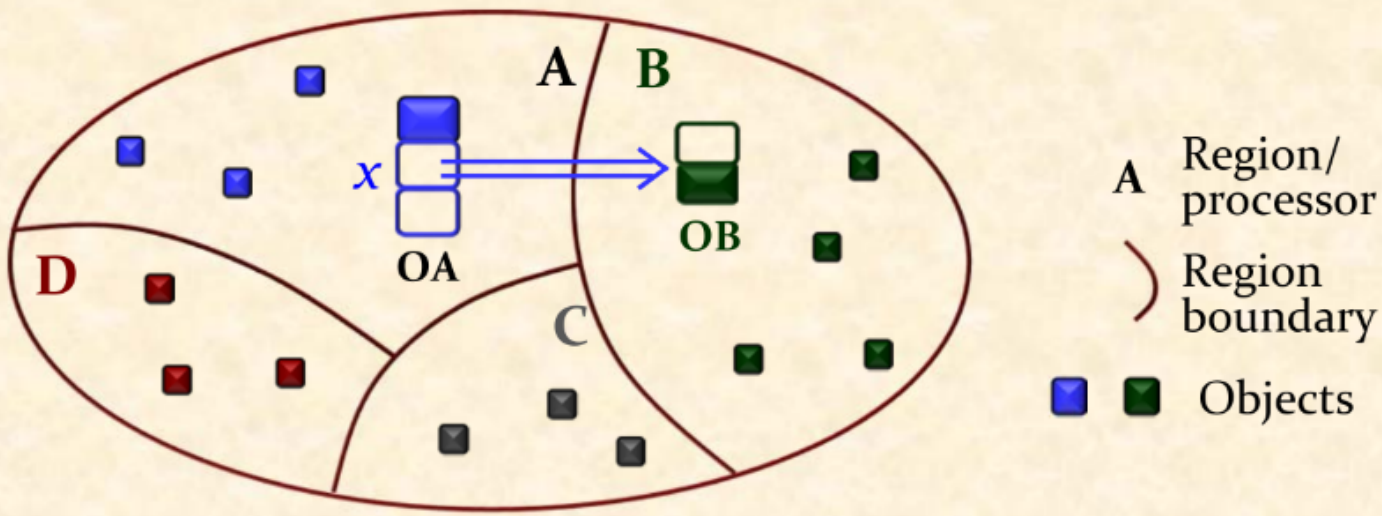
Will be mapped to computational resources

24

Regions

Objects are partitioned into regions

Each region has its processor



26

Lazy wait (or: wait by necessity)

How do we resynchronize after asynchronous (separate) call?

x.command1 (u, v)

x.command2 (a, b)

x.command3

...

value := x.query1

Answer: the client will wait when, and only when, it needs to

SCOOP resynchronizes only on *queries*

Reminder:

- A *command* does not return a result (procedure).
- A *query* returns a result (function or attribute).

40

Mutual exclusion in SCOOP

A call

x.r (*a*1, *a*2, ...)

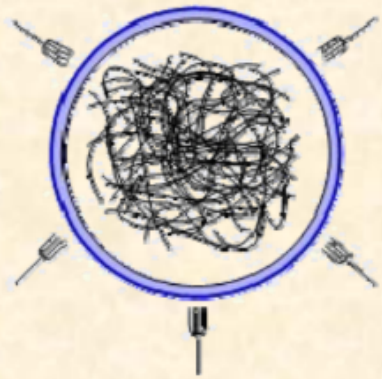
will *wait* until it has been able to lock *all* the separate objects associated with the arguments *a*1, *a*2, ...

Guarantees mutual exclusion

Applies to locking *any number of objects*

43

Dining philosophers in SCOOP



```
class PHILOSOPHER feature
        live
                do
                        from getup until is_over loop
                         think ; eat (left, right)
                        end
                end

        eat ( l, r : separate FORK )
                        -- Eat, having grabbed l and r.
                do ... l.pick ; ... r.pick ;  ... end

        getup do ... end
        is_over :  BOOLEAN
end
```

44

Condition synchronization in SCOOP

- Condition synchronization is provided in SCOOP by reinterpreting routine *preconditions* as *wait conditions*.
- This means that the execution of the body of a routine is delayed until its separate preconditions are satisfied
- A *separate precondition* is a precondition that involves a call to a separate target.

```
put (buf : separate QUEUE[INTEGER] ; v : INTEGER)
            -- Store v into buffer.
        require
            not buf.is_full
            v > 0
        do
            buf.put (v)
        ensure
            not buf.is_empty
        end
```

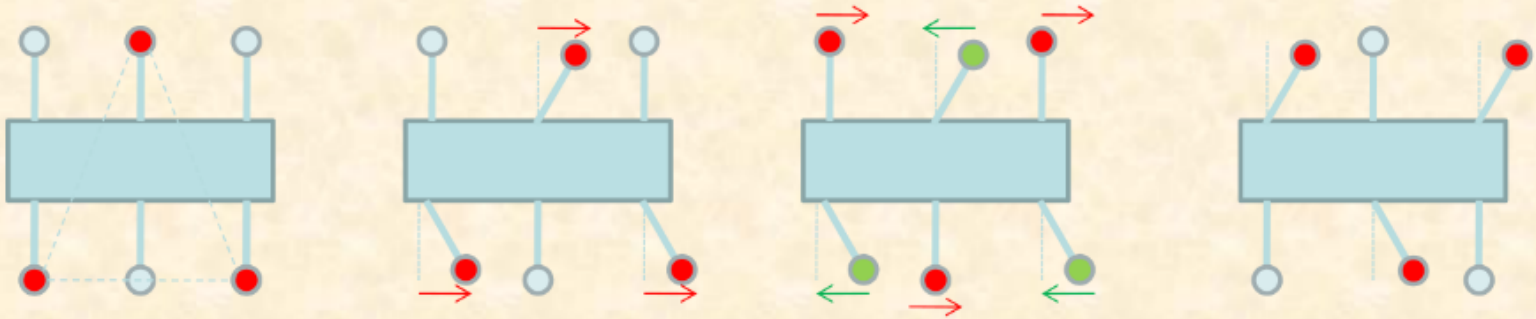
Correctness condition (no wait semantics)

Precondition becomes **wait condition**

51

Hexapod locomotion

Ganesh Ramanathan, Benjamin Morandi, IROS 2011



Alternating protraction and retraction of tripod pairs

- Begin protraction only if partner legs are down
- Depress legs only if partner legs have retracted
- Begin retraction when partner legs are up

