motivation

Robots are flexible machines.

Manufacturing requires agile robots.

Robots must interface with humans.

We want to be able to simply tell them, or show them, what we want them to do.

motivation

Robots are **flexible** machines.

Manufacturing requires **agile** robots.

Robots must **interface** with **humans**.

We want to be able to **simply tell** them,
or **show** them, **what we want them to do**.

motivation

Robots are **flexible** machines.

Manufacturing requires **agile** robots.

Robots must **interface** with **humans**.

The challenge for flexible and agile manufacturing relies on simple and natural human-machine interfaces.

We want to be able to **simply tell** them, or **show** them, **what we want them to do**.

motivation

Robots are flexible machines.

Manufacturing requires agile robots.

Robots must interface with humans.

The challenge for flexible and agile manufacturing relies on simple and natural human-machine interfaces.

We want to be able to simply tell them, or show them, what we want them to do.

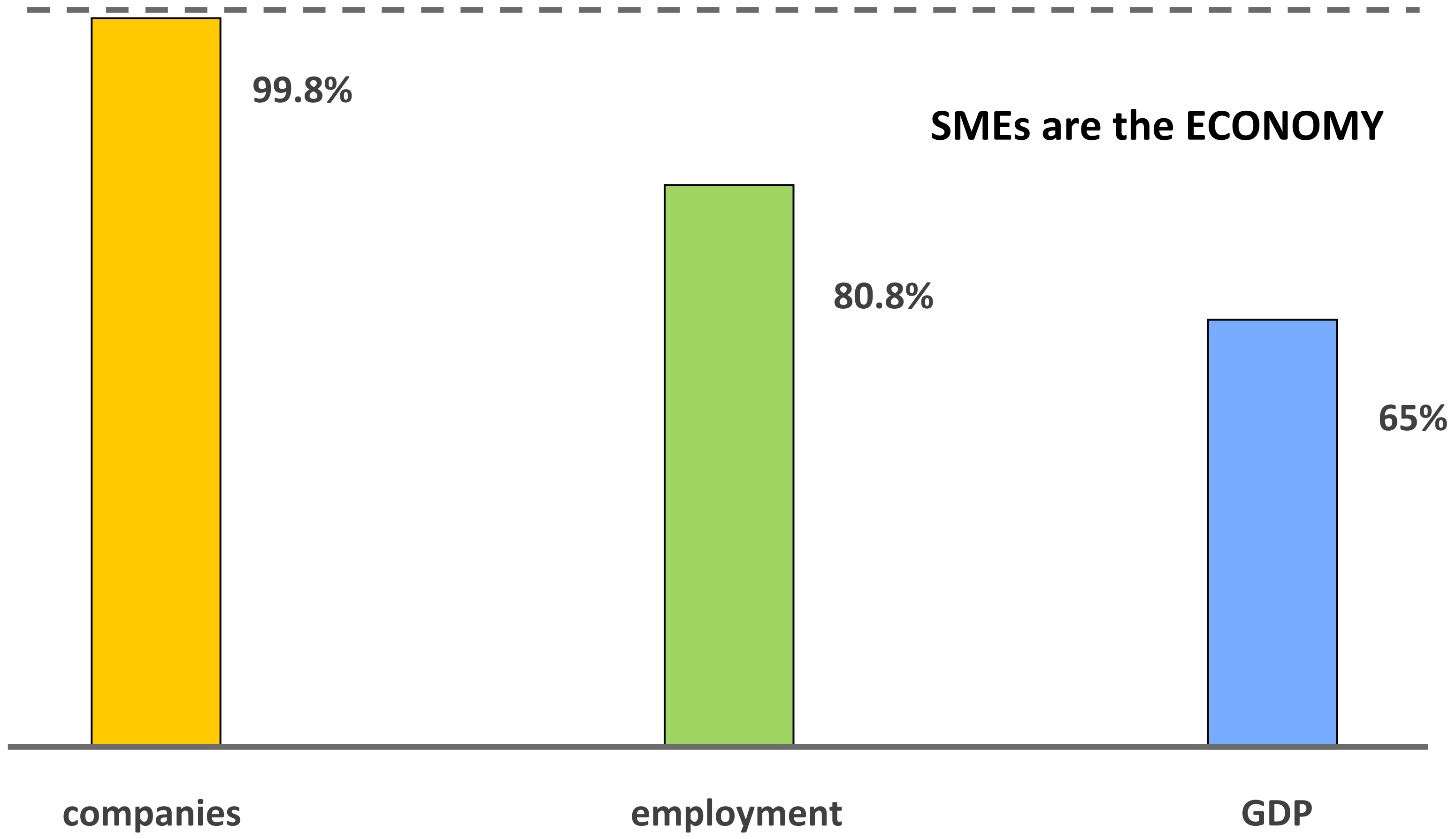
index & keywords

index

keywords

- human-machine interfaces
- productive robotics
- Programming robotic cells
- input-output devices
- devices & sensors
- programming-by-demonstration
- CAD interfaces
- SOA: service-oriented architectures
- coworker scenario
- hyperflexible cell
- cognytive factory
- speech interfaces
- code generation

challenge - think about SMEs



challenge - think about SMEs

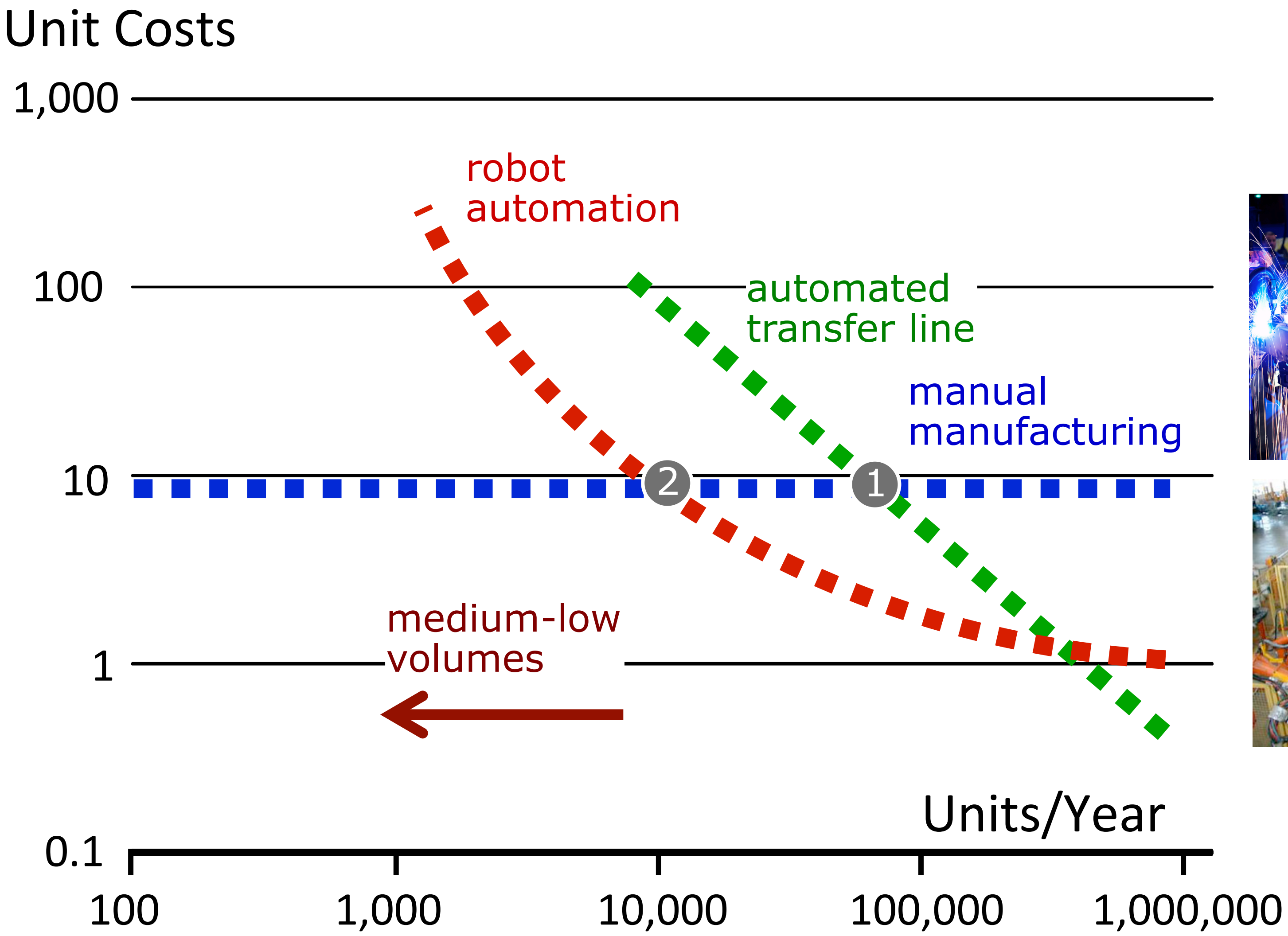
SMEs require radical new approaches:

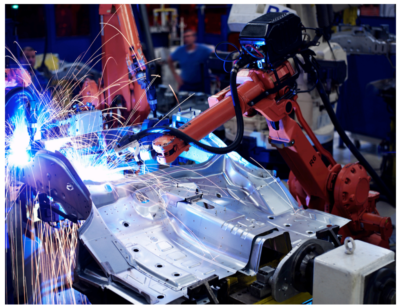
- business models
- flexibility and agility
- integration
- robot automation
- HMI

Everything is different:

- people
- requirements
- resources

challenge - think about SMEs





introduction

flexibility is the key.

the need is here…

Industrial robots are considered as a cornerstone of competitive manufacturing which aims to combine **high productivity**, **quality** and **adaptability at minimal cost**.

In 2007 more than one million industrial robot installations were reported with automotive industries the predominant users with a **share** of **more than 60%**.

However, **high-growth-industries** (in life-sciences, electronics, solar cells, food, and logistics) and **emerging manufacturing processes** (gluing, coating, laser-based processes, precision assembly etc.) **will increasingly depend on advanced robot technology.**

The production of industrial robots on the one hand, and the planning, integration and operation of robot workcells on the other hand are **largely independent engineering tasks**. In order to be produced in sufficiently large quantities, a **robot design should meet the requirements for the widest set of potential applications**.

