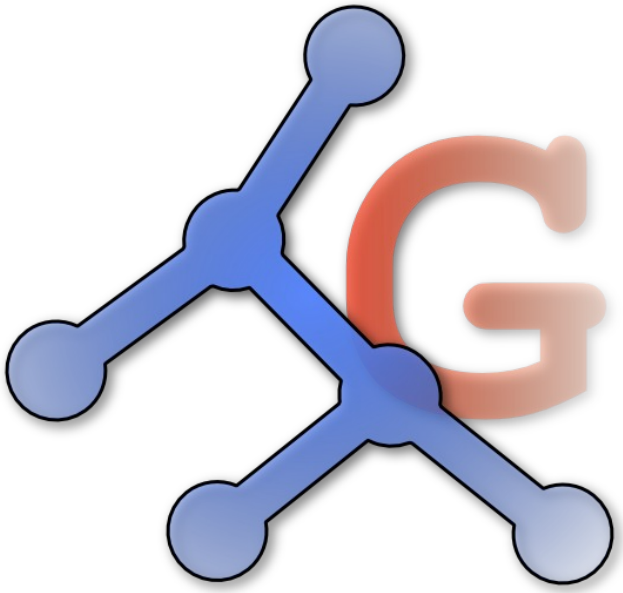


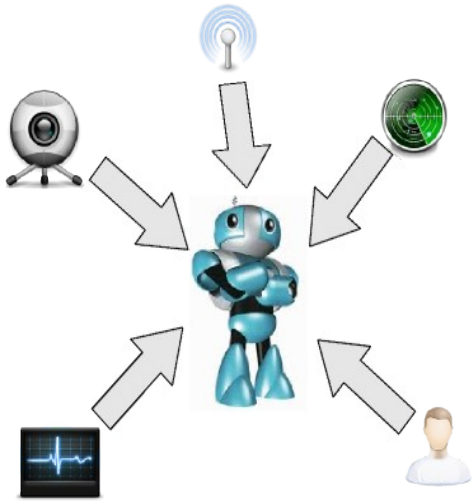
DCAI - 2012

Modeling a Mobile Robot using a Grammatical Model.

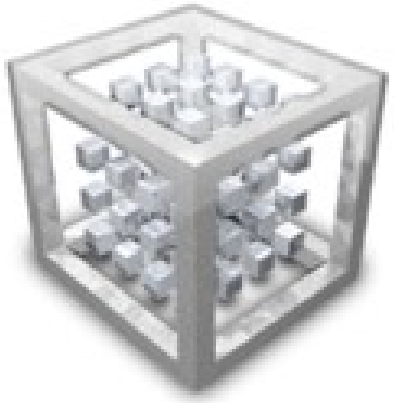
Authors:

Gabriel López García
Javier Gallego Sánchez
J.Luis Dalmau Espert
Rafael Molina Carmona
Patricia Compañ Rosique