53

SCOOP version

begin_protraction (*partner*, *me*: **separate** *LEG_GROUP*)

require

me.*legs_retracted*

partner.*legs_down*

not *partner*.*protraction_pending*

do

tripod.*lift*

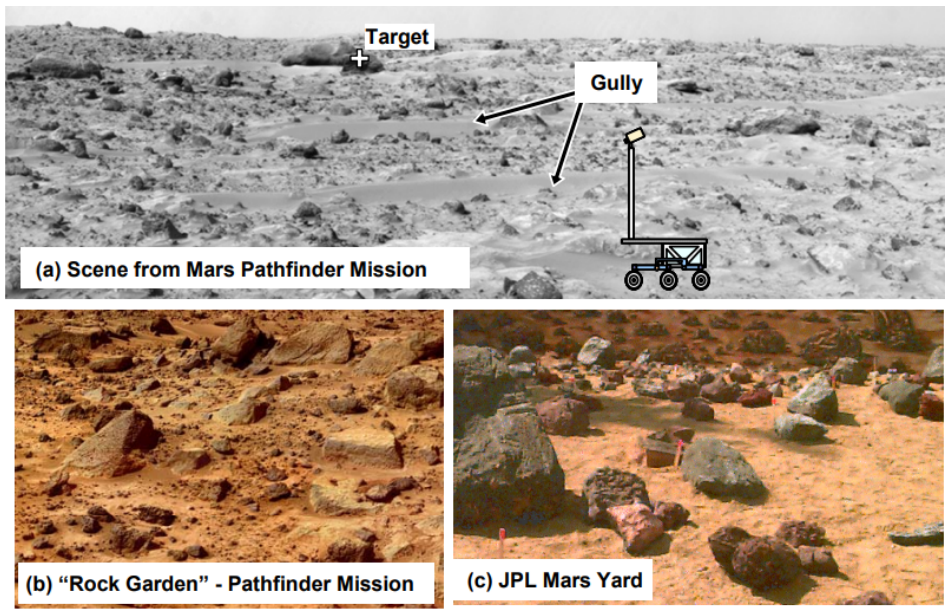
me.*set_protraction_pending*

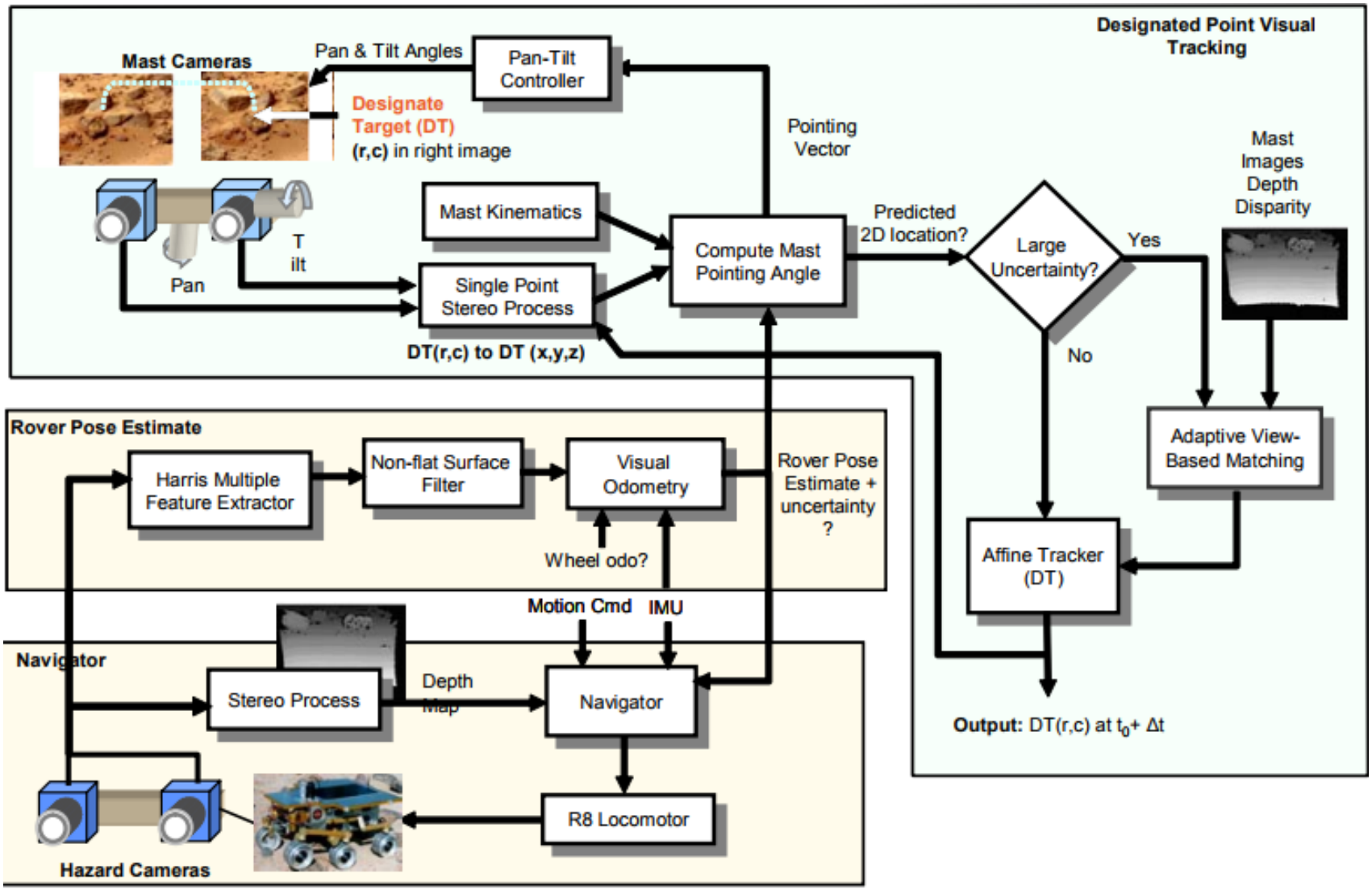
end

55

Issa Nesnas

- Не выложил слайды!
- Рассказывал в основном про навигацию мобильных роботов (роверов) и про архитектуру бортового ПО

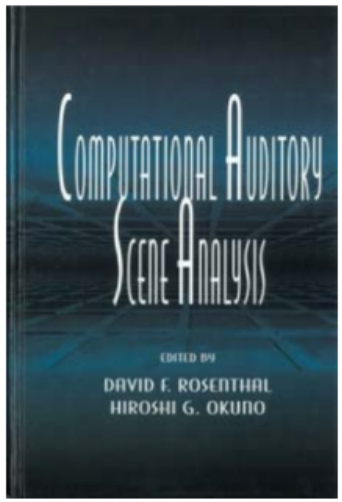



Designated Point Visual Tracking
Mast Cameras
Pan & Tilt Angles
Pan-Tilt Controller
Designate Target (DT) (r,c) in right image
Pointing Vector
Mast Images Depth Disparity
Mast Kinematics
Compute Mast Pointing Angle
Predicted 2D location?
Large Uncertainty?
Yes
No
Pan
Tilt
Single Point Stereo Process
DT(r,c) to DT (x,y,z)
Rover Pose Estimate
Harris Multiple Feature Extractor
Non-flat Surface Filter
Visual Odometry
Rover Pose Estimate + uncertainty ?
Wheel odo?
Adaptive View-Based Matching
Affine Tracker (DT)
Motion Cmd
IMU
Navigator
Stereo Process
Depth Map
Navigator
R8 Locomotor
Hazard Cameras
Output: DT(r,c) at t_0+ Δt

Computational Auditory Scene Analysis (CASA)