Since this is **difficult in practice**, we can find **specially designed robots** for assembly, handling, welding, painting, palletizing, machining, etc.

but… special robots?

introduction

too much engineering…
too much code …
too much time

4 times

Generally, a **robot workcell** consists of one or more robots with controllers and robot peripherals: grippers or tools, safety devices, sensors, and material transfer components for moving and presenting parts. Typically, the **cost of a complete robot workcell is typically four times the cost of the robots alone**.

radical change is needed…

A **robot workcell** is usually the **result** of customized **planning**, **integration**, **programming and configuration**, requiring **significant engineering expertise**. Standardized engineering methods, tools, and best practice examples have become available to reduce costs and provide more predictable performance .

Today's industrial robots are mainly the result of the requirements of capital-intensive large-volume manufacturing, mainly defined by automotive, electronics and electrical goods industries. **Future industrial robots** will not be a mere extrapolation of today's designs with respect to features and performance data, but will rather follow new design principles addressing a wider range of application areas and industries. At the same time, new technologies, particularly from the IT world, will have an increasing impact on the design, performance and cost of future industrial robots.

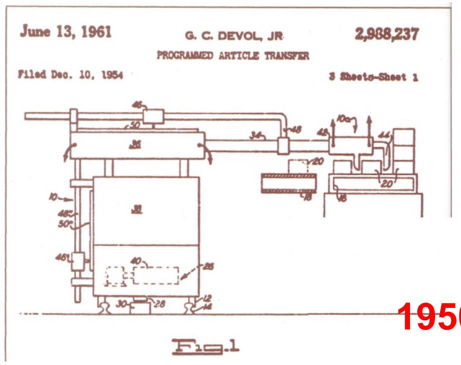
International and national standards now help to quantify robot performance, define safety precautions, geometry and media interfaces.

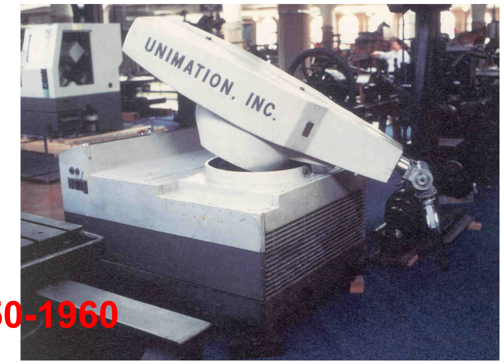
Most robots operate **behind secure barriers** to keep people at a safe distance.

Recently, improved safety standards have **allowed direct human-robot** collaboration **enabling robots and human factory workers** to share the same space in safety.

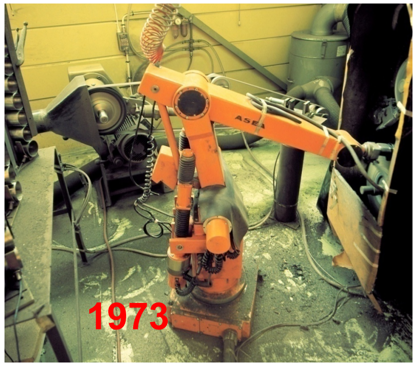
robots and people are usually away from each other…

brief history

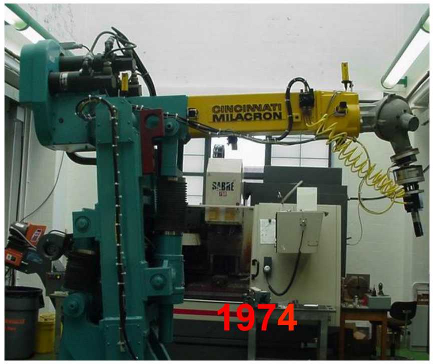




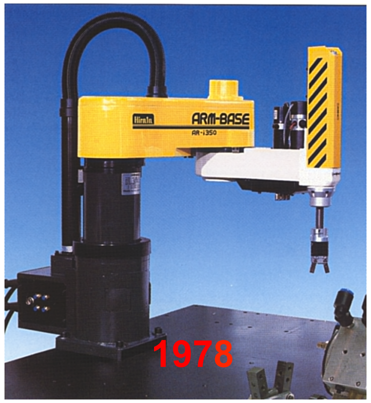
1950-1960



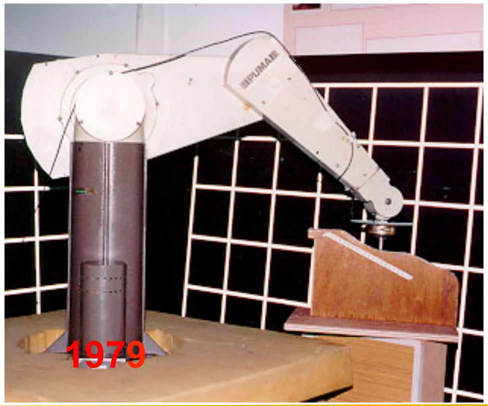
1973



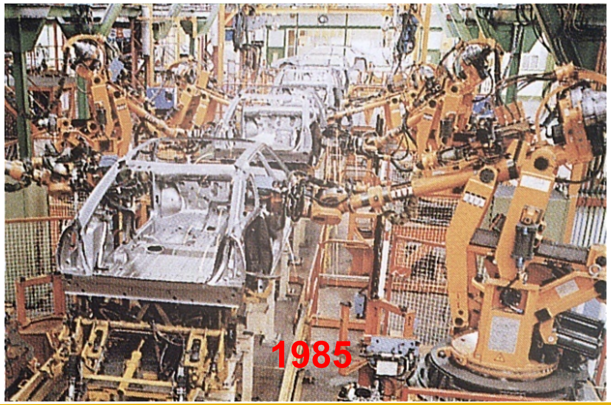
1974



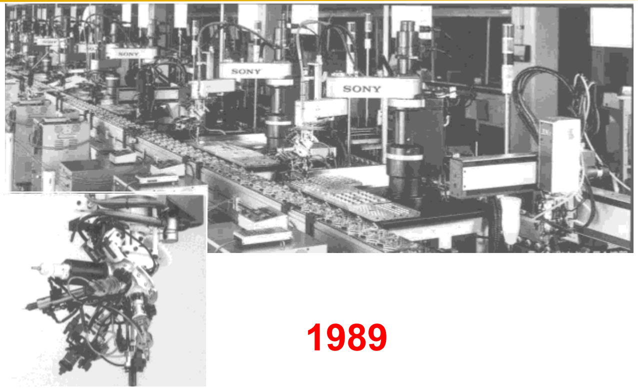
1978



1979



1985



1989

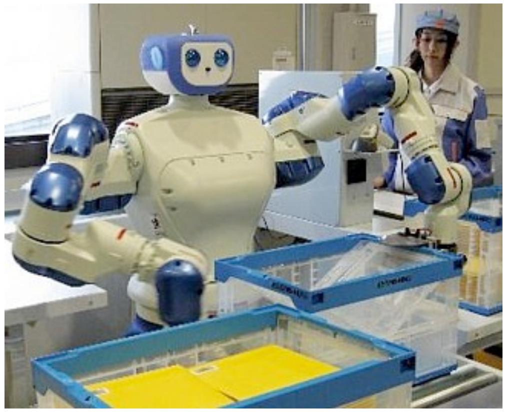
1999

2005

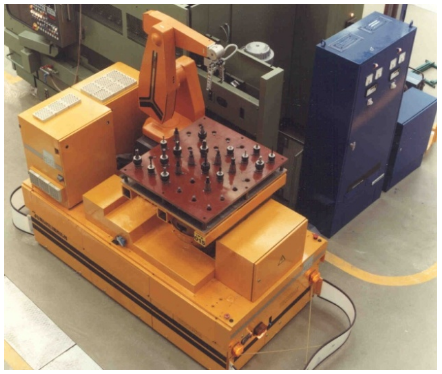
2008

© j. norberto pires 2012

brief history



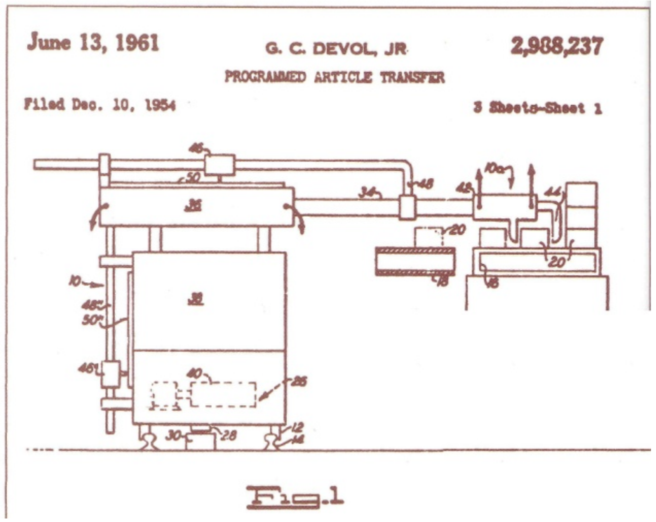
Automated Guided Vehicles (AGV)

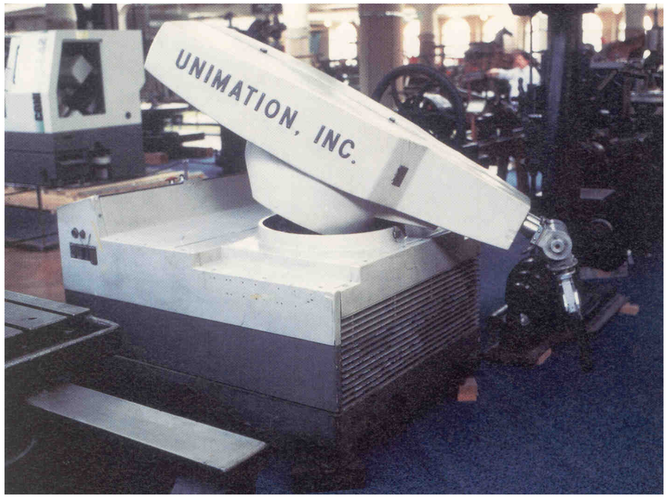


brief history

The invention of the industrial robot dates back to **1954** when George Devol filed a patent on a *Programmed Article Transfer*.

