Robotic Summer School 2009

Subsumption architecture

Andrej Lúčny

Department of Applied Informatics,

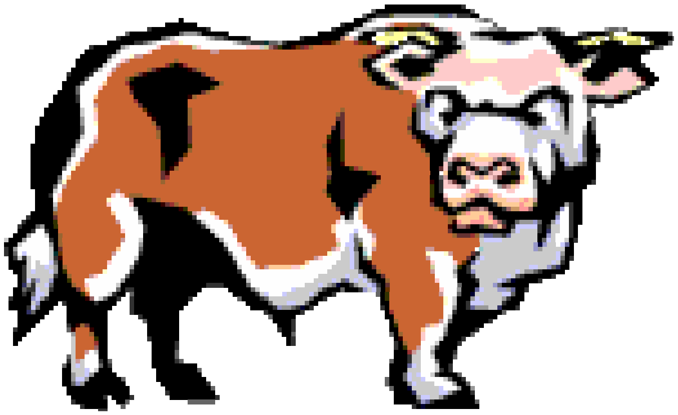
FMFI, Comenius University, Bratislava

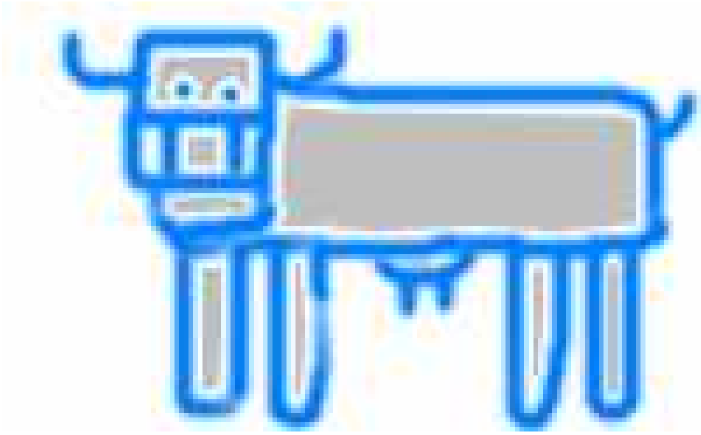
lucny@fmph.uniba.sk

www.microstep-mis.com/~andy

Biomimetic approach

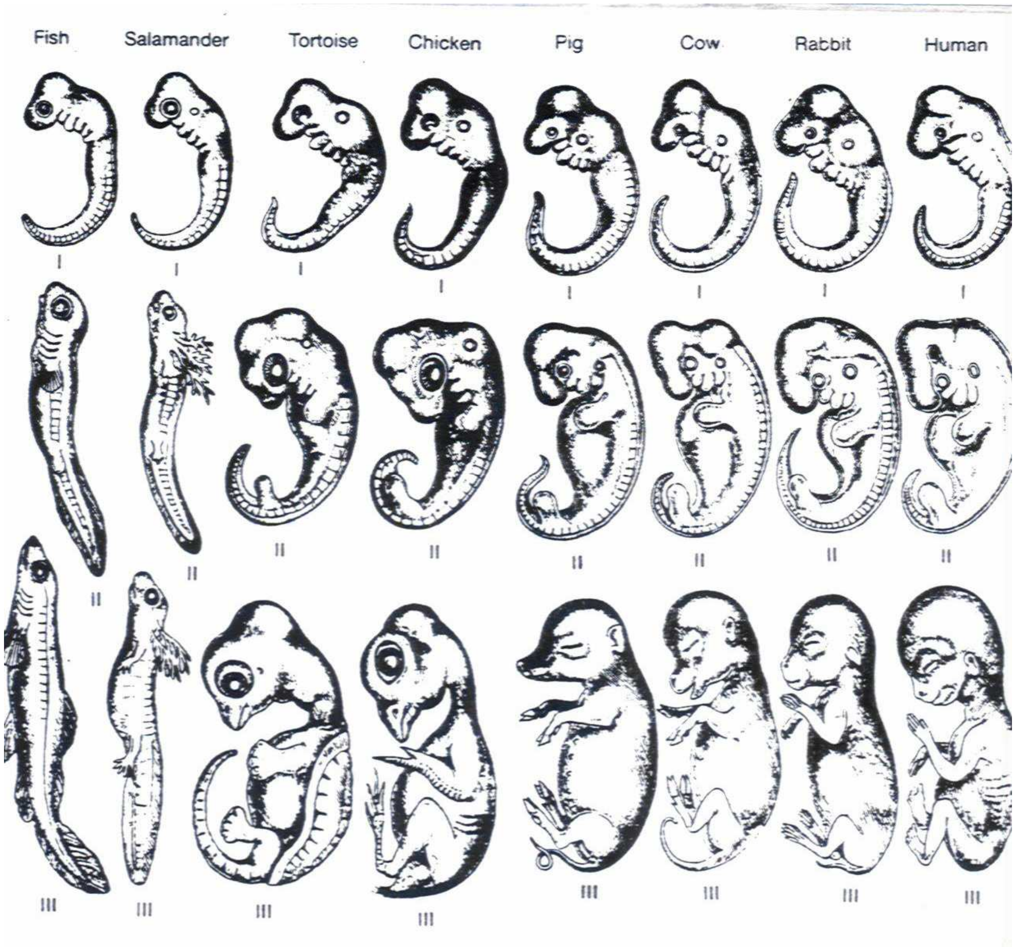
- Living creatures provide to engineers inspiration: engineers can mimic what nature has already created