Introduction



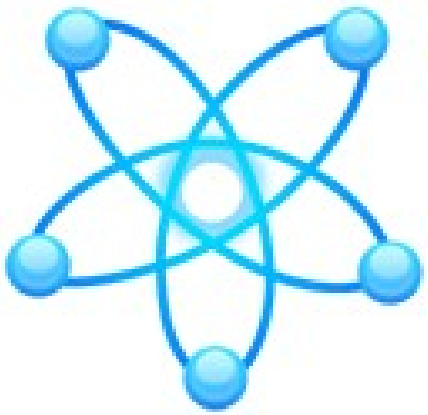
Growing disparity of available sensors



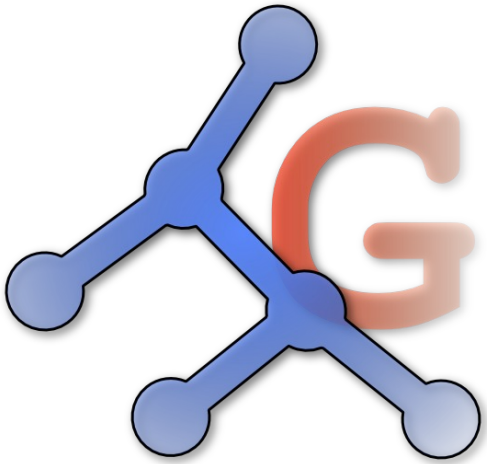
More complexity, but more accurate

Need integrate multiple information sense and complexity

Objectives



Integrate multimodal input and data of different nature



Use a grammatical model

Grammar definition integrate activities, visualization and interaction with user

VWG* Elements

Primitive (P)

Primitives describe objects

Transformation (T)

Transformations change the behaviour of primitives

*VWG = Virtual World Generator

VWG Elements

Actors (A^{D}_{ATTR})

Actors define system's activities in the virtual world

Events

Events cause the activation of a certain activity

Grammar

Rule 1. **WORLD** → OBJECTS

Rule 2. **OBJECTS** → OBJECT | OBJECT · OBJECTS

Rule 3. **OBJECT** → FIGURE | TRANFORM | ACTOR

Rule 4. **ACTOR** → $a^{H}_{attr}, a^{H}_{attr} \in A^{D}_{ATTR}$

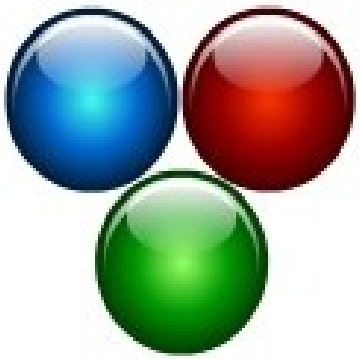
Rule 5. **TRANSFORM** → t(OBJECTS)

Rule 6. **FIGURE** → p^{+}

Semantic Functions

Sequence of primitives

Rule 6
Figures



$\alpha : \boldsymbol{P} \rightarrow \boldsymbol{G}$

Rule 5
Transformations

$\beta : \boldsymbol{P} \rightarrow \boldsymbol{G}$

$\delta : \boldsymbol{P} \rightarrow \boldsymbol{G}$

The scope is limited by '()' symbols

G:

Semantic Functions

Visualization Function

$$\varphi(w)=\begin{cases}\alpha(w) & \text{if } w\in P\\ \beta(t);\varphi(v);\delta(t) & \text{if } w=t(v)\wedge v\in L(M)\\ \varphi(s);\varphi(t) & \text{if } w=s\cdot t\wedge s,t\in L(M)\end{cases}$$

It is the function that draws a set of primitives and transformations to be displayed on a geometry

Semantic Functions

Rule 4 - Actors

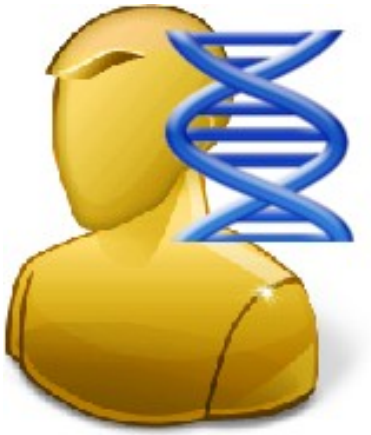
$$\lambda : \boldsymbol{A}^{D}_{\langle ATTR \rangle} \times \boldsymbol{E}^{\boldsymbol{D}} \rightarrow \boldsymbol{L}(\boldsymbol{M})$$

$$\lambda\left(a^{H}_{\langle ATTR \rangle}, e^{h}\right) = \begin{cases} u_0 \in L(M) & Si\ h = h_o \\ \cdots & \\ u_n \in L(M) & Si\ h = h_n \\ a^{d}_{\langle attr \rangle}(v) & Si\ f \neq d \end{cases}$$