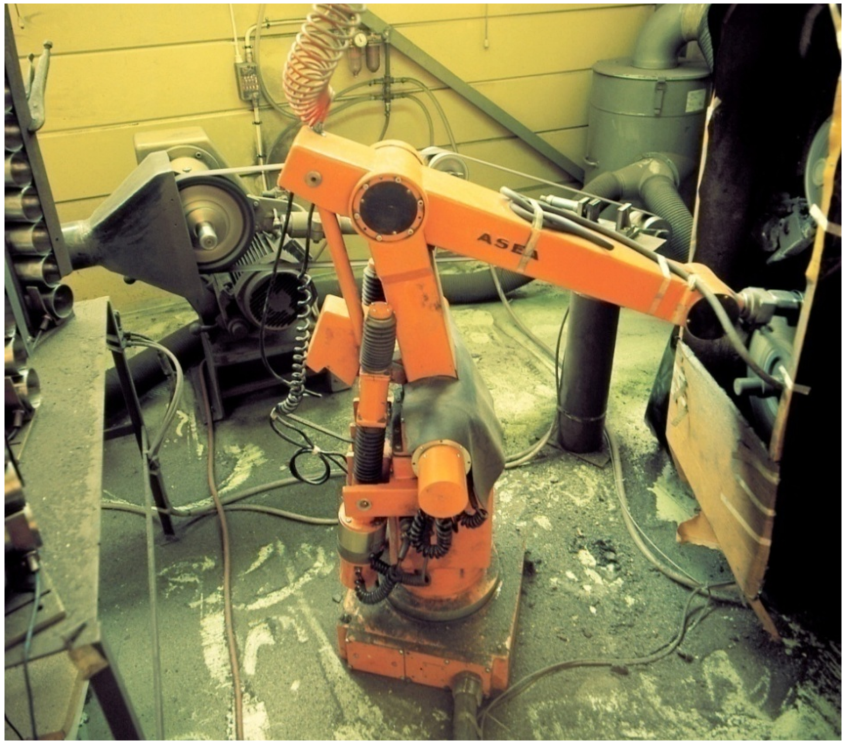
After teaming up with Joseph Engelberger, the first robot company, **Unimation**, was founded which put the first robot into service at a General Motors plant in 1961 for extracting parts from a die casting machine. Most of the hydraulically actuated “Unimates” were sold through the following years for workpiece handling and for spot-welding of car bodies. **An innovation driven industry was born**. However, it took **many years until** this industry became **profitable**.



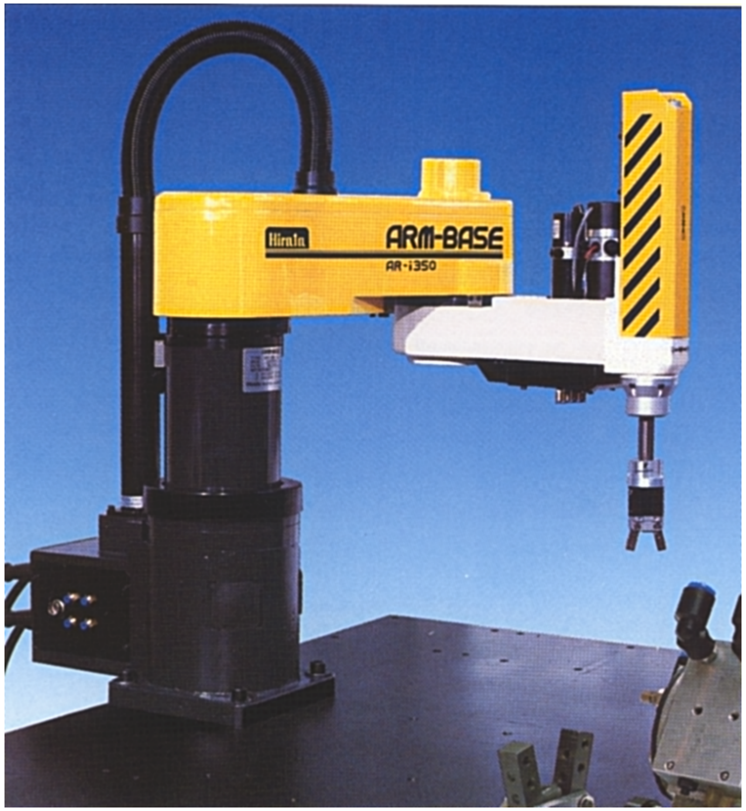


brief history

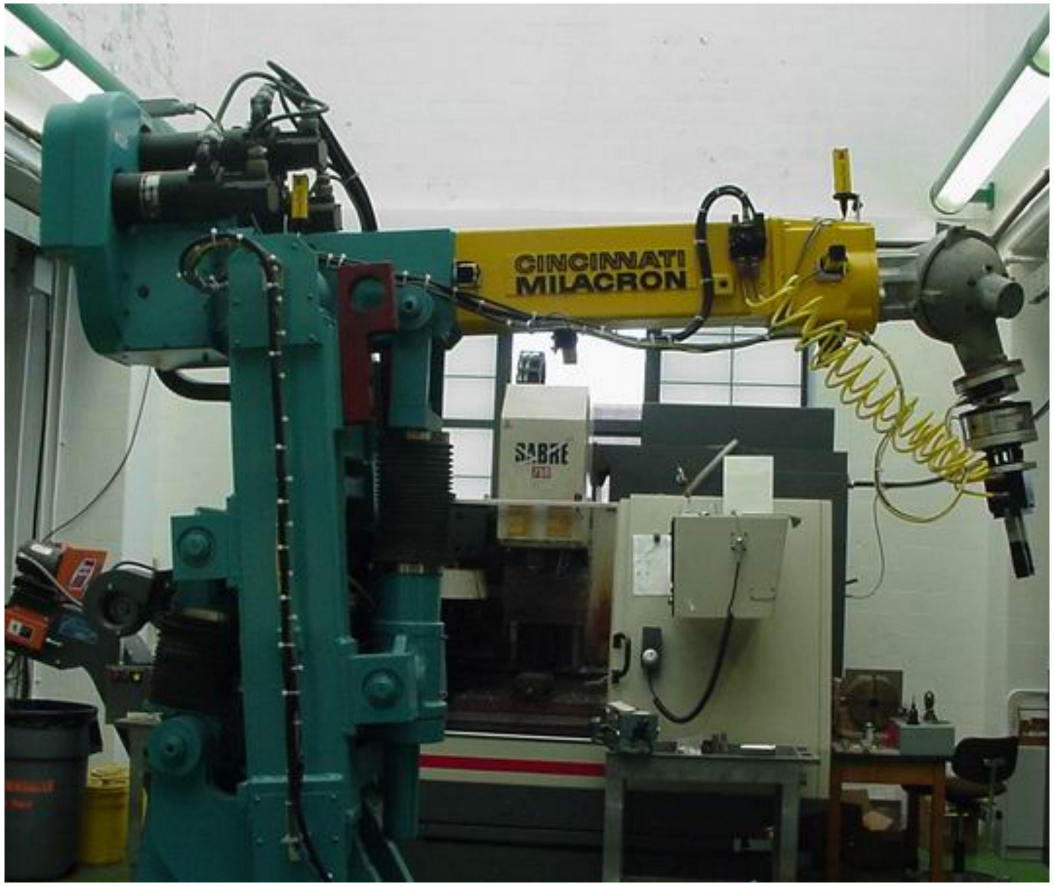
First introduced in **1973**, the **IRB-6** has been a **breakthrough** development as it was the **first serial robot** product which combined **all-electric-drives** technology and a **microcomputer** for programming and motion control. **The robot proved very robust**. **Life-times of more than 20 years** in harsh productions were **reported**.



The *Selective Compliance Assembly Robot Arm* (**SCARA**) is particularly suited for assembly tasks as it combines rigidity in the vertical axis and compliance in the horizontal axis. In **1978**, the first Hirata AR-300 was put together. The SCARA design combines three or four rotational and one translational axis.

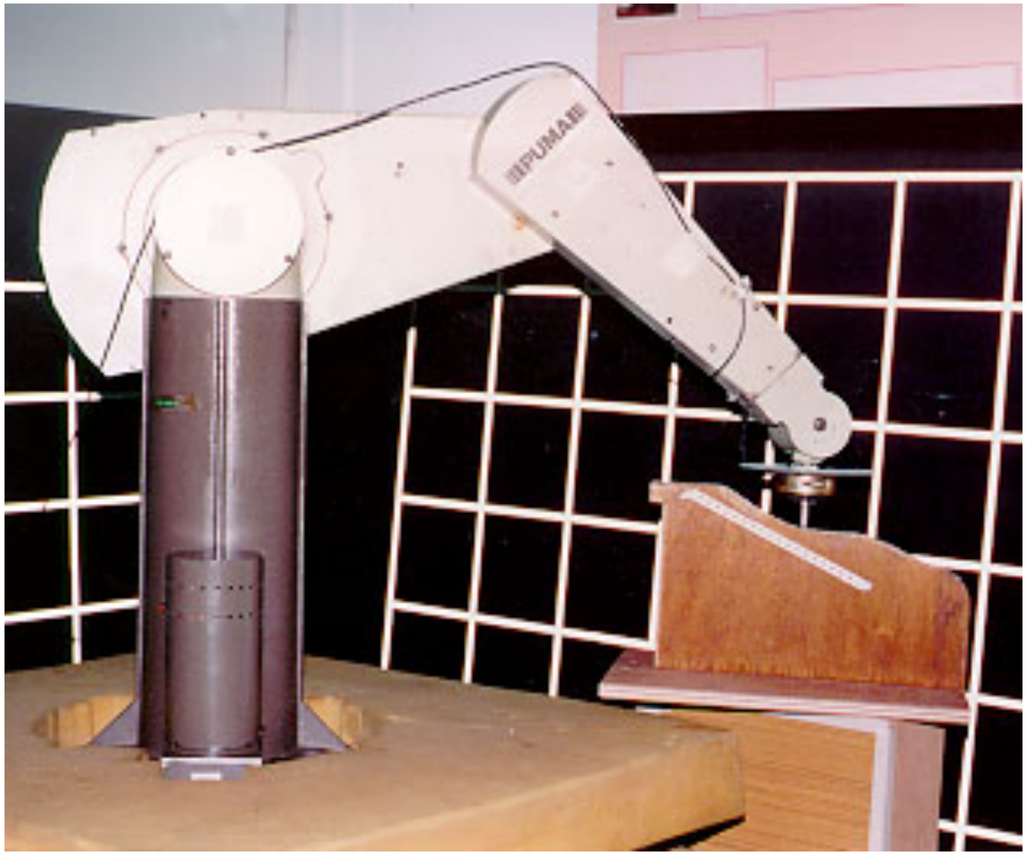


In **1974**, Cincinnati Milacron introduced the first microcomputer controlled robot. The first T3 (**"The Tomorrow Tool")** models used **hydraulic drives**, later they were replaced by **electric motors**. The CM robotics division was acquired by ABB in the late 1970s.