The main biological inspiration

- Structure of a living creature is a result of (Darwinian) evolution
- The structure reflects steps of the evolution



Adaptive systems versus incremental development

- Nowadays, biomimetic approach is mostly concerned as tied with domain of genetic algorithms, genetic programming and machine learning
- However, we present strategy how to mimic evolution in a hand-made way following general principles of individual’s structure

Another biological inspiration

- Structure of living creatures has several typical features: parallelism and hierarchy
- Hierarchy is based rather on regulation than activation

if one severs an eel’s head from its spinal cord, the eel does not stop its sinuous swimming but its movements become perfectly regular and continuous. It means its brain rather inhibits and regulates its spinal cord than controlling it directly

Photo: Andy Martinez

Why biomimetic approach ?

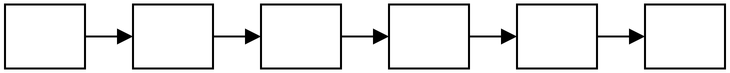
- Question: When biomimetic inspiration is necessary ?
- Answer: When the intended behavior of your robot is really complex

example: foreseeing vehicle

Scalability problem

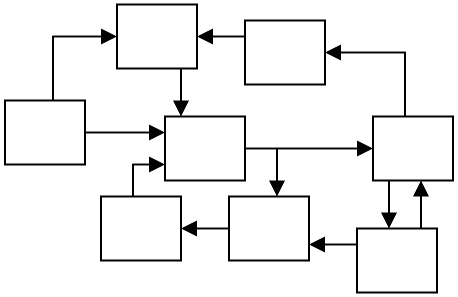
For many useful industrial applications it is sufficient to solve a particular problem

In this case, a simple pipeline is usually suitable architecture



Scalability problem

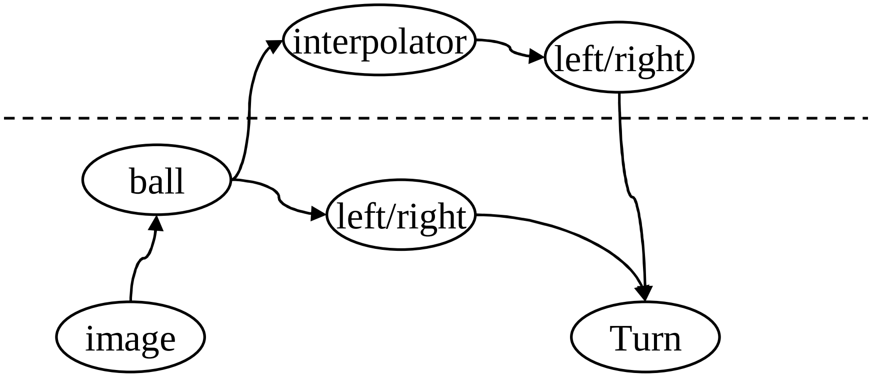
However other tasks like control of mobile robots or simulated creatures require a more complex behavior and more advanced architecture



Robot COG (AiLab MIT, 1993)

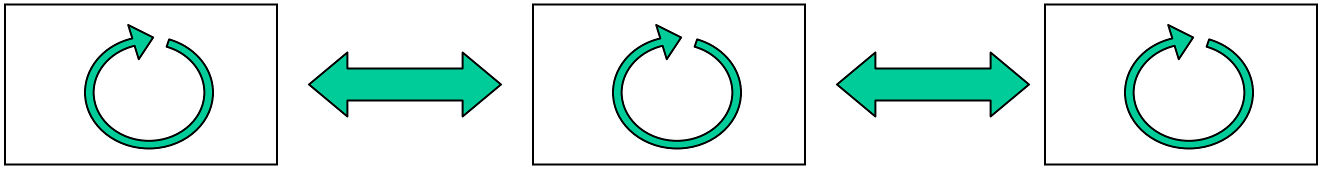
Minsky’s approach to architecture

- system containing many parallel modules (agents, resources)
- Complex global behavior emerges from simple local behaviours of individual modules (agents, resources)
- Control = activation of a proper set of modules at a proper situation



Looking for an architecture

- We have many modules continuosly running in parallel
- We have a mechanism of data exchange among the modules
- How can we provide the proper activation?



- A possible solution: subsumption

Subsumption

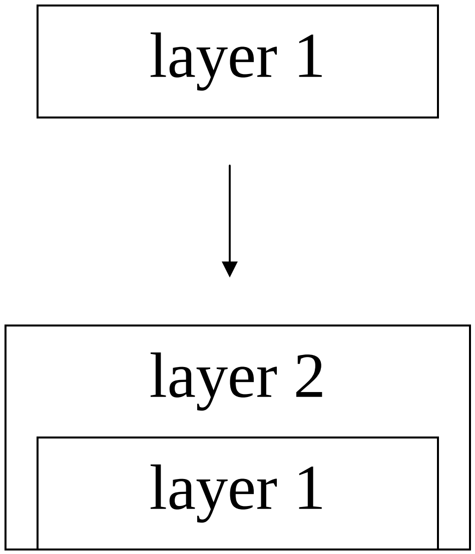
- It is a method for engineering of artificial systems with complex behavior
- It was proposed by R. Brooks in the mid-eighties