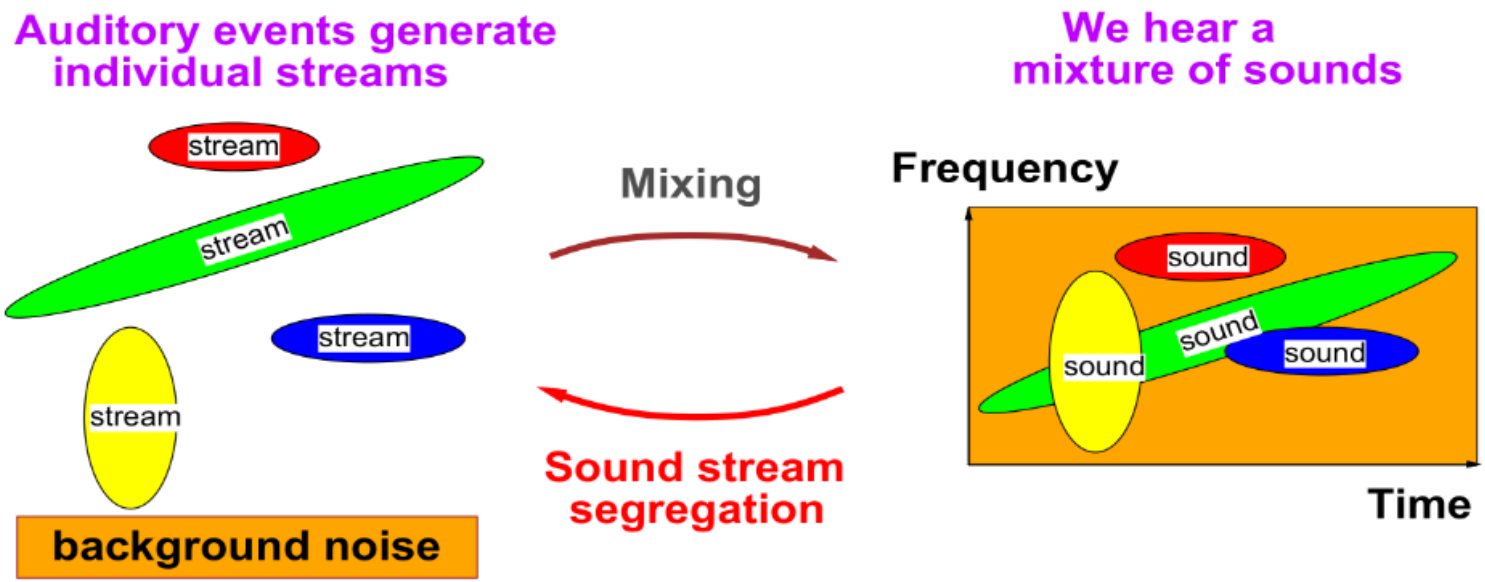
CASA aims to recognize & understand (auditory) environments by sounds. *cf.* (visual) scene analysis

- Computer modeling of hearing and listening
- Recognize general sounds including speech, music, environmental sounds and their mixture
- Three Basic functions are required for CASA
  1. **Sound source localization (SSL)**
  2. **Sound source separation (SSR),**
  3. **Separated sound recognition (ASR)**
     speech/music/onomatopoeia recognition
- Robot audition is a typical application of CASA.



Our Dream: Making all the robots in the world Prince-Shotoku robots

How do we hear sounds?

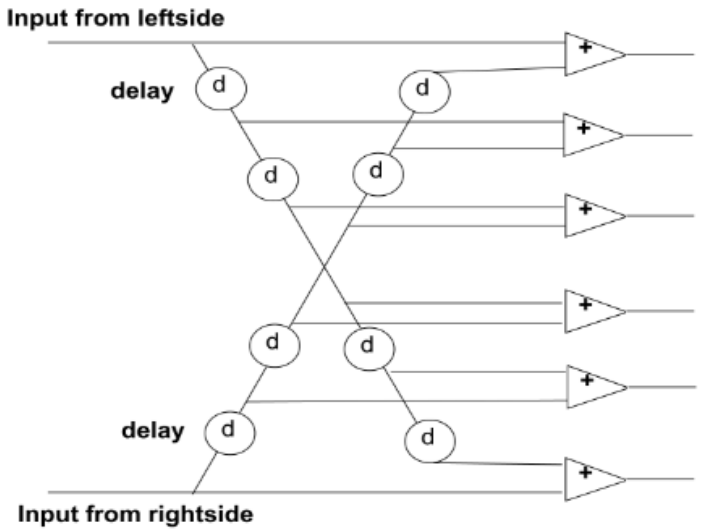


We understand "*we hear a mixture of sounds*"

as "*we hear a mixture of* ***sound streams***."

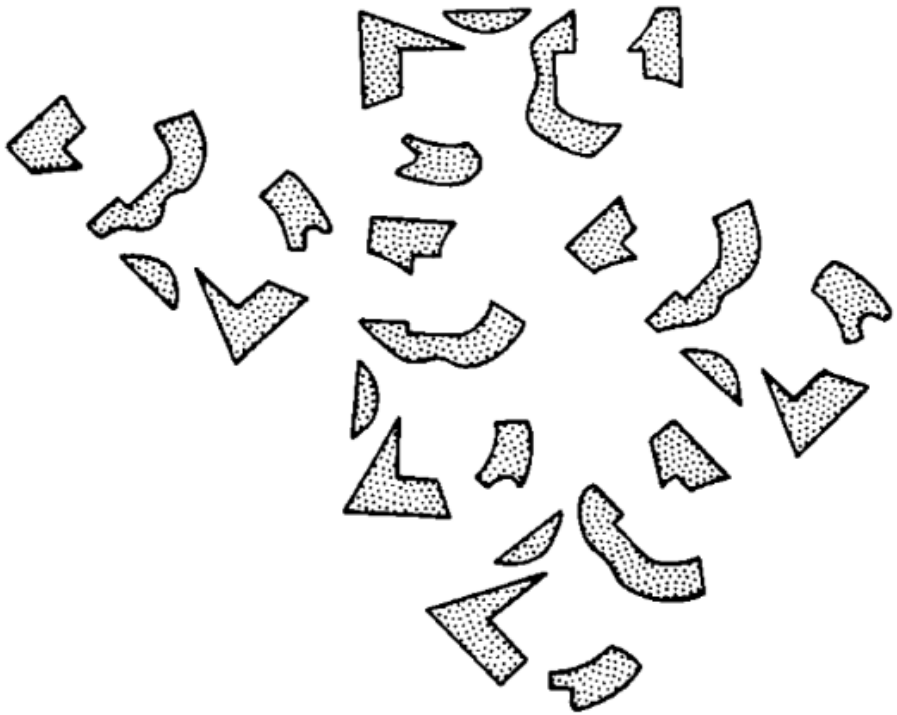
Jeffress' Model of Localization

- Interaural Time Difference (ITD), Delay Time of Arrival (DTOA)
  - ITD (Interaural Time Difference)
  - IPD (Interaural Phase Difference)
  - ILD (Interaural Level Difference)
  - IAD (Interaural Amplitude Difference)
  - IID (Interaural Intensity Difference)