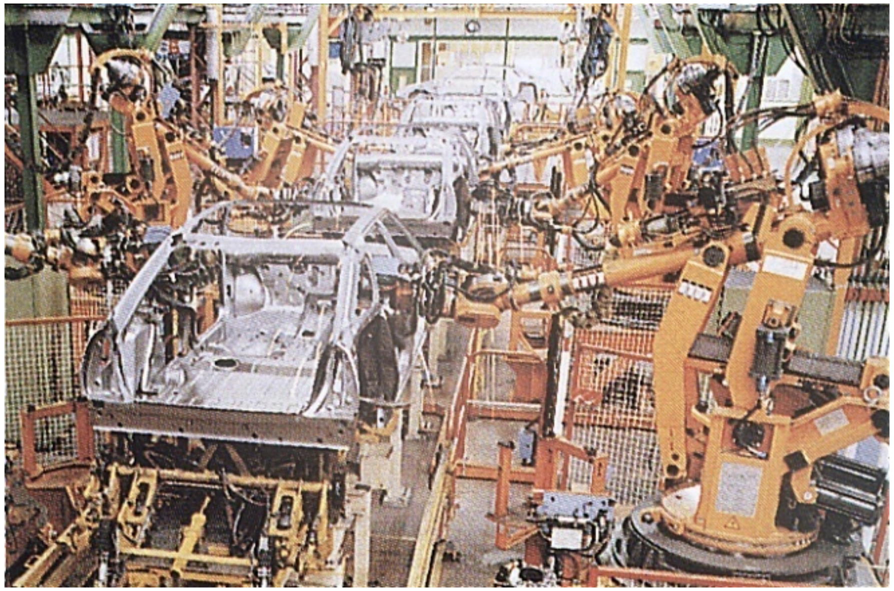
brief history

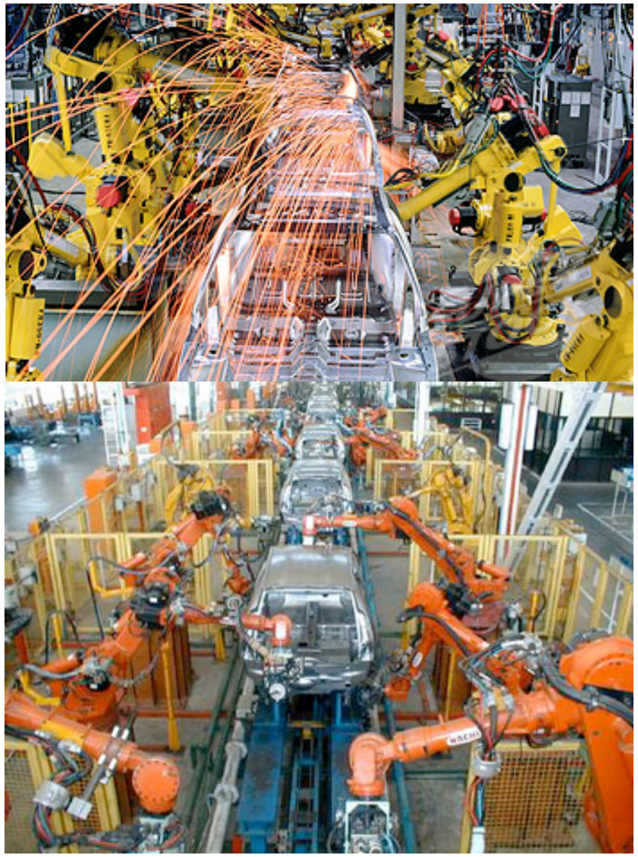
This 6 axis **PUMA** (Programmable Universal Machine for Assembly) came close to the dexterity of a human arm. After its launch in **1979** by Unimation it became one of **the most popular arms and was used as a reference in robotics research for many years.**



brief history

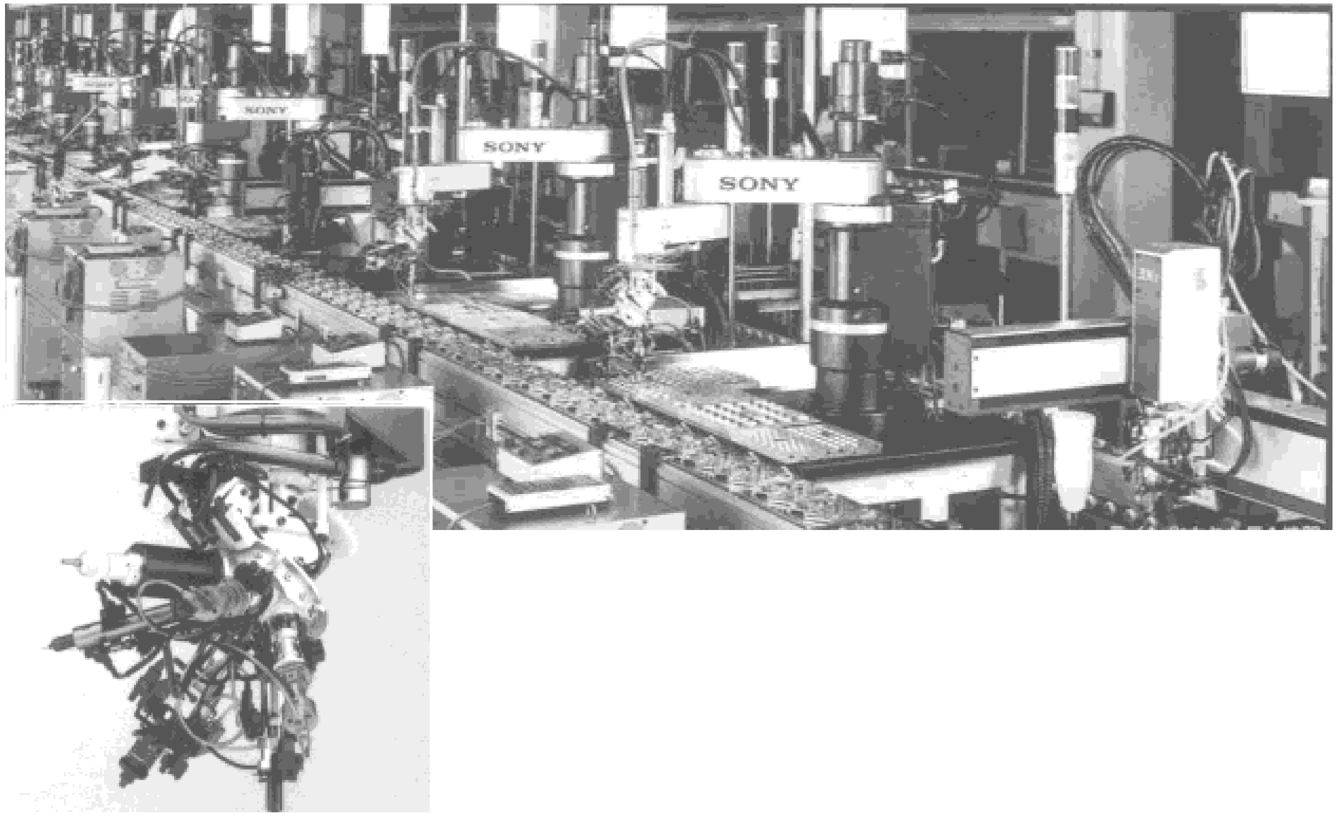
Spot welding quickly became a primary application for robots as these **jobs were particularly exhausting and hazardous for workers**. A typical car body welding line from **1985** is displayed. The car model shown is a French Citroën CX.





brief history

An automated VCR assembly line (ca. **1989**) with SCARAs carrying a turret with multi-gripper tools. Typically five parts are added by one robot before the VCR is moved to the next station of the **fully automated assembly line**.



brief history

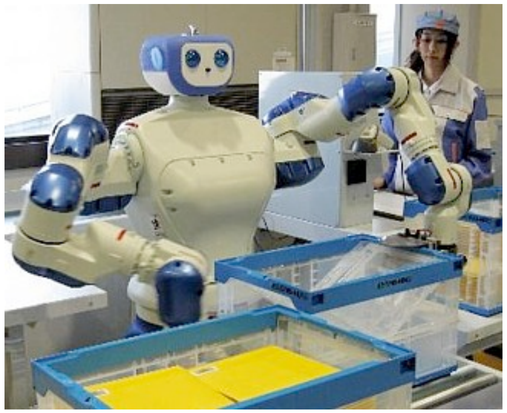
Parallel kinematic machines (PKM) represent an interesting approach to achieve **high stiffness** at **low inertia**, thus allowing **accurate**, **high speed motions**. Initially suggested by Clavel this 4 axis robot is used for high speed pick-and-place tasks. The robot reaches **accelerations of up to 10 g**.

brief history

The *KUKA light-weight robot (LWR)* is the result of a long research and development process towards an arm design with a **weight-to-payload ratio of 1:1**. The **7 axis arm**, which is suited for human-robot cooperation, imitates the **dexterity of a human arm**.

The **roboLoop** of Güdel is a curved-track gantry and transfer system. **One or more robot arms** circulate as carriers in a closed system. The system can be installed suspended, in gantry configuration, or as a floor-standing system. **A signal bus allows the control and coordination of multiple robo-carriers.**

Motoman's DA-20 dual-arm (family) robot provides **high-speed motion with two six-axis arms** (13 axis) that provide enhanced, **human-like flexibility** of movement. The robot also provides jig-less operation with one robot arm holding part while the other performs operations on the held part. The *7-axis IA20* is a powerful arm for assembly operations. The *SDA10D dual arm* (15 axis) robot system provides **human-like flexibility** of movement and **fast accelerations**.