- It mimics simplified biological evolution

Subsumption

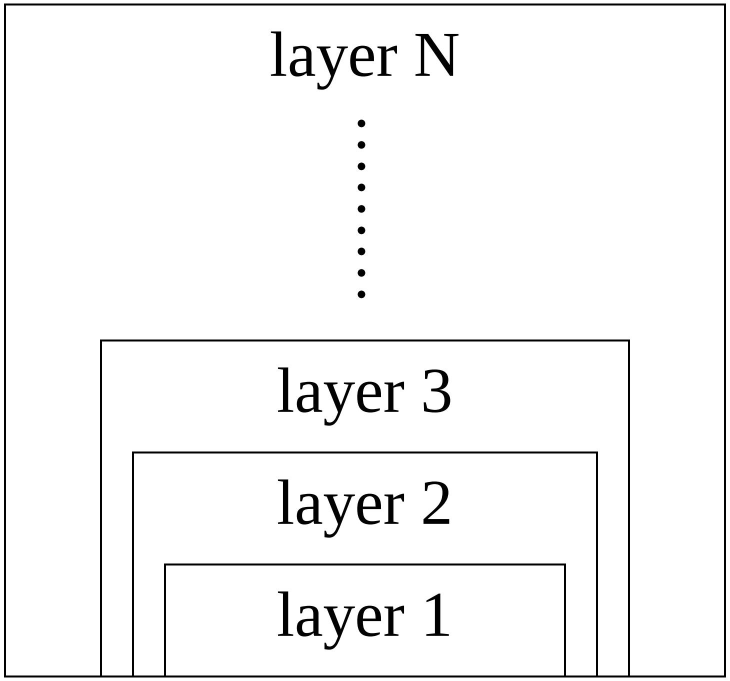
It is based on the evolutionary fact that any complex control has an origin in a simpler ancestor

The relation between the ancestor and its descendant is simplified here in such a way that the descendant contains exactly the same control mechanism as the ancestor, enriched just by an additional layer of control.



Simplification of evolution

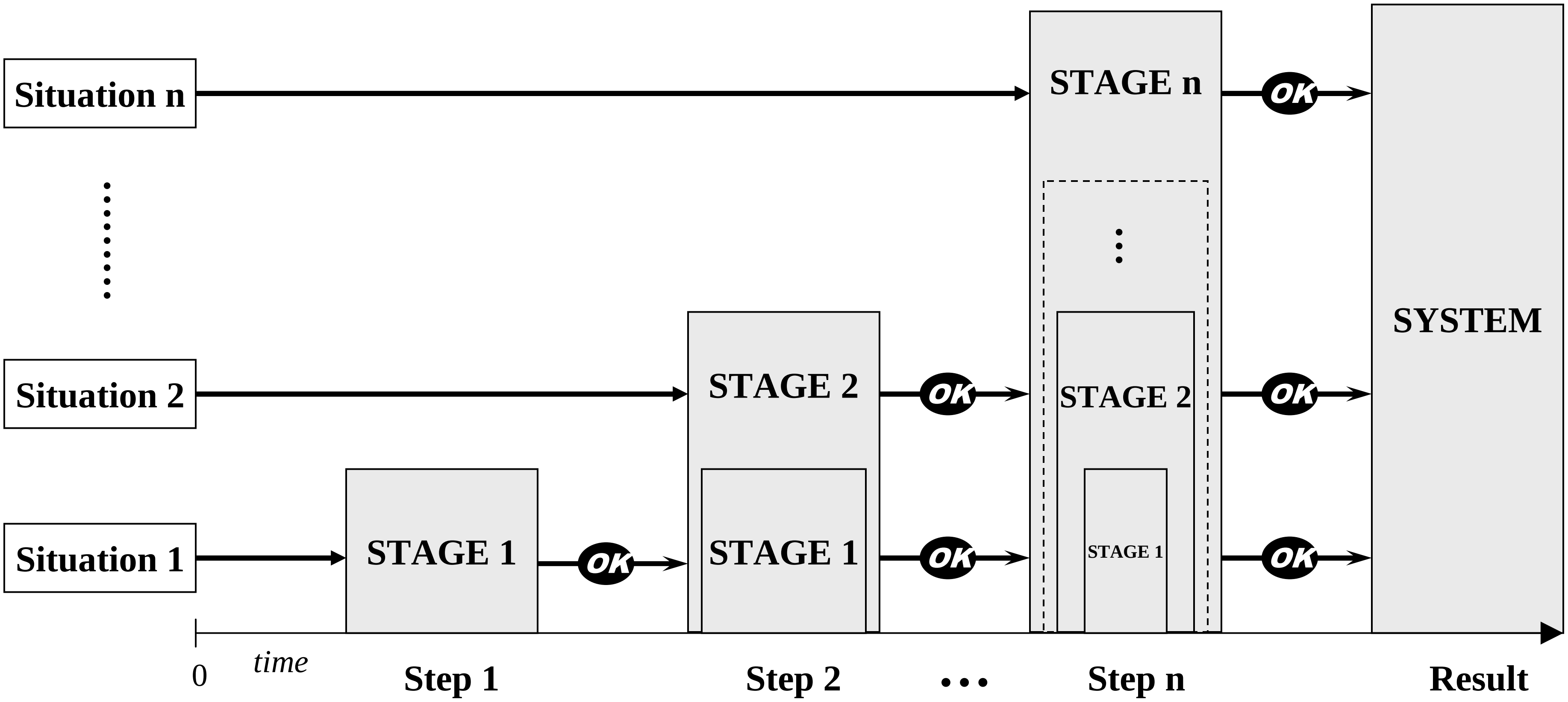
- In other words, the descendant mechanism subsumes complete mechanism of its ascendant; therefore the principle is called *subsumption*.