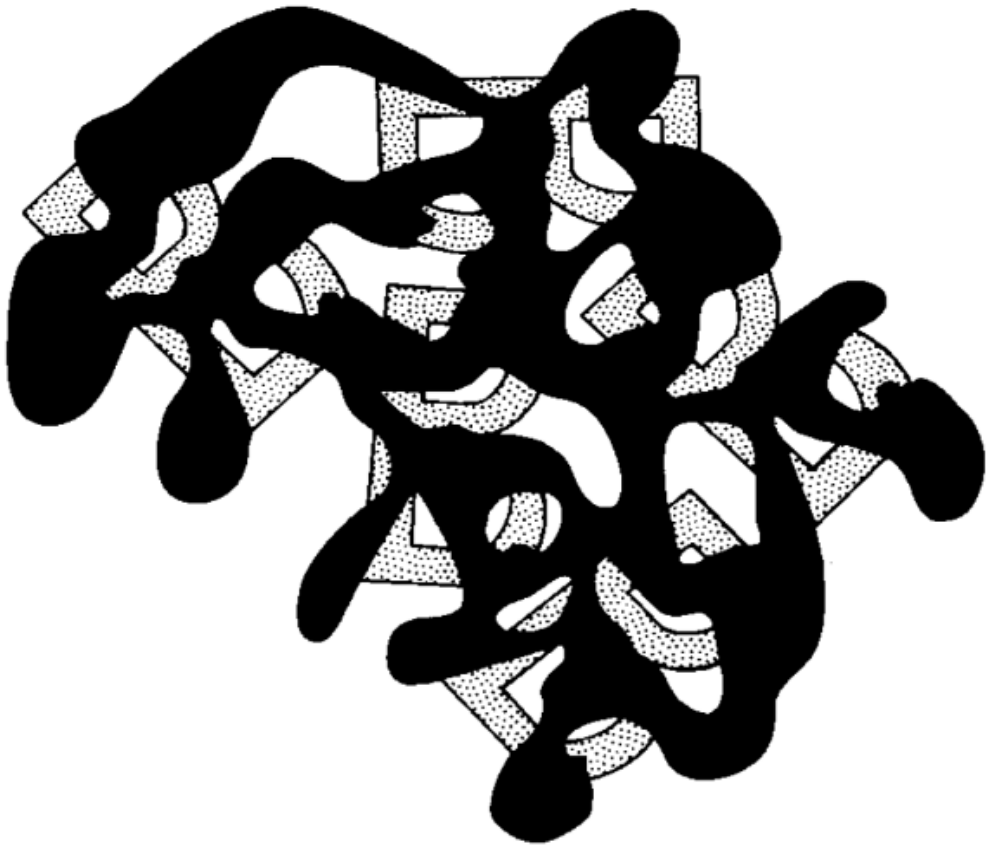
- How do we identify a pair of sounds?
  - *Similar to corresponding points in stereo vision*
- How does the shape of head affect localization?
  - *Binaural audition instead of stereo audition*

Gestalt Principle of Closure



What is the same capital letter ?

Gestalt Principle of Closure

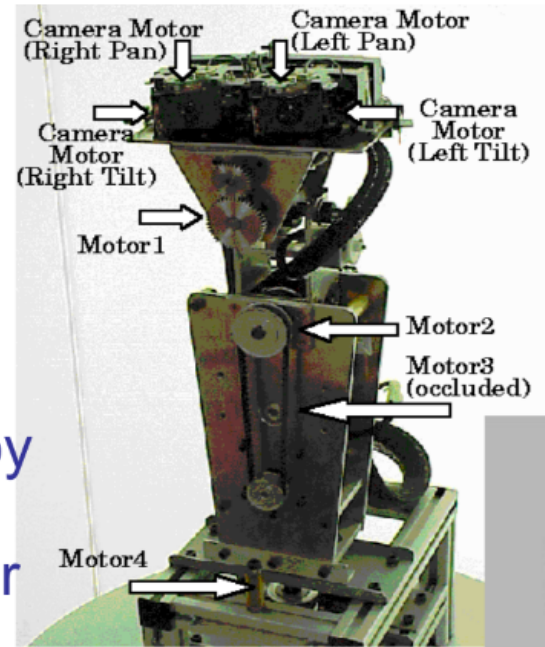


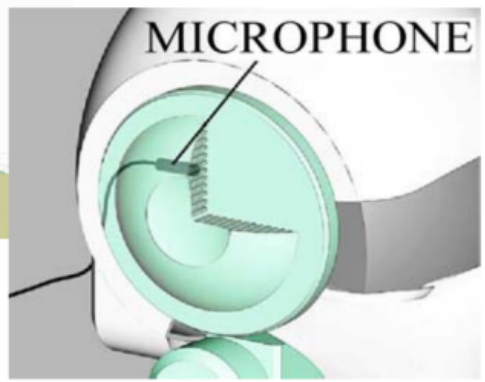
J. Bregman: Auditory Scene Analysis, MIT Press, 1990

Humanoid *SIG*

Testbed for social interaction

- 4 motors
- 2 pairs of microphones
- A pair of cameras
- Cover designed by a professional industrial designer





SIG

Assumptions for HARK

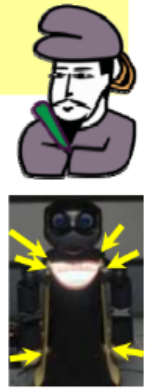
1. Spatial sparseness (at least 20 degrees)
2. Not temporal sparseness,
3. Not frequency sparseness,
4. Not the fixed number of talkers,
5. Not static talkers.

Problem Statement:

Input: a sound mixture

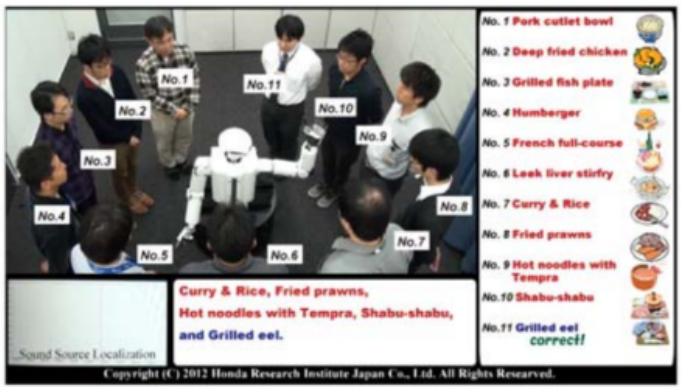
Output: 3D localization, separated sounds and speech recognition results (if speech) in almost real-time