DA-20

IA-20

SDA10D

brief history

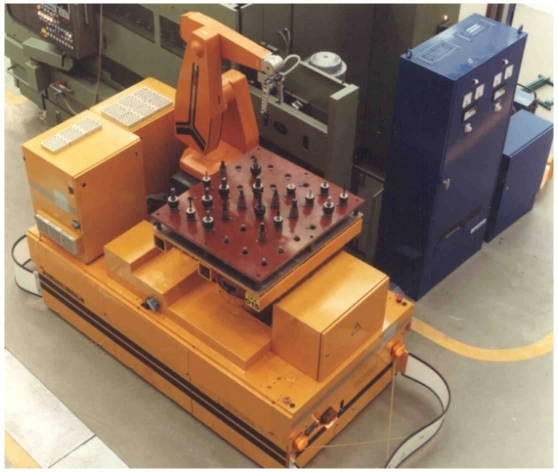
A **huge number** of independent small manufacturers are offering **small**, open, **easy to use robot arms** with **customer selected number of joints**, **force-control**, **sensors**, etc. For example, Universal Robots (Denmark) and Barrett (USA).

Universal Robots (Denmark)

Barrett (USA)

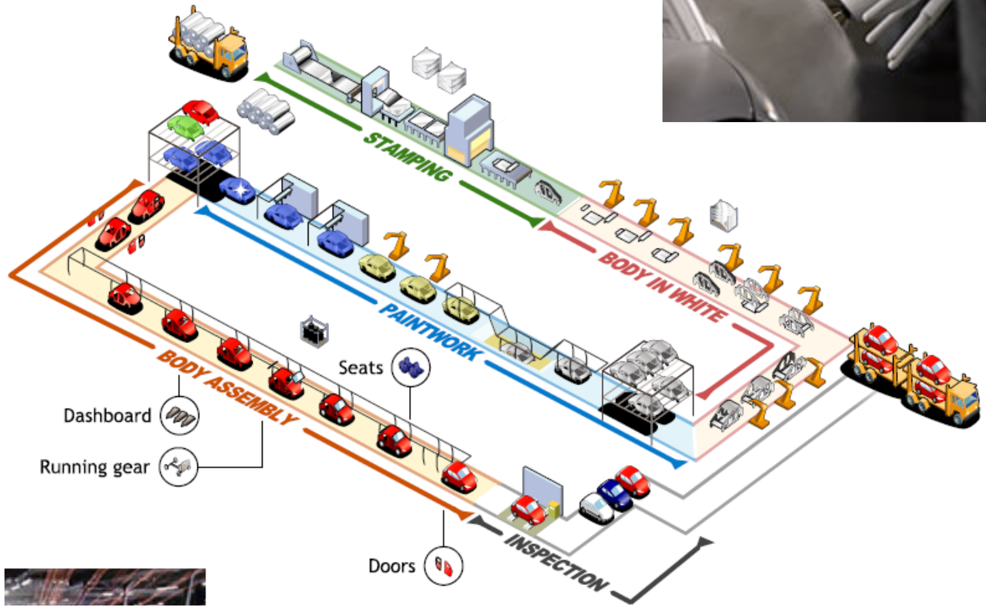
brief history

In parallel to industrial robots Automated Guided Vehicles (AGVs) have emerged. These mobile robots are used for moving workpieces or loading equipment from point to point. Within the concept of automated flexible manufacturing systems (FMS) AGVs have become an important part of their routing flexibility. Initially AGVs **relied on prepared floors such as embedded wires or magnets for motion guidance**. Meanwhile, freely navigating AGVs are used in large scale manufacturing and logistics. Usually, **their navigation is based on laser scanners which provide an accurate two-dimensional map of the actual environment for self localization and obstacle avoidance**. Early on **combinations of AGVs and robot arms were realized to automatically load and unload machine t**ools**.** Only in some **selected environments** such as (un-)loading process equipment in the semiconductor industry, these mobile arms **were economically advantageous**.



typical applications

- Welding
- Painting
- Car body assembly
- Material transfer automation
- …



main trends

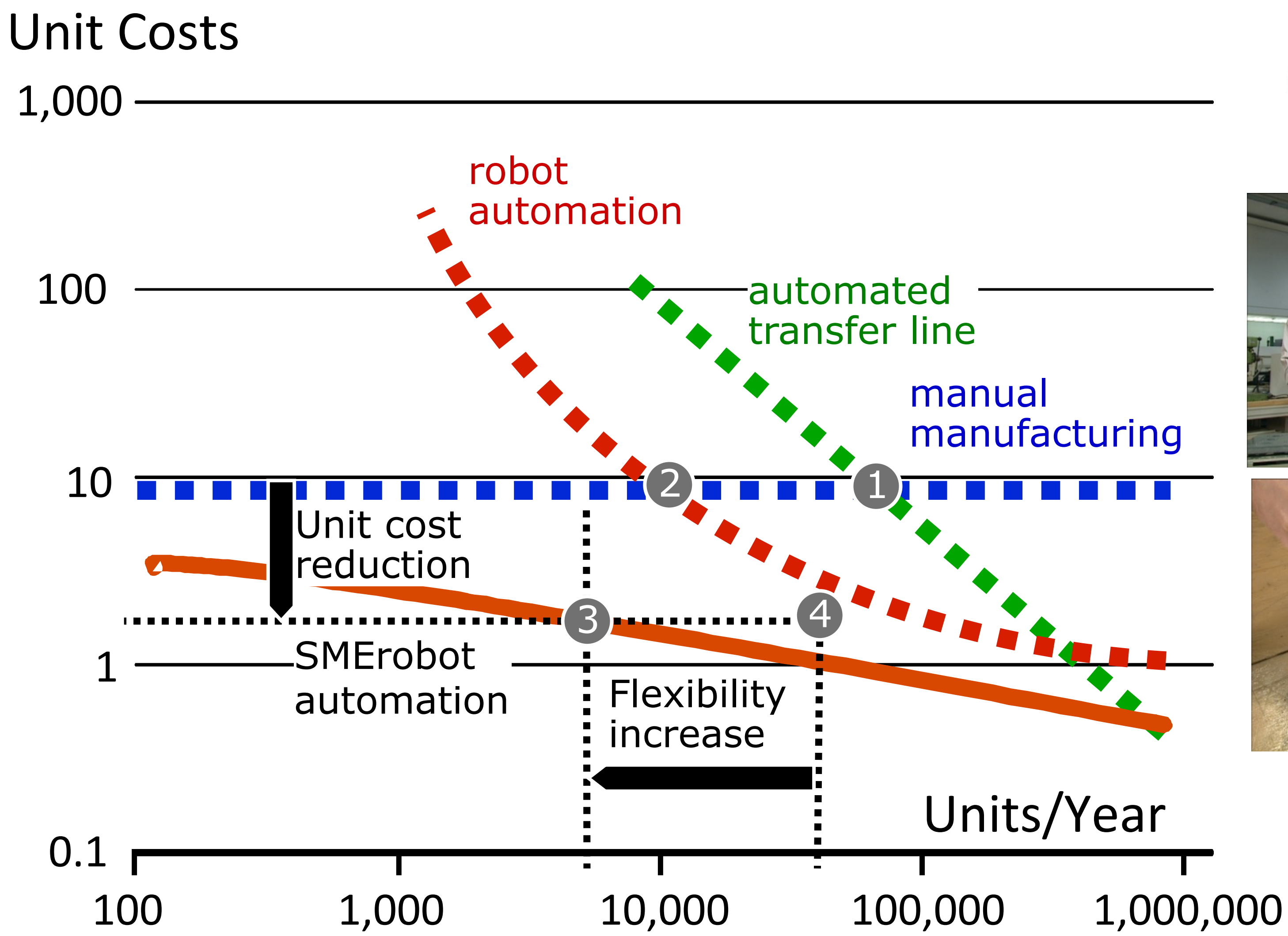
By 2007, the evolution of industrial robots was marked by the following main trends:

- **The average robot unit price fell to about one third of its equivalent price in 1990 which means that automation is becoming more affordable.**
- **At the same time, the robot performance such as speed, load capacity, mean-time-between-failure (MTBF) has dramatically improved.** These improvements provide a **faster return on investment**, particularly for **small**, **short-run batch production**.
- **Off-the-shelf components from PC technologies, consumer software and the IT industry have contributed to improved performance-cost ratios.** Today, most manufacturers integrate PC- based processors in their controllers as well as PC-related software for programming, communication, simulation, and maintenance.
- **Multiple robots can be programmed and synchronized in real-time by one controller** which allows robots to precisely cooperate on a single workpiece.
- Increasingly, **vision systems** for object identification, localization, and quality control become an integral part of the robot controller.
- **Robots are networked** by fieldbuses or Ethernet for control, configuration, and maintenance.

main trends

- **New financing arrangements** allow end-users to rent a robot or even have a robot workcell operated by a specialized company or even the robot supplier in order to reduce risks or to save on investment capital.
- **Training and education programmes** have become important services to the end-users to increase acceptance of robot technology. Specific multimedia material and courses aim at educating industrial engineers and workforce to effectively plan, operate and maintain industrial robot workcells.

challenge - think about SMEs



SME robot

discussion

Robots are extensively used in large-volume market industries (like automotive and consumer electronics) to execute all sorts of operations. Consequently, robotics development was mainly driven by the needs of these high-volume industries, **which resulted in machines and systems less adapted to the needs of smaller, knowledge-based and innovation-driven businesses**.

Those companies, which constitute the **vast majority** of the European industrial tissue in **terms of turnover** and **employment**, **require less complex systems**, **easier to program and operate by non-skilled operators**, much more **flexible** and **agile**.

And this **means radical new approaches** in the **design**, **development**, **distribution** and **support** of the **future robotic systems**.

But it also requires the **adoption of manufacturing scenarios**, across all sorts of industries and business dimensions, where **maximum flexibility**, **productivity** and **agility** is obtained with **major reductions in the life-cycle costs of manufacturing equipment and skilled personnel**.