The evolution function is responsible for the activity of system

Semantic Functions

Evolution Function



$$\eta(w,S)=\begin{cases} w & \text{if } w\in P \\ t(\eta(v,S)) & \text{if } w=t(v) \\ \prod_{\forall e^i\in S}(\lambda(a^d_{\langle attr\rangle},e^i)) & \text{if } w=a^H_{\langle ATTR\rangle}(y) \\ \eta(u,S)\cdot\eta(v,S) & \text{if } w=u\cdot v \end{cases}$$

It is the function that makes the system evolve

Semantic Functions

Rule 4 - Actors

$$\lambda : \boldsymbol{A}^{V}_{\langle ATTR \rangle} \times \boldsymbol{E}^{\boldsymbol{V}} \rightarrow \boldsymbol{L}(\boldsymbol{M}')$$

$$V \subseteq D, E^{V} \subseteq E^{D}, L(M') \subset L(M)$$

$$\theta(a^{H}_{\langle ATTR \rangle}, e^{v}) = \begin{Bmatrix} z_0 \in L(M') & Si\ v = v_o \\ \cdots & \\ v_n \in L(M') & Si\ v = v_n \\ \varepsilon & Si\ v \notin H \cap V \end{Bmatrix}$$

The visualization function is responsible for translating actors into primitives and transformations