Development by subsumption

- At the first we design suitable sensors and actuators which are expected to be sufficient.
- Then we imagine a sequence of evolutionary steps which could result in the desired control starting from a simple base.
- Then we incrementally develop each step as an additional layer to the previous simpler version.
- In doing so, each step brings a set of new features, but causes no harm to features which have been already implemented.

Development by subsumption



Situatedness

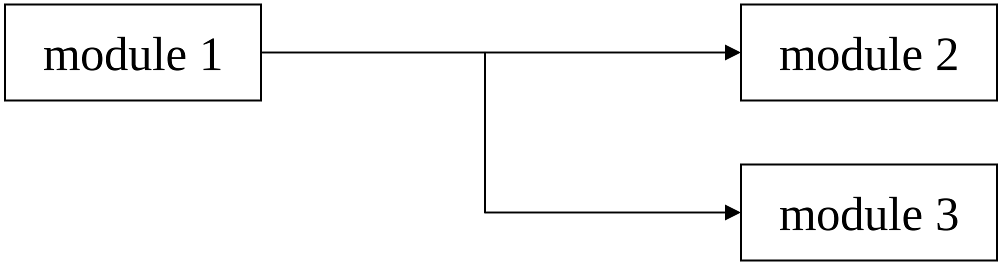
- It is recommended to design the evolutionary steps in such a way that each step corresponds to the desired control under simplified conditions.
- When the real situation is as simple as concerned for a particular step, it will be handled only by the corresponding layer and layers which are (evolutionary) older.
- Getting to more and more difficult situation, newer and newer levels are activated to influence the resulting control.

Appropriate modularity required

- However, how could the newer levels influence the older ones?
- The older levels have been designed for a particular purpose and have no interfaces for future development!
- The answer is: they have to have modular structure which enables the influence.

Subsumption architecture

- level consists of quite simple modules
- these modules communicate by messages sent through ‘wires’



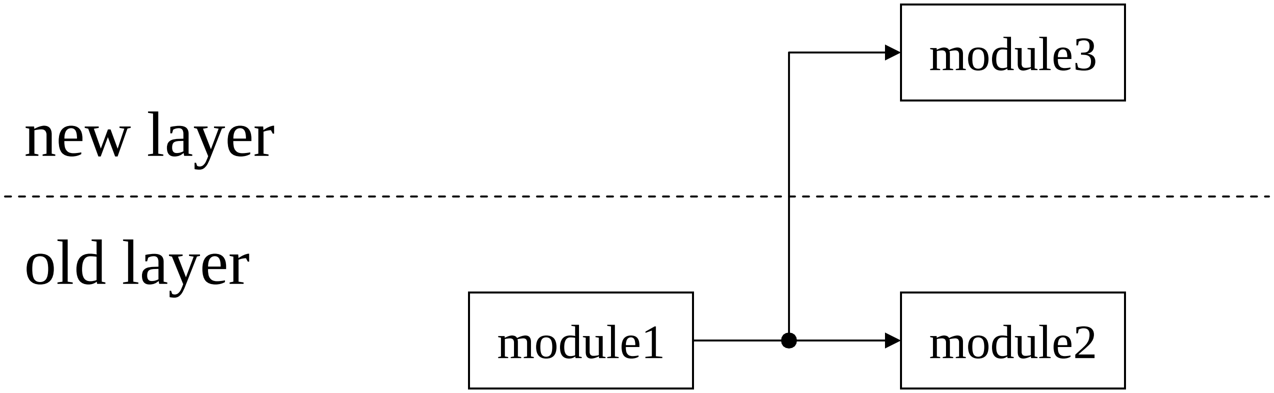
Influence mechanisms

Subsumption architecture suppose three mechanisms supporting the influence:

- Monitoring
- Inhibition
- Suppression

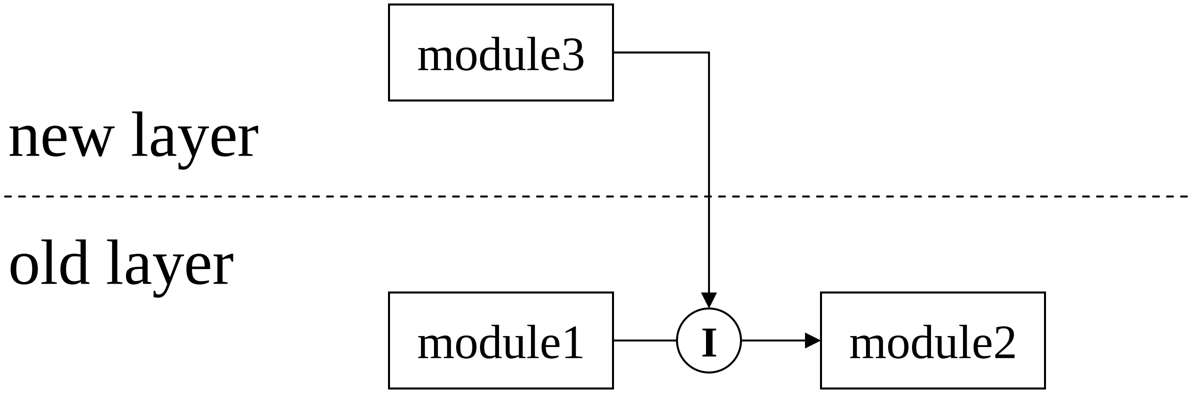
Monitoring

- the newer level can monitor messages communicated between modules in the older level by connecting to the same wire.