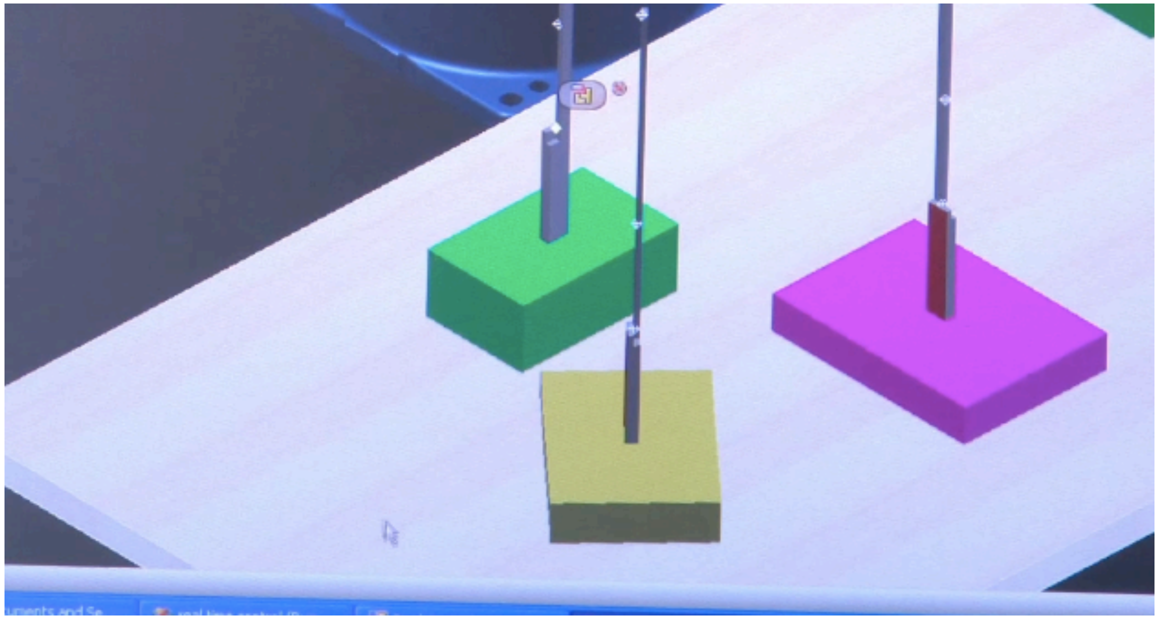
discussion – application pull

New families of robots need to respond to the needs of smaller businesses**, enabling integrators to easily develop robotic manufacturing solutions specially adapted to low-volume and emerging markets (*application pull*)**, i.e., solutions that can be **designed**, **installed**, **supported** and **maintained** at much **lower costs**, but also **easily operated by regular non-skilled personnel**.

Example: Universal Robots

discussion – technology push

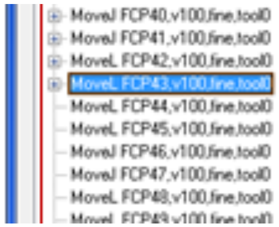
New families of robots need **also to interface better with humans and adapt easily to working conditions and setups**. This means that novel technologies from the world of ICT, sensors and electronic consumer markets will have a major influence on the design and performance of new robotic automation systems (***technology push***).



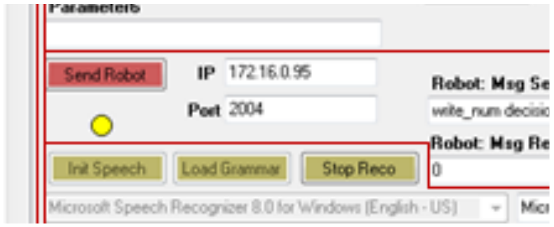
Example: CAD programming, devices and speech

discussion – creativity push/pull

The adoption of scenarios where humans and robots share the same workspace and need to cooperate to achieve a common goal (**co-worker scenario**), and/or where robots are designed to make decisions as the common task unfolds (**cognitive factory scenario**) requires machines that can incorporate knowledge from other non-technical disciplines, can interface with all sorts of sensors and other machines, can operate more or less autonomously responding to sensor information in accordance with built-in process knowledge, i.e., **machines that are hyper-flexible, open, fully based on standards, very easy to deploy and integrate into IT environments (plug-and-play), safe to integrate (human augmentation), easy to instruct interactively using human-like mechanisms (programming-by-demonstration instead of regular code writing), flexible to environment and component geometry, capable of high forces and moments without compromising safety, very easy to setup and start working (plug-and-produce) and adapted to customer needs (*creativity push/pull*)**.



```
MoveJ FCP40,v100,fine,tool0
MoveJ FCP41,v100,fine,tool0
MoveL FCP42,v100,fine,tool0
MoveL FCP43,v100,fine,tool0
MoveL FCP44,v100,fine,tool0
MoveL FCP45,v100,fine,tool0
MoveJ FCP46,v100,fine,tool0
MoveJ FCP47,v100,fine,tool0
MoveL FCP48,v100,fine,tool0
MoveL FCP49,v100,fine,tool0
```



IP 172.16.0.95
Port 2004
Robot: Msg Se
write_num decisio
Robot: Msg Re
0
Microsoft Speech Recognizer 8.0 for Windows (English - US)
Mic

Example: Programming-by-demonstration

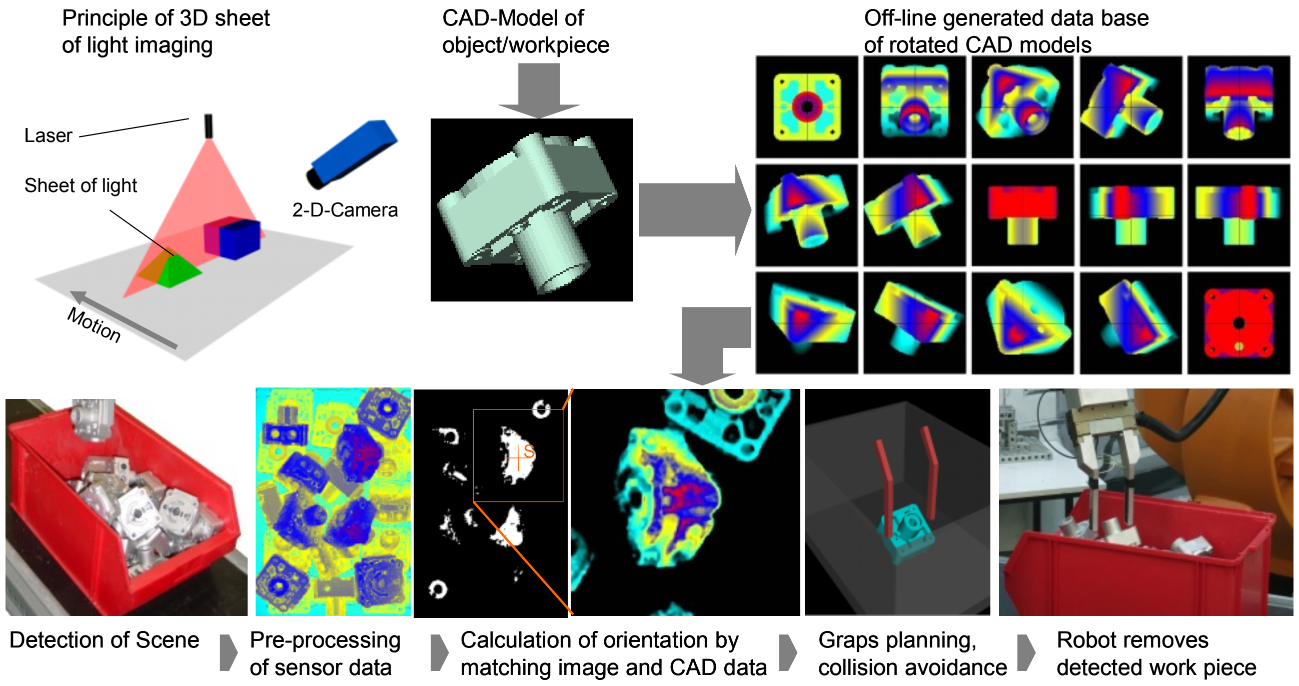
programming

We cannot yet instruct a robot in the same way that one would instruct a human worker how to carry out a task. More specifically (*wish-list*) we would like to teach robots by:

- **Manually guiding the robot** to the positions of interest, or even along the desired paths or trajectories if human accuracy is enough.
- Having simple ways to make use of **CAD** data whenever available.
- **Using different complementary modalities** (paths of communication between the human and the robot), such as speech and gestures.
- **Choreographing the task movements**, for instance loops and conditions, without requiring extensive programming competencies.
- **Means of describing acceptable variation**, e.g. as expected or normal deviations from the nominal path.
- **Specification of how external sensing should be used for new types of motions or for handling unknown variation**.

non-traditional: been picking

A scene containing the objects is acquired by a 3D-sensor, e.g., based on the laser triangulation principle. Beforehand the **CAD-object-model** has been virtually turned in discrete spatial angles in an off-line calculation. Feature histograms for each view are generated and stored in a database. **A best match** between actual feature histograms and the simulated sets of histograms **determines the location of the object**. A grasp has to be selected and a **collision-free trajectory is generated**. A typical cycle time of a location process is between 1 and 3 s.