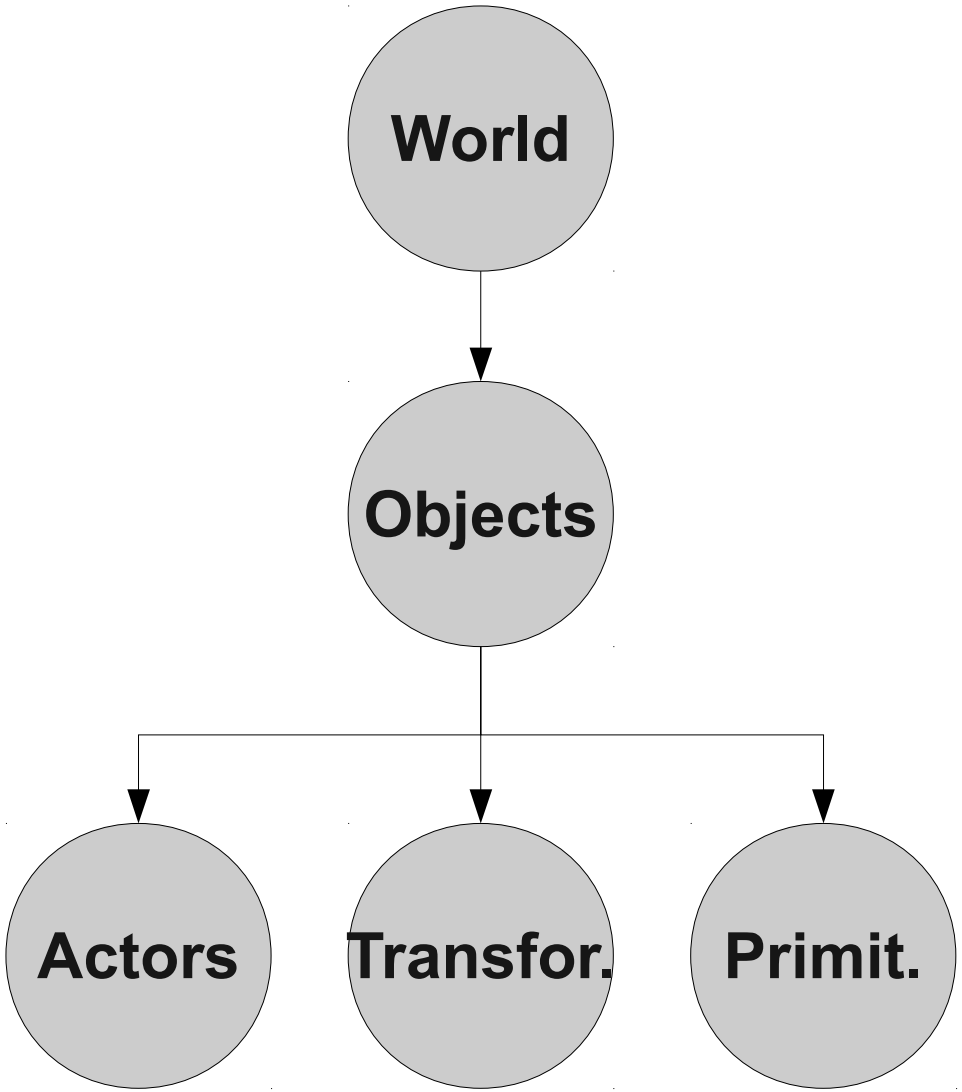
Semantic Functions

The visualization function of the system

$$\pi(w, S') = \begin{cases} w & \text{if } w \in P^{+} \\ t(\pi(v, S')) & \text{if } w = t(v) \\ \prod_{\forall S' \in e^{i}} \theta(a^{v}_{\langle attr \rangle}, e^{i}) & \text{if } w = a^{H}_{\langle ATTR \rangle} \\ \pi(u, e^{v}) \cdot \pi(v, e^{v}) & \text{if } w = u \cdot v \end{cases}$$

It is the function that translates every actors into primitives and transformations