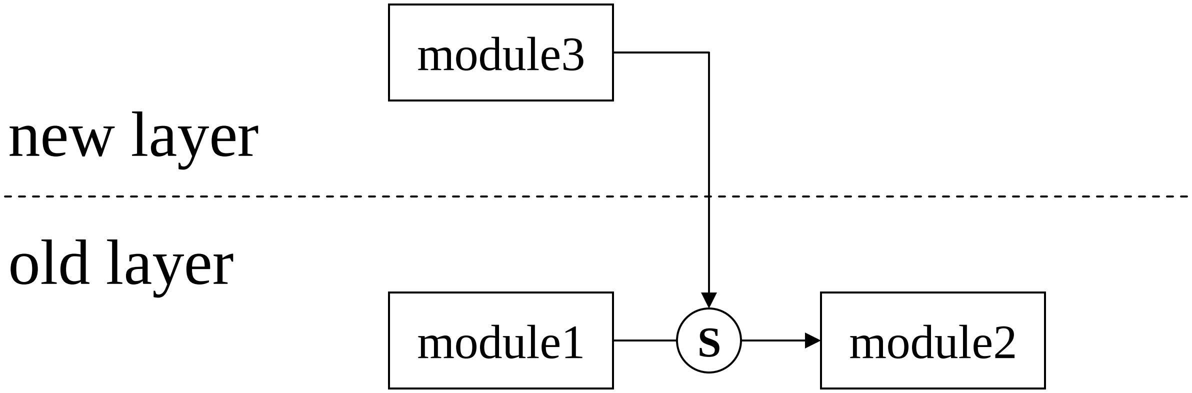
Inhibition

- it can also inhibit the communication by temporary interruption of the wire

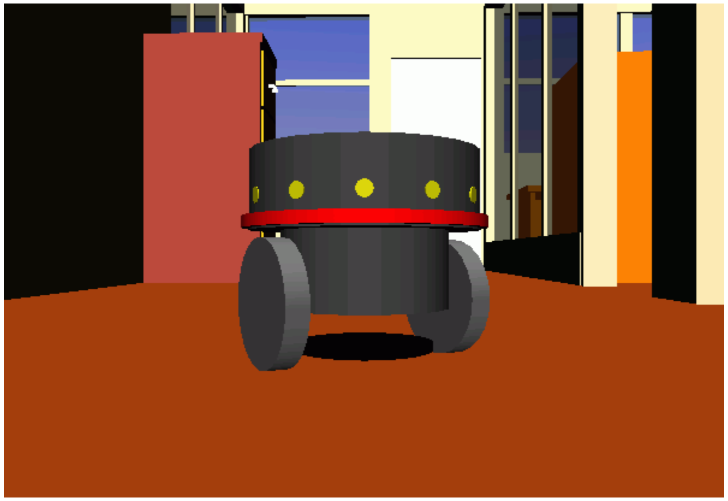


Suppression

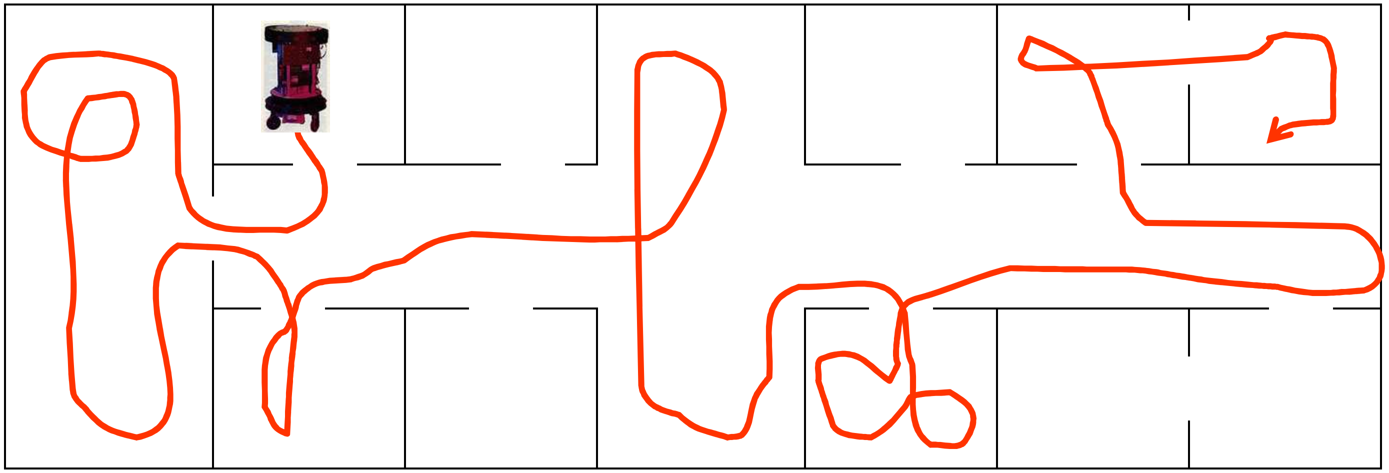
- even it can replace communicated messages