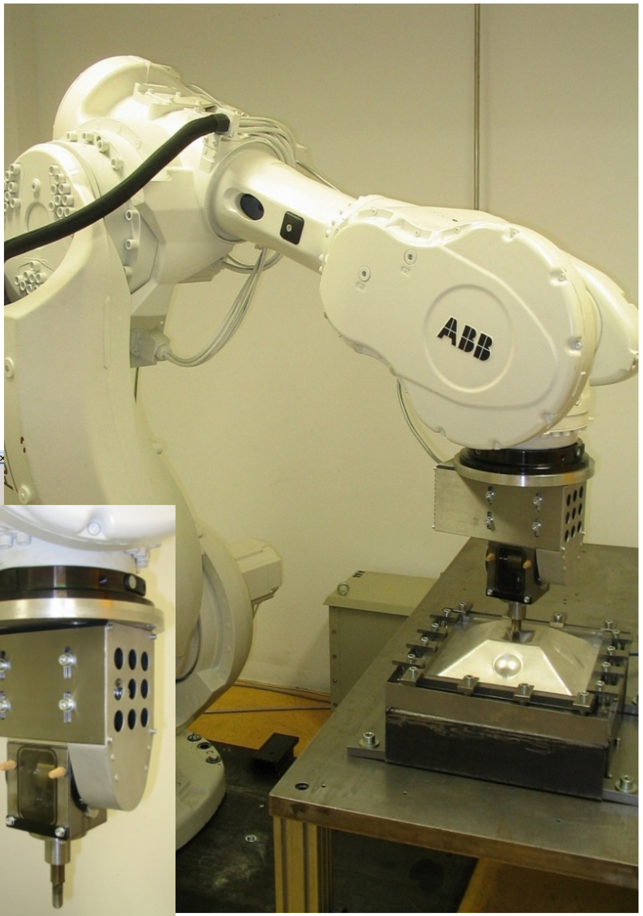
Principle of 3D sheet of light imaging
Laser
Sheet of light
2-D-Camera
Motion
CAD-Model of object/workpiece
Off-line generated data base of rotated CAD models
Detection of Scene
Pre-processing of sensor data
Calculation of orientation by matching image and CAD data
Graps planning, collision avoidance
Robot removes detected work piece

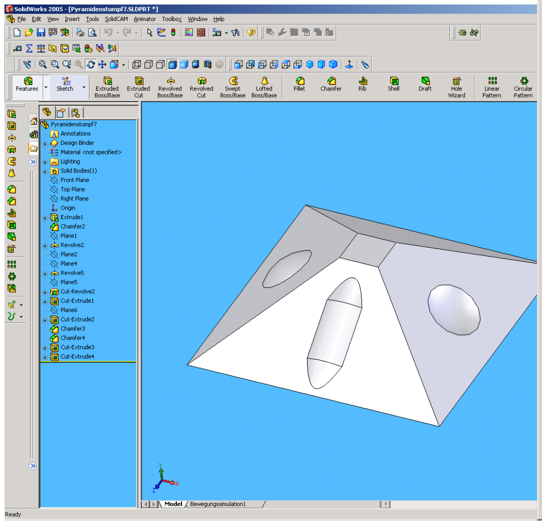
Fraunhofer IPA

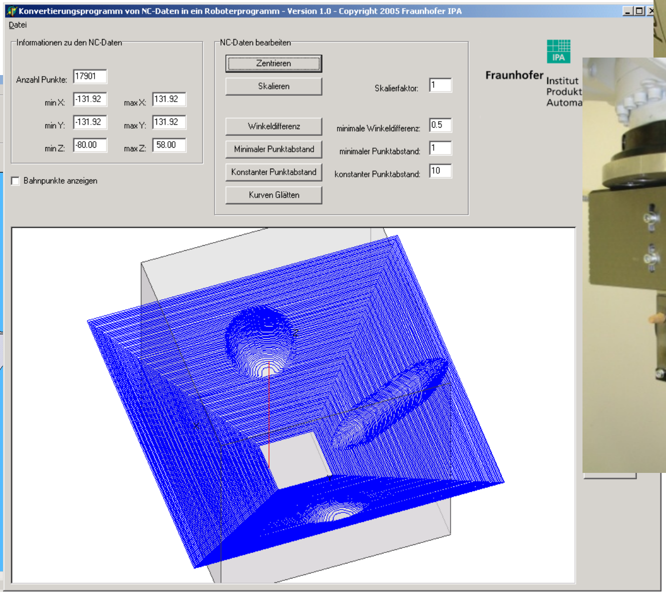
non-traditional: machinning

In this example, the forming process of metal sheets is based on an oscillating stamp (amplitude 1 mm, 50 Hz frequency) which locally plastifies the metal in incremental steps. From the CAD model (top left), the robot's trajectories are calculated on the basis of specific material models (bottom left). Each line represents a part of the tool trajectory. The robot's program is calculated automatically off-line and communicated to the robot controller when required.

Fraunhofer IPA




SolidWorks 2005 - [Pyramidenstumpf7.SLDPRT *]
File Edit View Insert Tools SolidCAM Animator Toolbox Window Help
Features Sketch Extruded Boss/Base Extruded Cut Revolved Boss/Base Revolved Cut Swept Boss/Base Lofted Boss/Base Fillet Chamfer Rib Shell Draft Hole Wizard Linear Pattern Circular Pattern
Pyramidenstumpf7
Annotations
Design Binder
Material <not specified>
Lighting
Solid Bodies(1)
Front Plane
Top Plane
Right Plane
Origin
Extrude1
Chamfer2
Plane1
Revolve2
Plane2
Plane4
Revolve5
Plane5
Cut-Revolve2
Cut-Extrude1
Plane6
Cut-Extrude2
Chamfer3
Chamfer4
Cut-Extrude3
Cut-Extrude4
Model
Bewegungssimulation1
Ready



Konvertierungsprogramm von NC-Daten in ein Roboterprogramm - Version 1.0 - Copyright 2005 Fraunhofer IPA

Informationen zu den NC-Daten

Anzahl Punkte: 17901

min X: -131.92 max X: 131.92

min Y: -131.92 max Y: 131.92

min Z: -80.00 max Z: 58.00

☐ Bahnpunkte anzeigen

NC-Daten bearbeiten

Skalierfaktor: 1

minimale Winkeldifferenz: 0.5

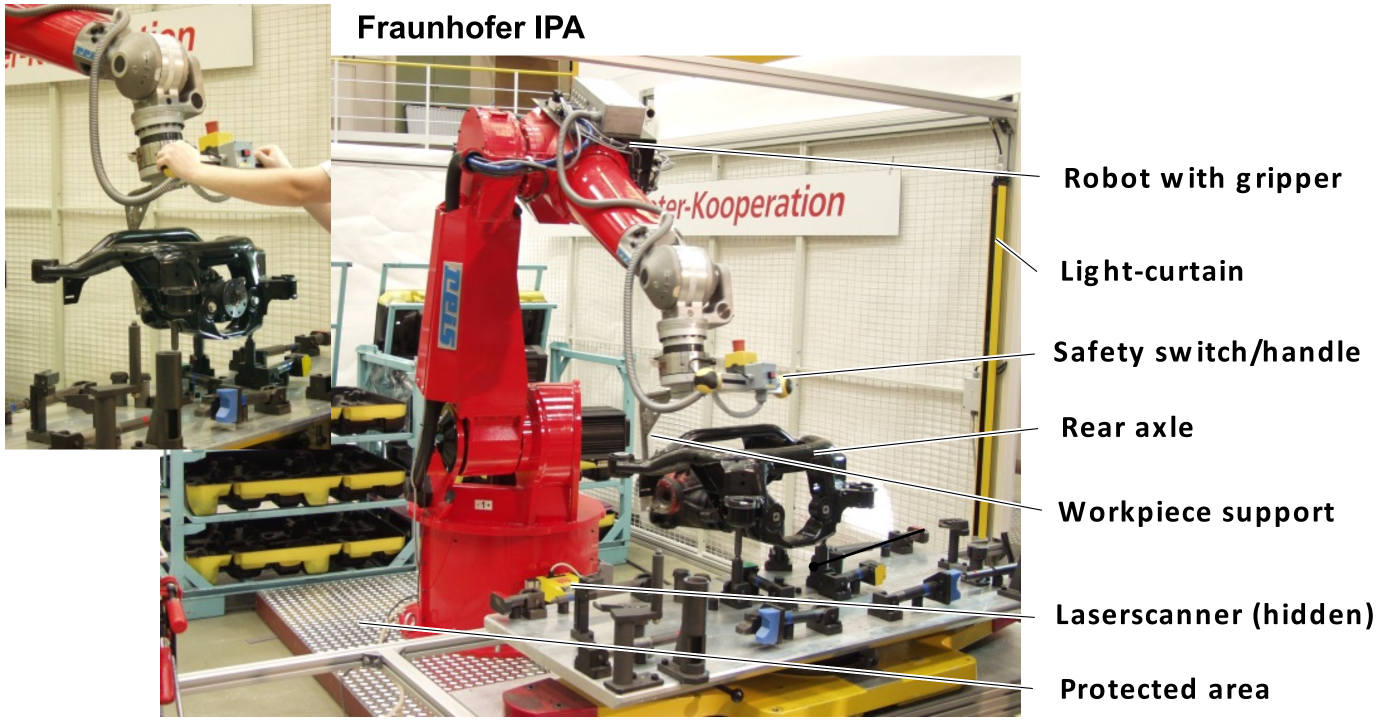
minimaler Punktabstand: 1

konstanter Punktabstand: 10

non-traditional: HMI

Inside a regular *workcell* which is secured **by light curtains, the robot handles gear boxes at regular speed in fully automated mode. Upon approaching the light curtain at reduced speed, the** worker grasps the safety switch which activates both the reduced-speed mode and the force-torque sensor. **The worker guides the robot almost effortlessly by its handle so that the gear-box is balanced with precision into the rear axle frame for final assembly**.