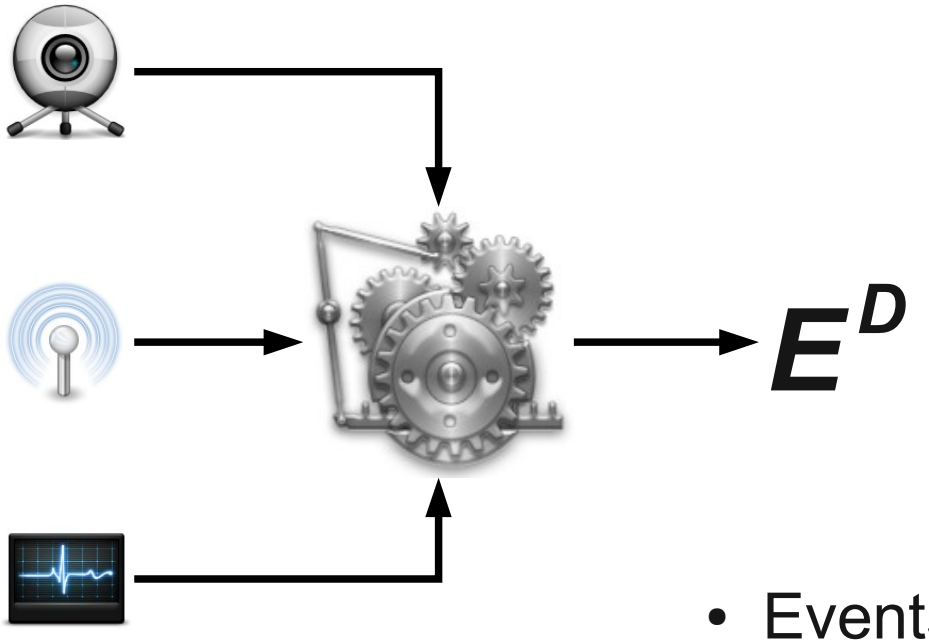
Semantic Functions



Rule 3,2,1

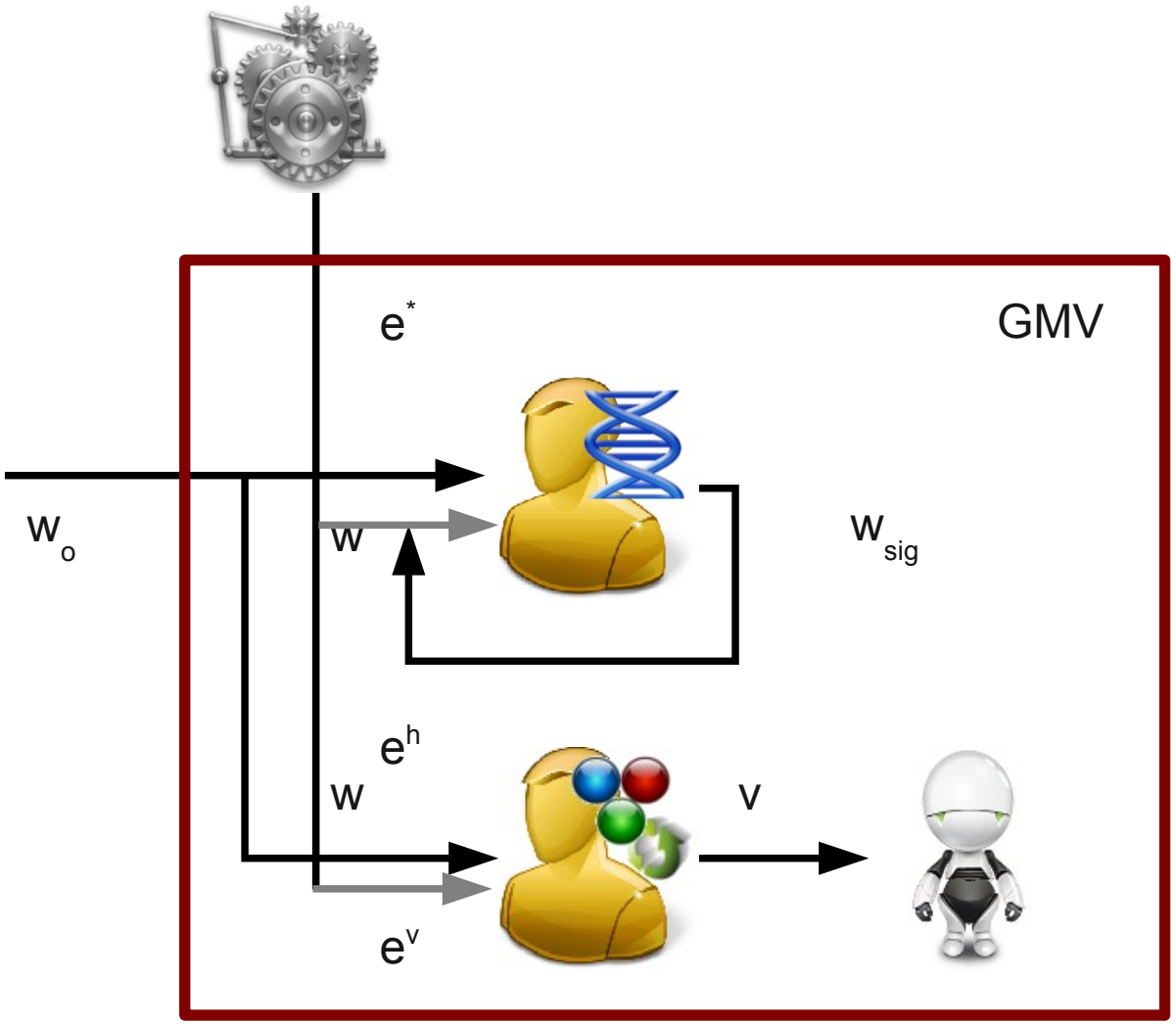
These rules break down the strings and convert them into substrings