Demos of Simultaneous Talkers



Iq#5336#z lwk#5#p lfurskrqhv
Xqghughwhup lqhg#frqglwrq

Iq#5339 KDUN#z lwk#;0fk#p lfurskrqh#bud|

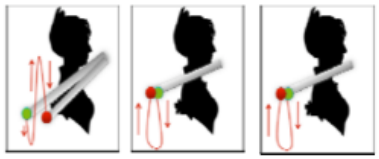


Simultaneous Speech Recognition

~ Meal Order Taking ~

Iq#5345#z lwk#490fk#p lfurskrqh#bud|

Music co-player robots



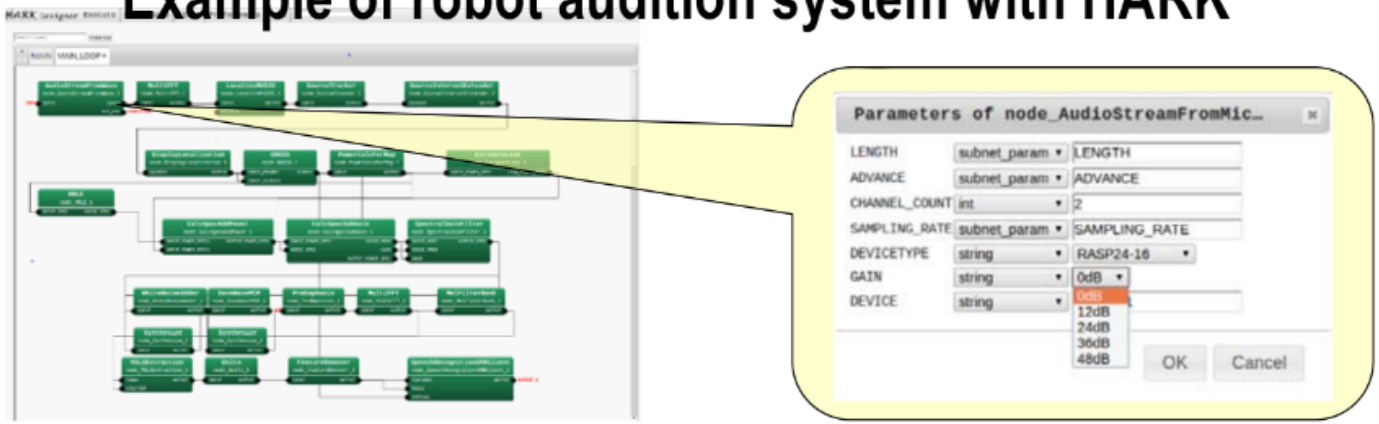
Nao listens to the shaker to detect the beat.

Nao knows the tempo and can groove with the humans.

Features in HARK (1)

- **New GUI programming environment HARK Designer**
  - Web-based programming environment (jQuery, node.js, HTML5)
  - Chrome/Firefox on Linux/Windows
  - no overhead in module communication (frame-based processing) provided by Batchflow of FlowDesigner

Example of robot audition system with HARK

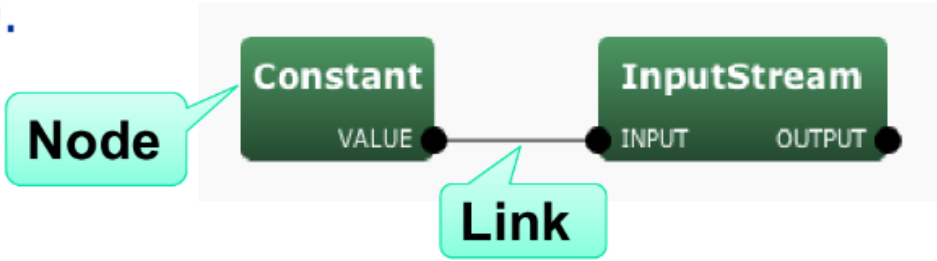


a) Module network

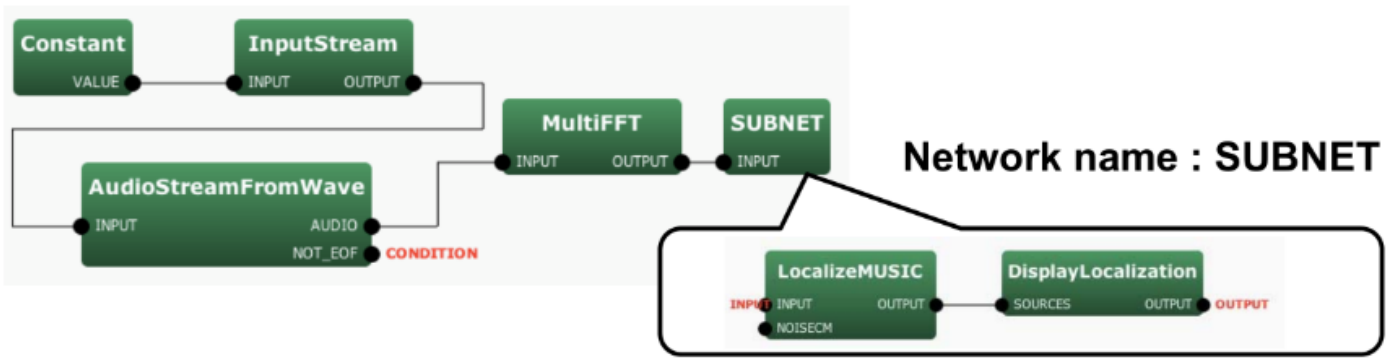
b) Property setting window

HARK Designer : What is a Network?

- A **network** is made of nodes (functions) and their links (data pathways).



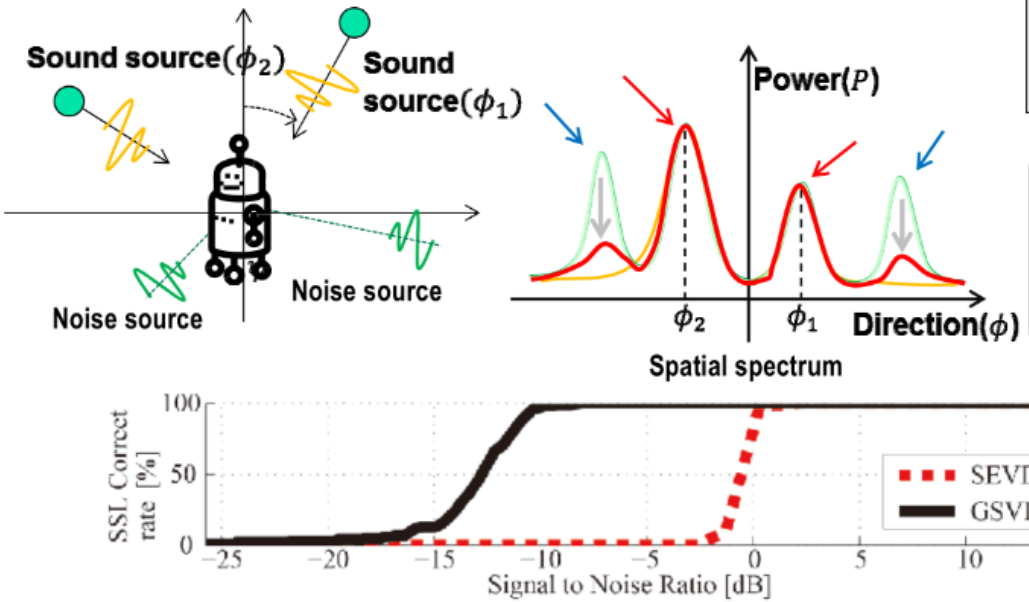
- Networks have a hierarchical architecture, and can contain **sub-networks**.

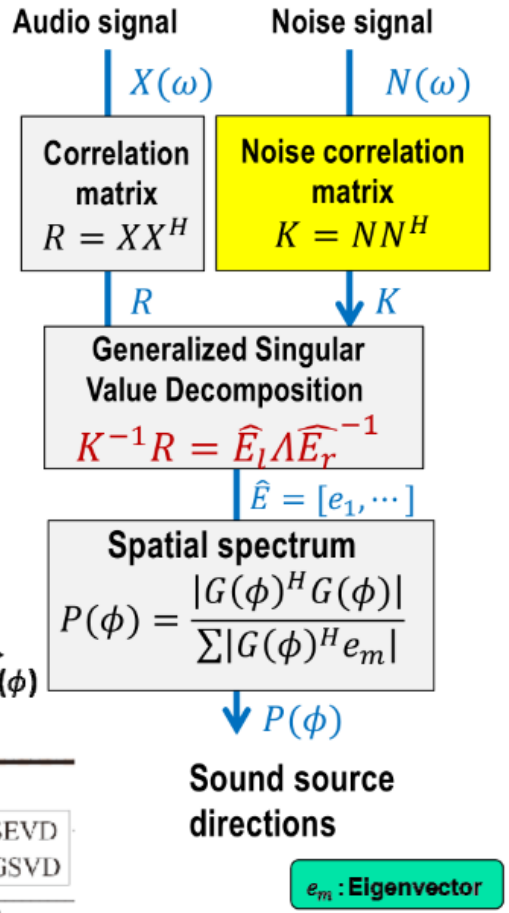


SSL: GSVD-MUSIC

GSVD-MUSIC [Nakamura 2010]