An example

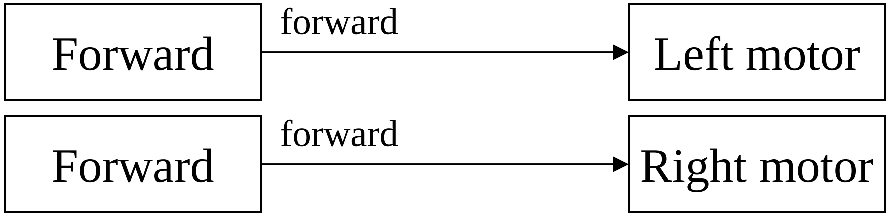


- Two wheeled robot navigating in a bureau (modified reimplementation of ALLEN, Brooks 1986)



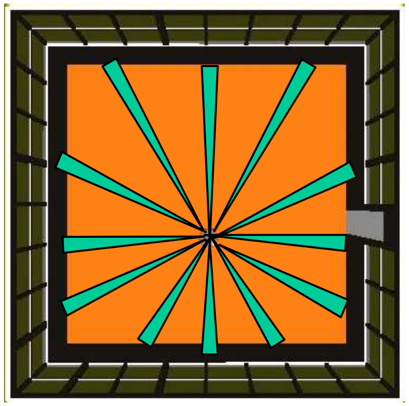
Example – step 1

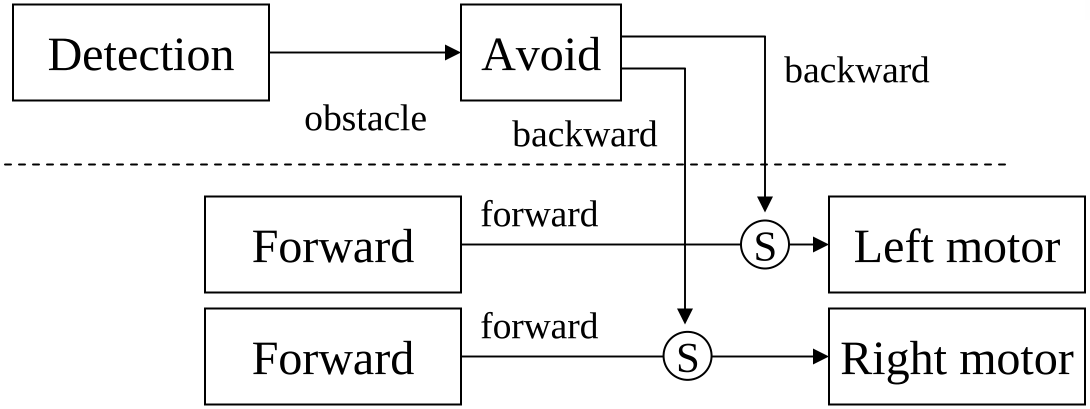
- *we start with robot which just goes forward*



Example – step 2

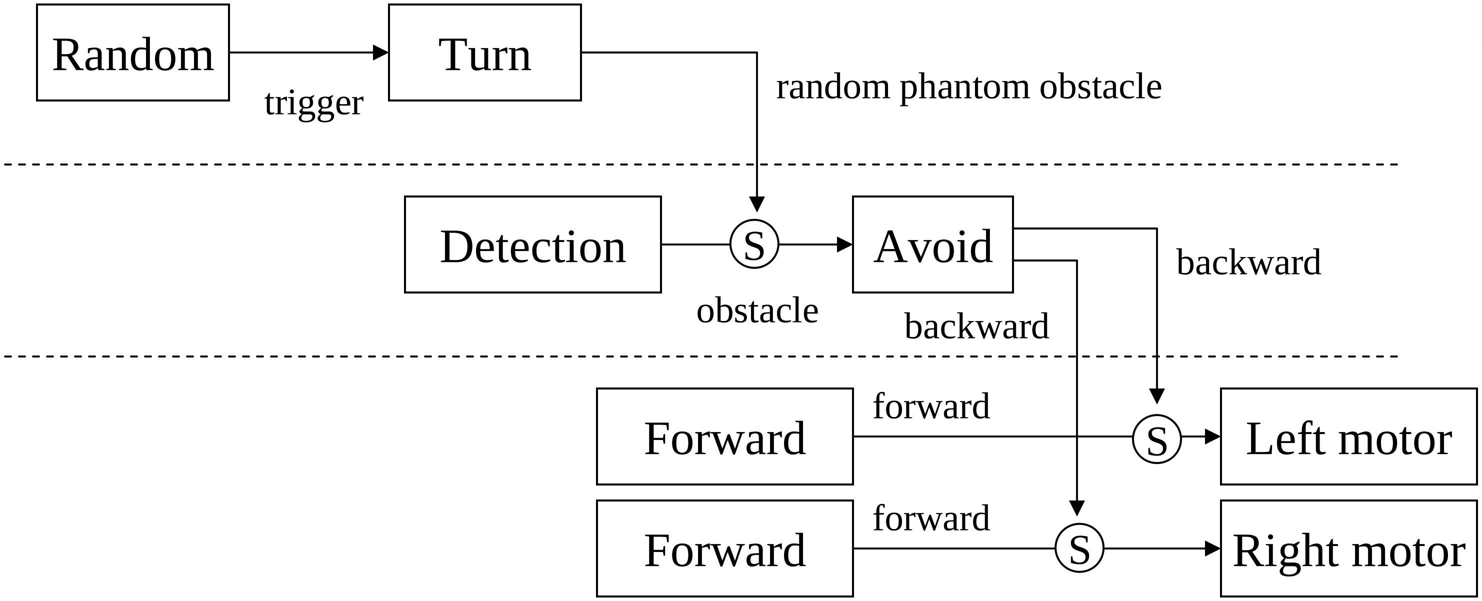
- *Then we add a layer which recognizes obstacles and while they are detected, the layer replaces messages for one wheel to backward. As a result, the robot does not collide.*