Event Generators

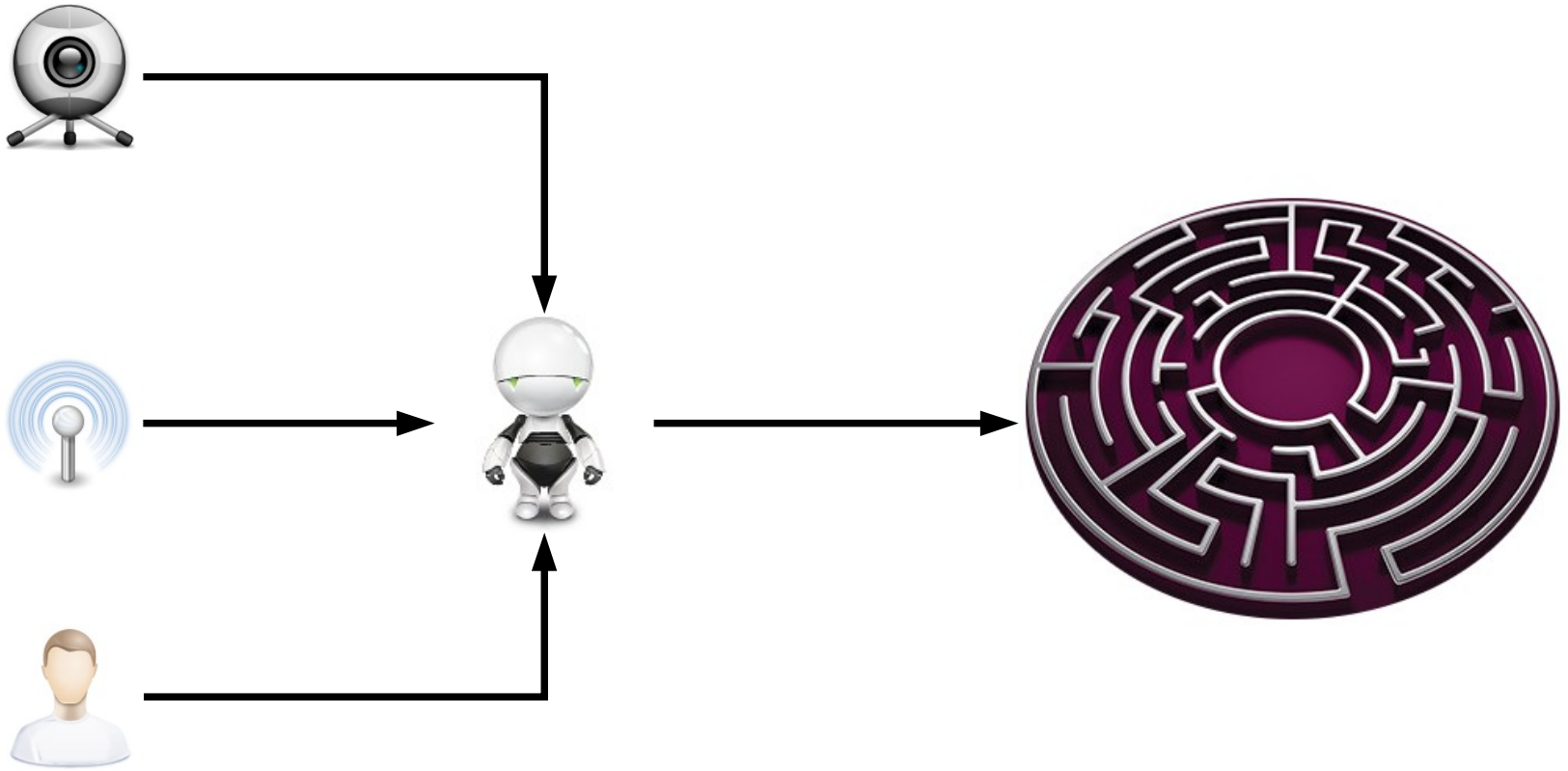


- Events are the mechanism to model activity in the system
- The event generator is a function to generate events from different devices