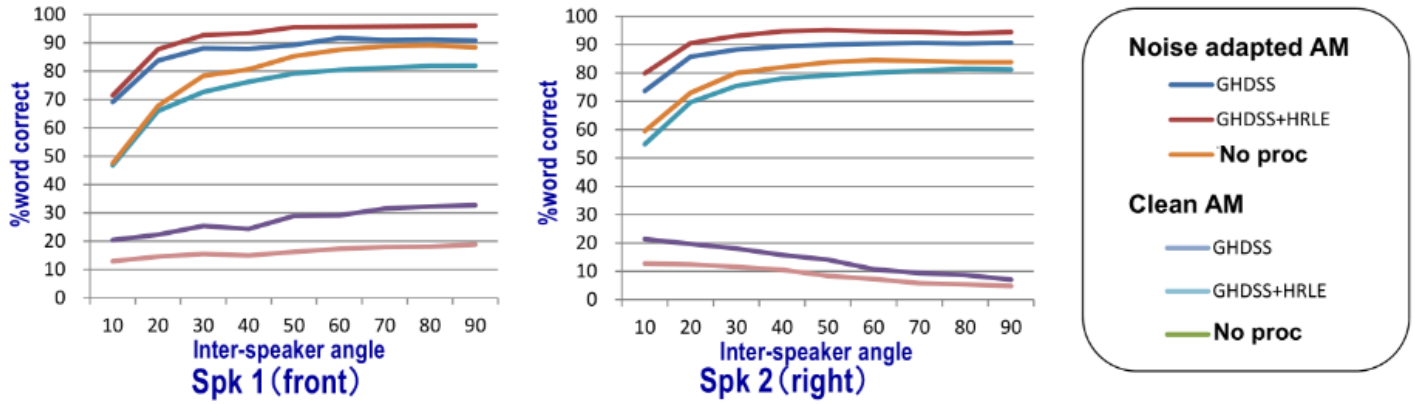
- A kind of *adaptive beamformer*, MUSIC
- MUSIC based on Generalized Singular Value Decomposition
- Introduction of *GSVD to whiten (=suppress) noise sources using a noise correlation matrix, K,* generated from pre-captured noise



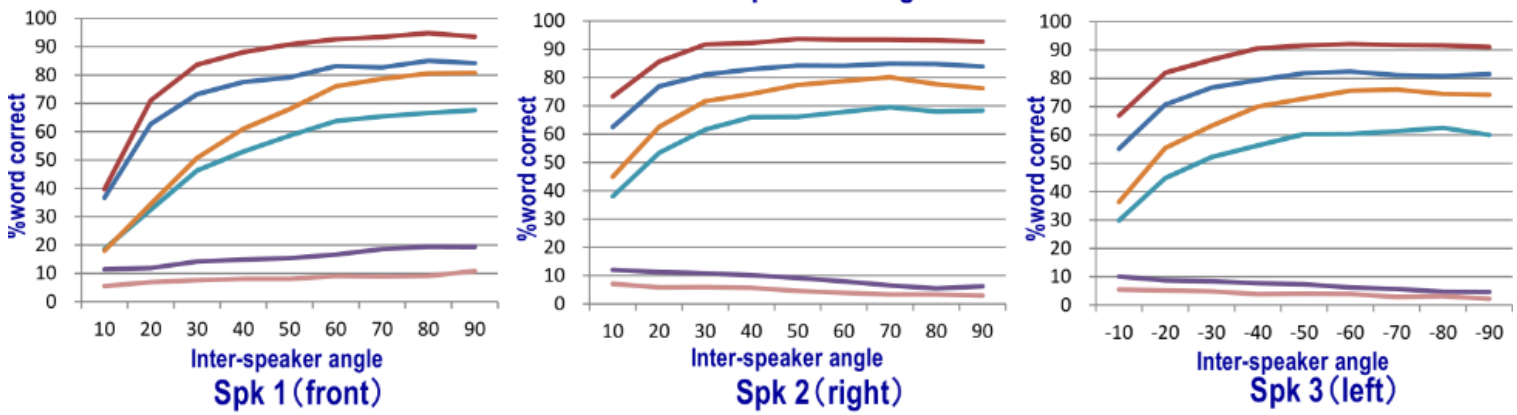


Simultaneous Speech Recognition

Room: 4m x 7m, RT20=0.2[s], **Mic. Array:** 8ch circular array around the head of Hearbo
Test data: ATR216 phonetically-balanced words, word/spk-open, **Distance:** 1[m]