Example – step 3

- *However easily it can happen that it stays in the same region, moving in a cycle. Thus we add a layer which sometimes causes its random turn. We perform such a turn only when no obstacles are detected and we implement it just by apparent detection of obstacles*



Random

trigger

Turn

random phantom obstacle

Detection

S

obstacle

Avoid

backward

backward

Forward

forward

S

Left motor

Forward

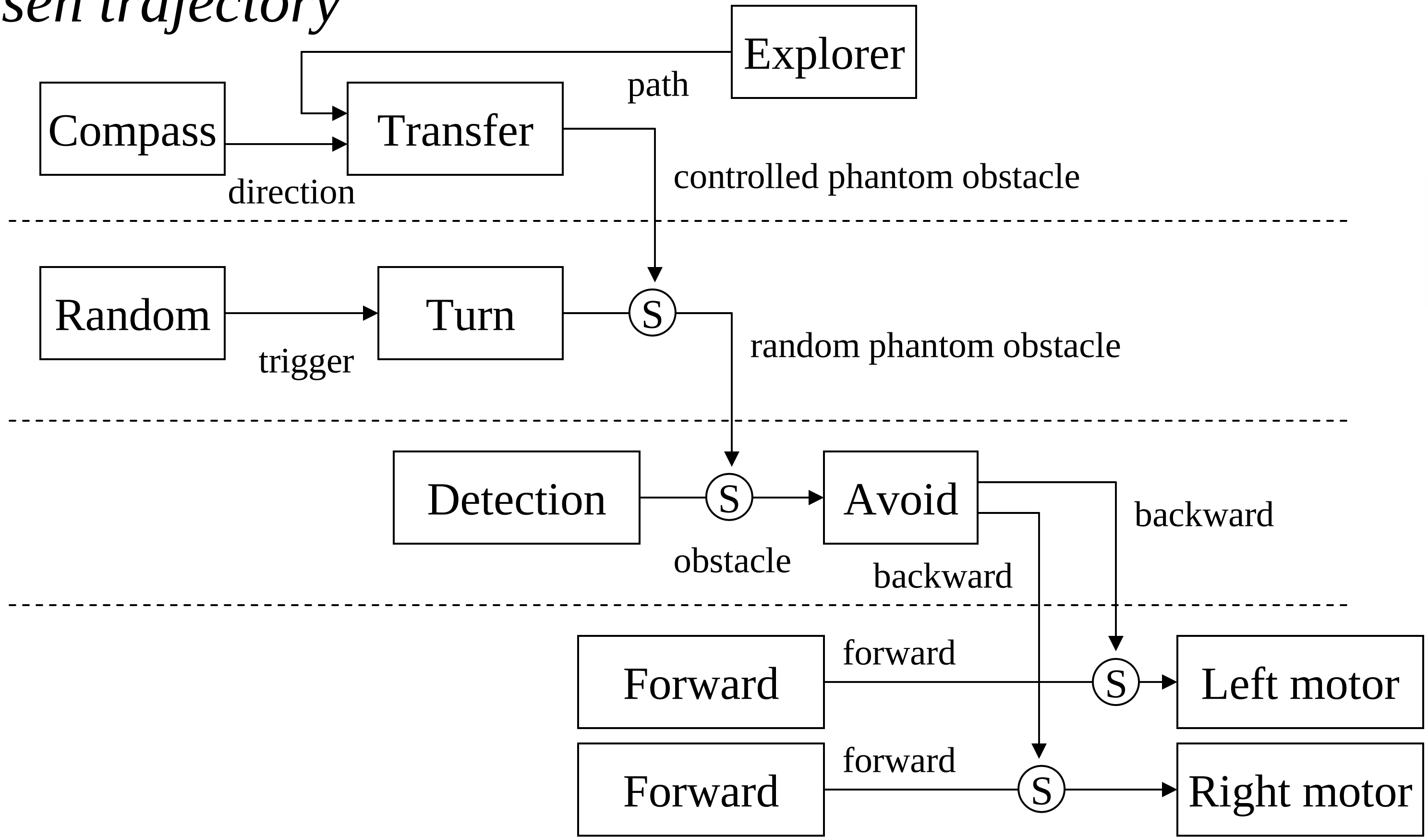
forward

S

Right motor

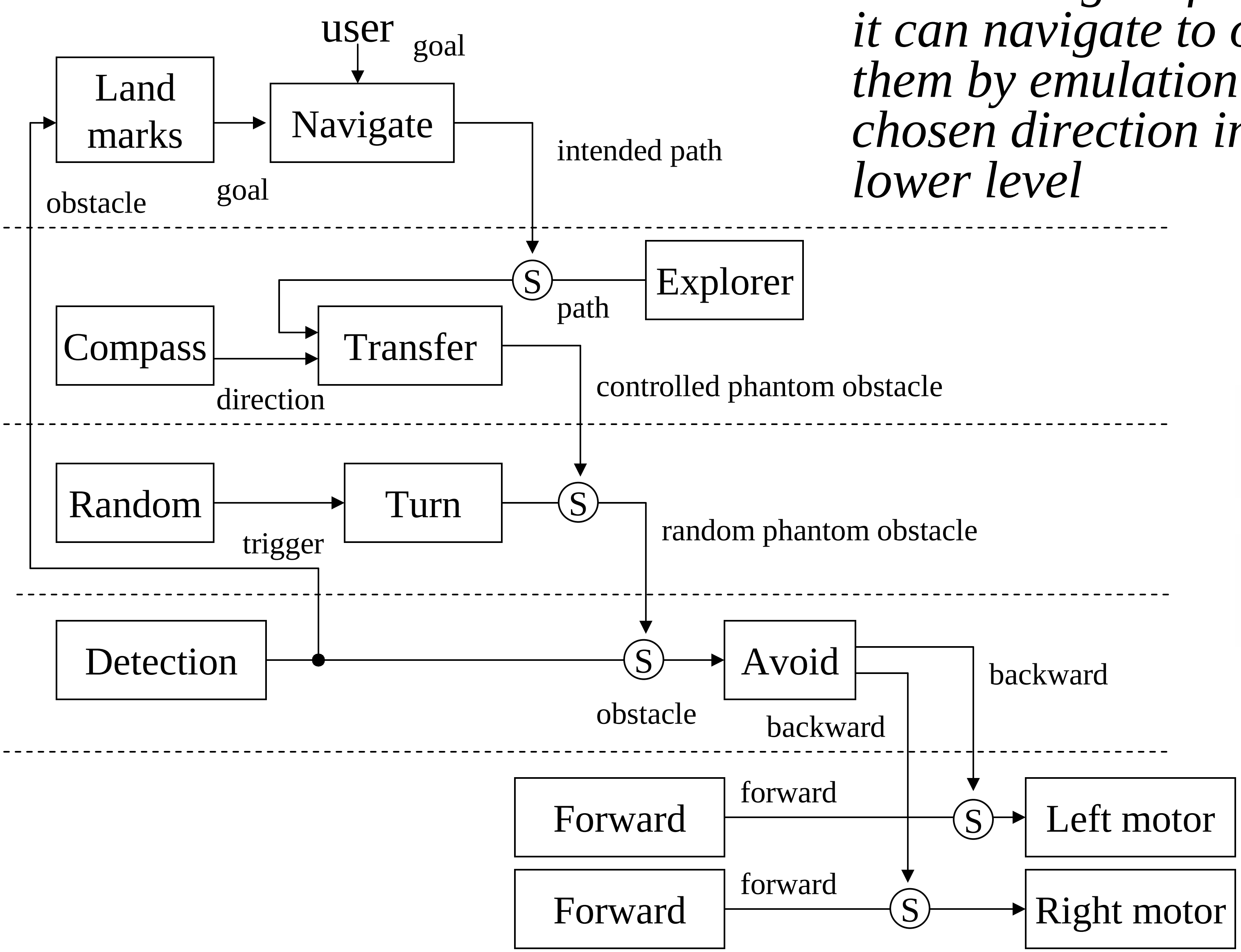
Example – step 4

- *another layer can a global movement in an absolute direction – from one part to another part of bureau. Once such direction is chosen, we implement its following by turns which are apparently random for the older layers, but in fact they keep the robot at the chosen trajectory*



Example – step 5

- *Other level can detects landmarks and having received a goal from user it can navigate to one of them by emulation of the chosen direction in the lower level*



Derivates of subsumption architecture

- *behavior-based architectures*: restriction of the influence to suppression of layer outputs (simplification)

- *fine-grained architecture*: accumulation of various actions generated by various levels is enabled (data fusion, more close to neural networks)

- many others

Derivate implemented at Comenius University, Bratislava

- Long tradition of embodied approch for engineering due to Jozef Kelemen (1992 common work with Marvin Minsky)
- *agent-space architecture*: extension of the influence potential by modernizing the architecture which **overcomes the limitations of the hardware layout** typical for the original concept

(Lucny 2004)

Conclusion

- Subsumption architecture is a biologically inspired method of development of complex systems
- Typical features: incremental development, situatedness, decentralization, influence by inhibition and suppression
- Reference: Brooks, R.: Cambrian Intelligence, MIT Press, Cambridge, 1999

Thank you !

Andrej Lúčny

Department of Applied Informatics

FMFI, Comenius University, Bratislava

lucny@fmph.uniba.sk

www.microstep-mis.com/~andy

www.robotics.sk