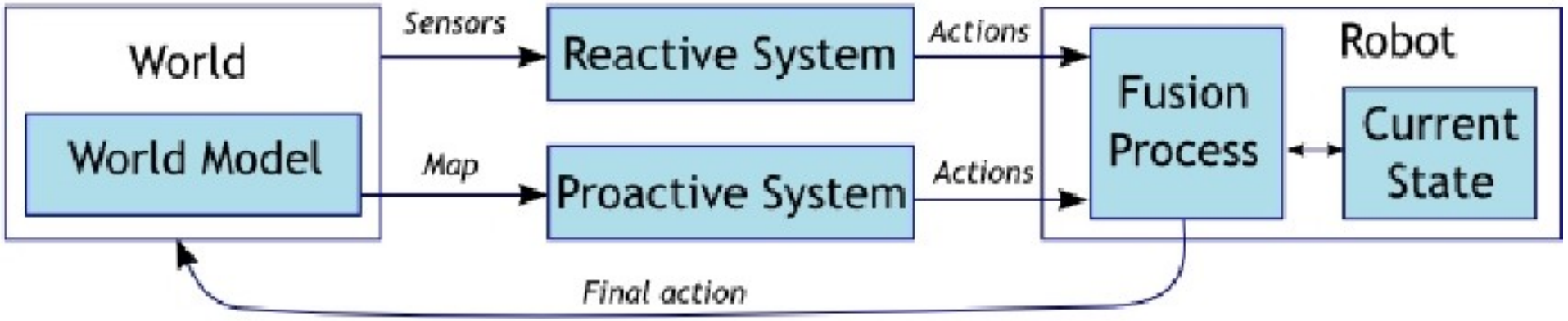
Algorithm



Case study

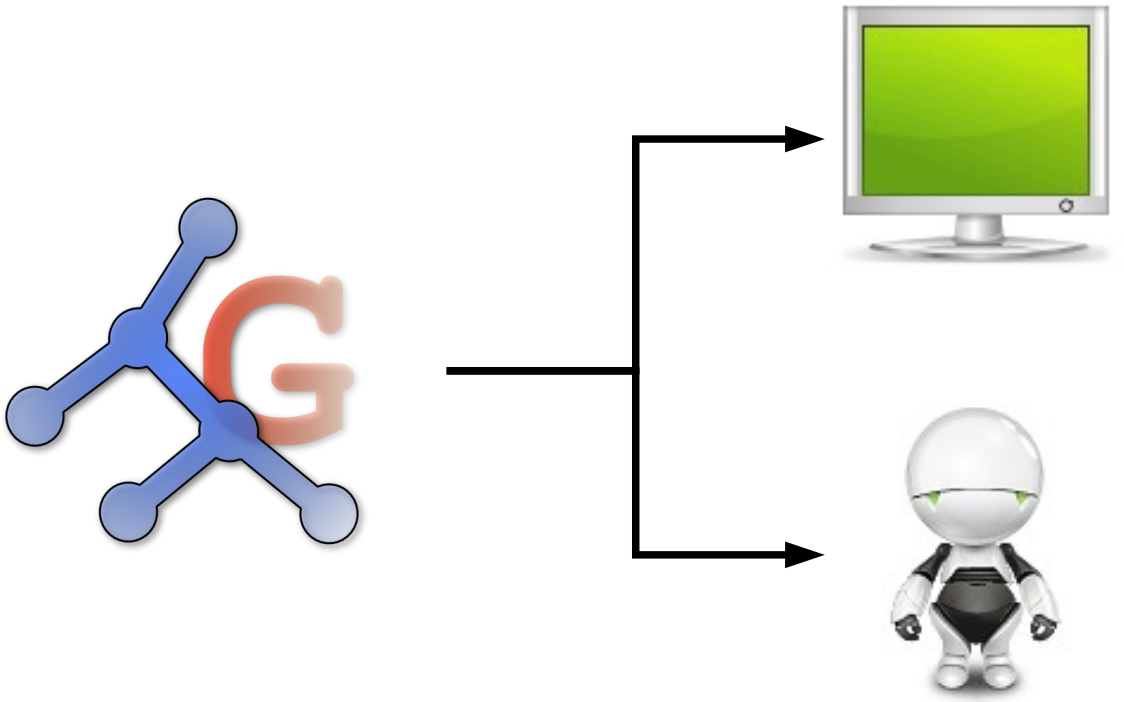


Case study



This kind of system can be modeled by an hybrid scheme that can be adapted using VWG