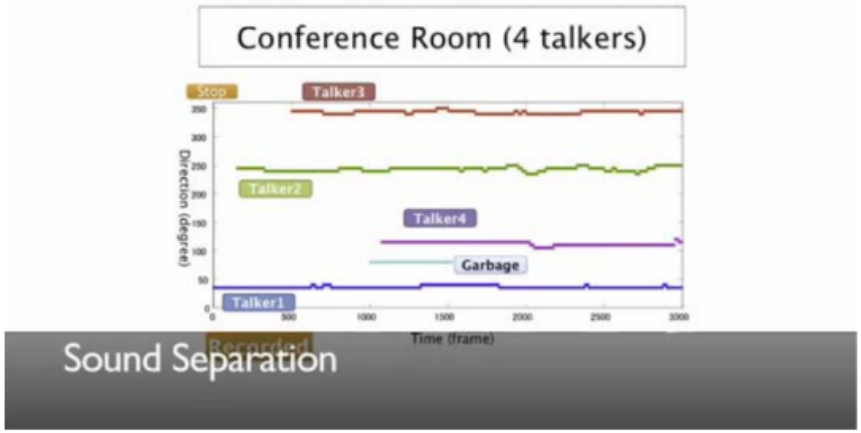
Two simultaneous speech recognition

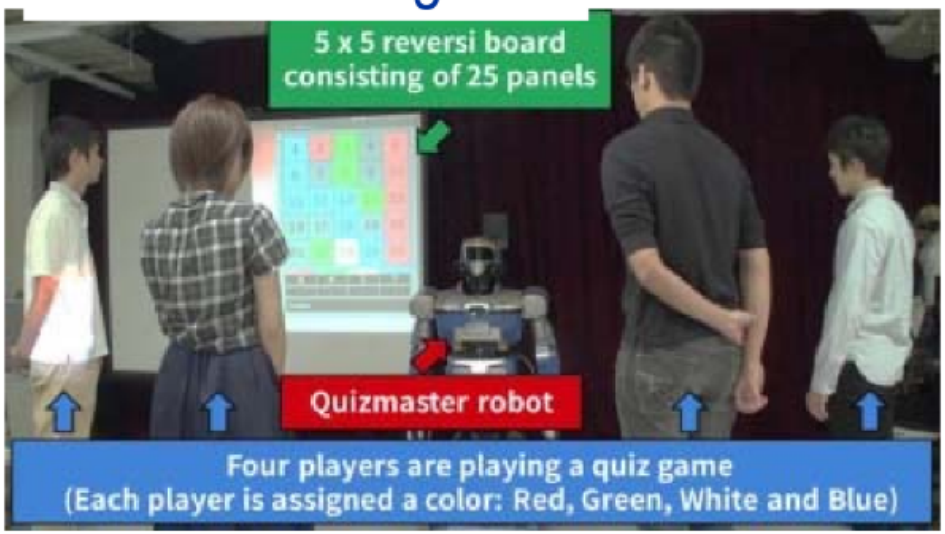


Three simultaneous speech recognition

Multiparty interactions

Telepresence robot at Willow Garage





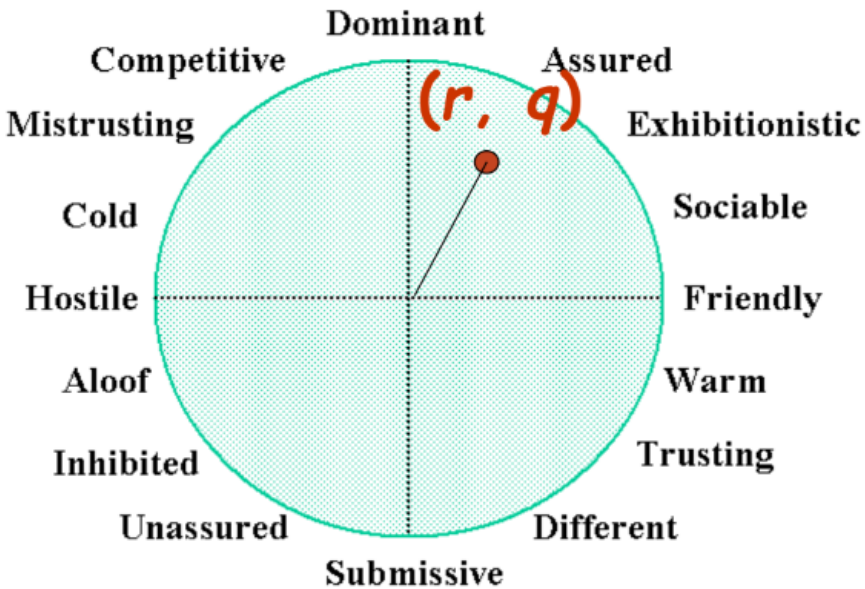
Robot quizmaster for a fasterst-voice-first quize game

Current HMI for IVI system (car navi)

Personality in Focus-of-Attention

Personality is represented as a point (r, q) in the **Interpersonal Circumplex**.

keep attention



Look the other way

Follow speakers

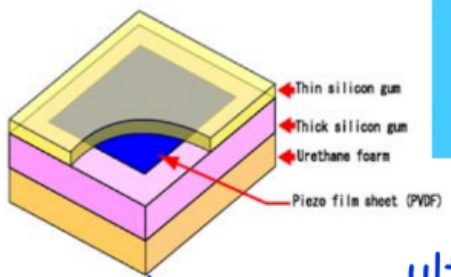
Turn to new stream

Humanoid Robot SIG2

SIG2

- **Input**
  - Two microphones （auditory）
    - ➢Each microphone is covered with silicon in the shape of the human external ear.
  - Two cameras （visual）
  - Tactile sensors （touch）
- **Output**
  - Four-degree-of-freedom
    - ➢ （neck 3、body 1）
  - Vehicle （movement）
  - Speaker （utterance）
    - ➢two types (directional, omni-directional)