Case study



Simulator

Real

We can execute the same string into a simulator environment or a real robot

Case study

PRobot	Draw the robot in the GS	No action
TMove *<dist>*	Move a distance 'dist' in the GS	Move a distance 'dist'
TRotate *<angle>*	Rotate an angle 'angle' in GS.	Rotate an angle 'angle'

Case study

Events

gCamera when a marker is detected

gLaser when the laser detects an obstacle

gObjetive when the user sets an objective

gDecide when the robot decides an action

gExecute when the robot executes an action

Case study

Evolution function

$$\lambda\left(ARobot^{E}_{\langle g,r,c,an,o,act\rangle}, e\right)=\begin{cases} ARobot^{E}_{\langle g',r,c,an,o,act\rangle} & if\ e=eLaser_{\langle dist,angle\rangle} \\ ARobot^{E}_{\langle g',r,c,an,o,act\rangle} & if\ e=eCamera_{\langle marker\rangle} \\ ARobot^{E}_{\langle g,r',c',an',o,act'\rangle} & if\ e=eDecide \\ ARobot^{E}_{\langle g,r,c,an,o,act\rangle} & if\ e=eExecute \\ ARobot^{E}_{\langle g,r,c,an,o',act\rangle} & if\ e=eObjetive_{\langle marker\rangle} \\ ARobot^{E}_{\langle g,r,c,an,o,act\rangle} & Otherwise \end{cases}$$

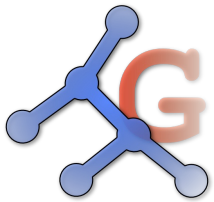
Analysis

Introduce new AI algorithm

Change or increment new devices

Multi-robot system

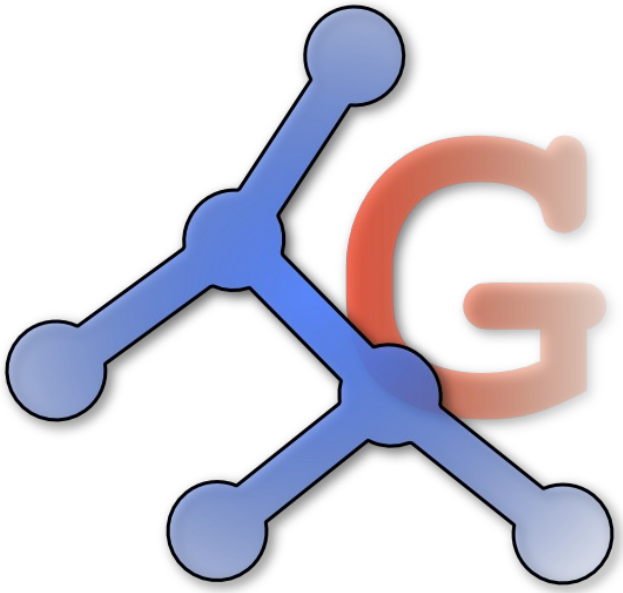
Conclusion



The new model defines a virtual world, independently from the underlying physical layer

The model allows an abstract representation

It can change, simulate or add new devices



DCAI - 2012

Modeling a Mobile Robot using a Grammatical Model.

Universitat d'Alacant
Universidad de Alicante

informática industrial e inteligencia artificial

creative commons

Authors:

Gabriel López García
Javier Gallego Sánchez
J.Luis Dalmau Espert
Rafael Molina Carmona
Patricia Compañ Rosique