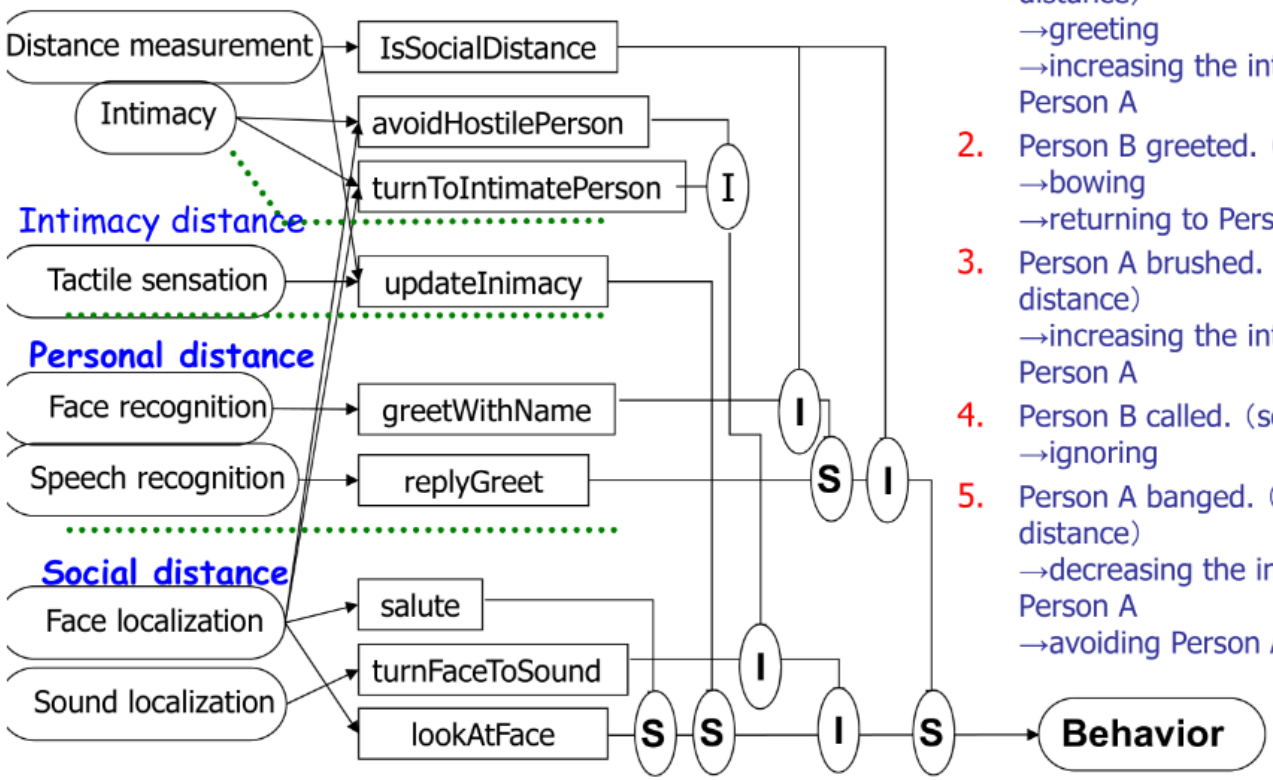
external ear



piezo sensor

ultra-directional parametric loud speaker

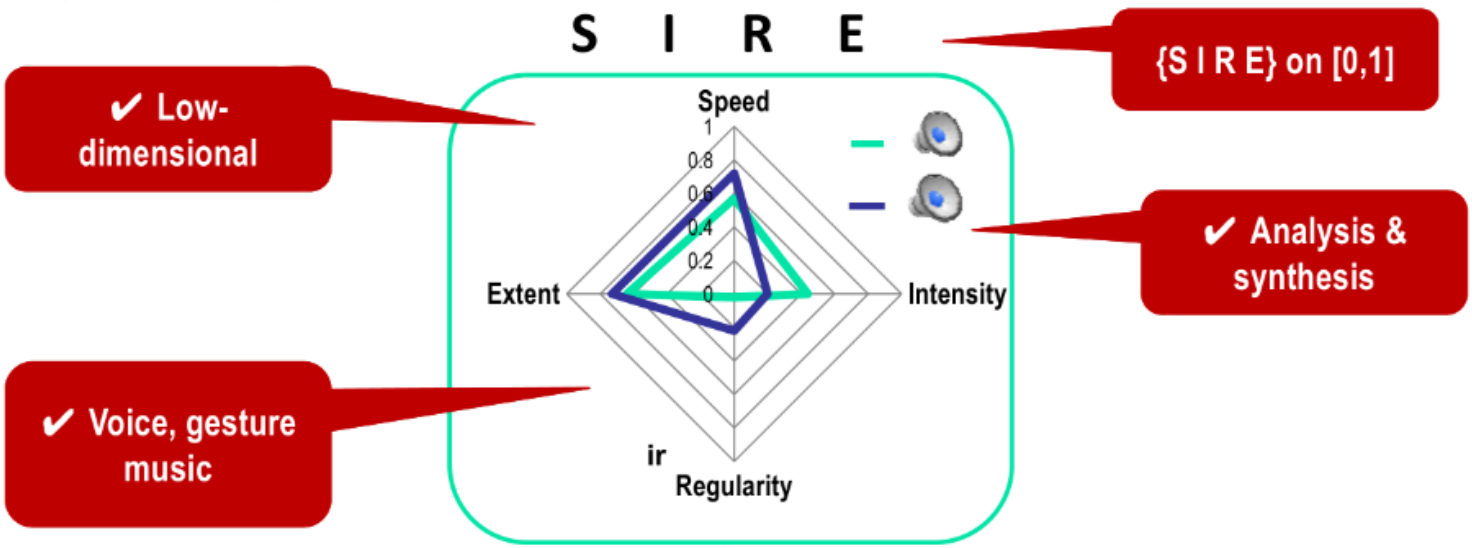
Changing behavior by intimacy (implementation)



1. Person A greeted. (intimate distance)
   →greeting
   →increasing the intimacy with Person A
2. Person B greeted. (social distance)
   →bowing
   →returning to Person A
3. Person A brushed. (intimate distance)
   →increasing the intimacy with Person A
4. Person B called. (social distance)
   →ignoring
5. Person A banged. (intimate distance)
   →decreasing the intimacy with Person A
   →avoiding Person A

Approach: The SIRE Model

SIRE: Description of Emotion based on **Speed**, **Intensity**, **irRegularity** and **Extent**. We represent emotion based on its **dynamic** characteristics.



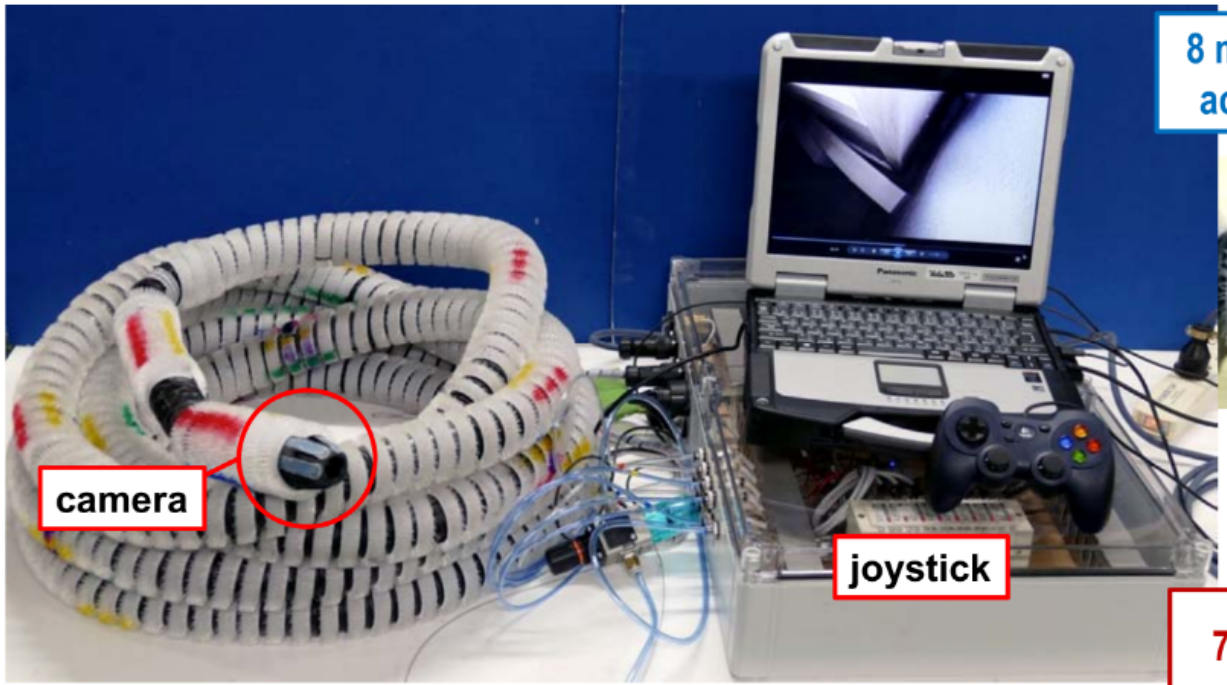
Gesture Parameter Mappings

Speed	Intensity	irRegularity	Extent
Joint speed $t_0 = t_0$ $t_1 = \max(S \cdot t_1, \underline{m})$ $t_2 = \max(S \cdot t_2, \underline{m})$	***Joint acceleration*** $t_0 = t_0$ $t_1 = \max(I \cdot t_1, \underline{m})$ $t_2 = t_2$	***Phase offset*** $\delta_t = (1 - R) \cdot \underline{r}$ $t_i = \delta_t + t_i$	***Gesture size*** $p_1 = p_0 + E \cdot (p_1 - p_0)$

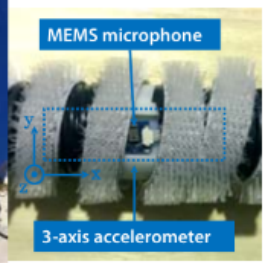
0

1

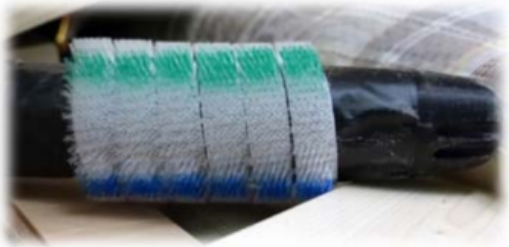
Hose-Shaped Robot: Active Scope Camera

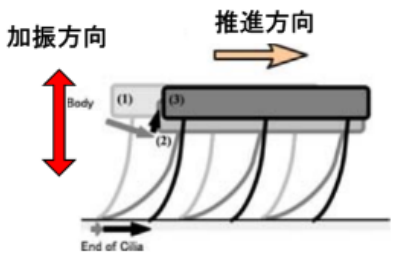


8 microphones & accelerometers



7 loudspeakers





2016/11/11 ImPACT TRC 3rd Field Evaluation

Successfully detect whistle sounds in a rainy condition
(The same task succeeded in all rehearsals)