Fraunhofer IPA



Pictures: Fraunhofer IPA

projects that supported our developments

The work reported in this presentation was developed under the support of the following projects and initiatives:

European Projects

- SMErobot^{TM}: 6th Framework Research Program (www.smerobot.org)
- ECHORD: 7th Framework Research Program (www.echord.info)

FCT projects

- Force Control in Industrial Robotics
- HLP for Industrial Robotics Cells: capturing human body motions

ADI & QREN projects

- Robotic Deburring for the Cutlery Industry: IVO Cutlery + FCTUC

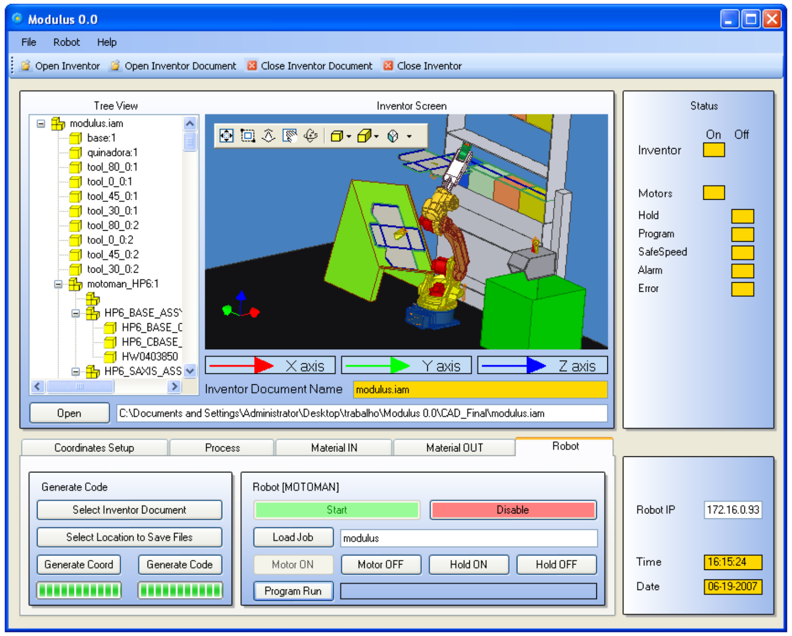
Industrial Projects

- Several in cooperation with equipment manufacturers (robots, sensors, etc.) and end-users

Internal Initiatives

- Robotic solutions for dental implantology
- Force control software
- Interface solutions related with several devices and sensors
- Speech interfaces

Programming - CAD



modulus.iam
base:1
quinadora:1
tool_80_0:1
tool_0_0:1
tool_45_0:1
tool_30_0:1
tool_80_0:2
tool_0_0:2
tool_45_0:2
tool_30_0:2
motoman_HP6:1
HP6_BASE_ASS
HP6_BASE_C
HP6_CBASE_
HW0403850
HP6_SAXIS_ASS

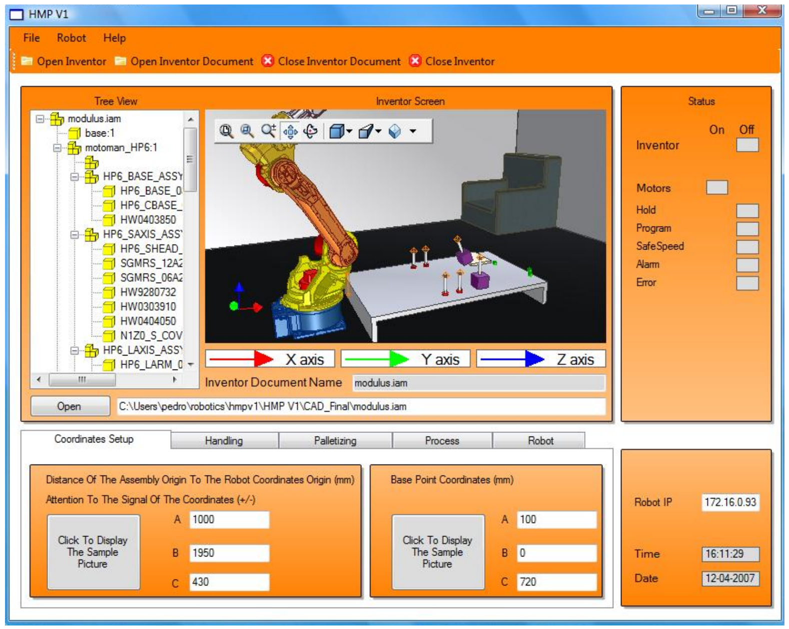
Inventor Document Name: modulus.iam

C:\Documents and Settings\Administrator\Desktop\trabalho\Modulus 0.0\CAD_Final\modulus.iam

Status	On	Off
Inventor	🟨	
Motors	🟨	
Hold		🟨
Program		🟨
SafeSpeed		🟨
Alarm		🟨
Error		🟨

Load Job: modulus

Robot IP	172.16.0.93
Time	16:15:24
Date	06-19-2007



modulus.iam
base:1
motoman_HP6:1
HP6_BASE_ASSY
HP6_BASE_0
HP6_CBASE_
HW0403850
HP6_SAXIS_ASS
HP6_SHEAD_
SGMRS_12A2
SGMRS_06A2
HW9280732
HW0303910
HW0404050
N1Z0_S_COV
HP6_LAXIS_ASS
HP6_LARM_0

Inventor Document Name: modulus.iam

C:\Users\pedro\robotics\hmpv1\HMP V1\CAD_Final\modulus.iam

Status	On	Off
Inventor		⬜
Motors	⬜	
Hold		⬜
Program		⬜
SafeSpeed		⬜
Alarm		⬜
Error		⬜

Distance Of The Assembly Origin To The Robot Coordinates Origin (mm)
Attention To The Signal Of The Coordinates (+/-)

A	1000
B	1950
C	430

Base Point Coordinates (mm)

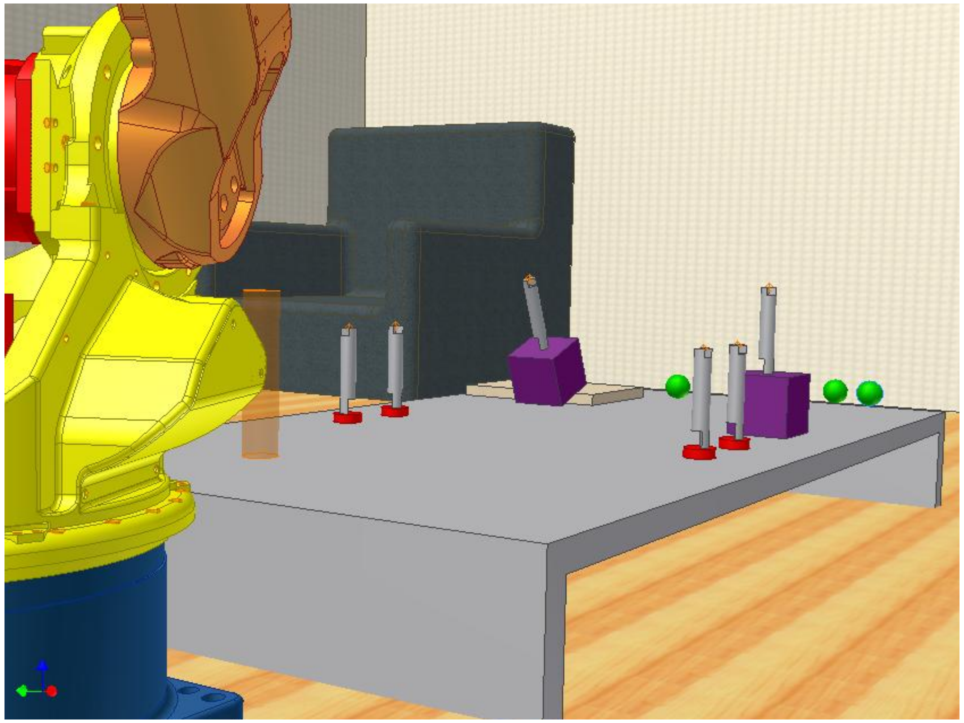
A	100
B	0
C	720

Robot IP	172.16.0.93
Time	16:11:29
Date	12-04-2007

A CAD application could be the environment used for specifying how the robot should perform the required operations on the specified parts. This is not quite task-level programming since human operators do the overall planning.

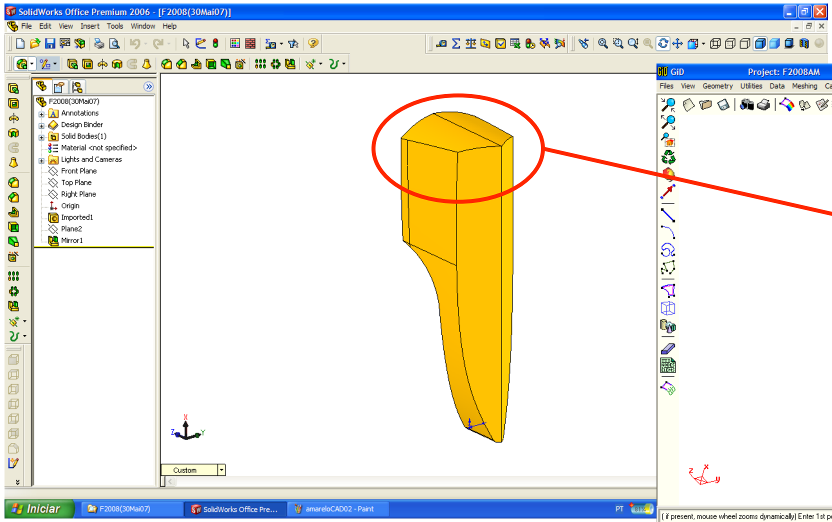
CAD software packages are **powerful 3D tools** and are now **very common** among manufacturing companies.

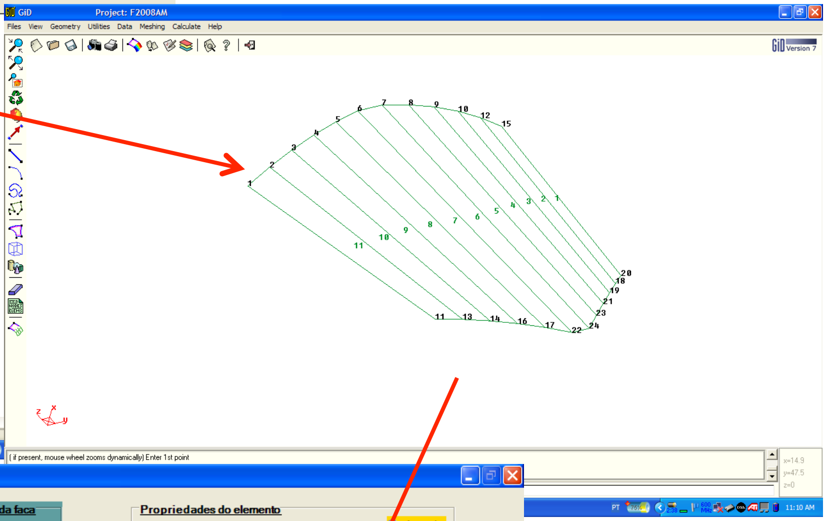
Consequently, **using those tools for robot programming is desirable** since the operator may start the off-line programming of the necessary manufacturing operations using the 3D model of the product.



programming – CAD & code generation

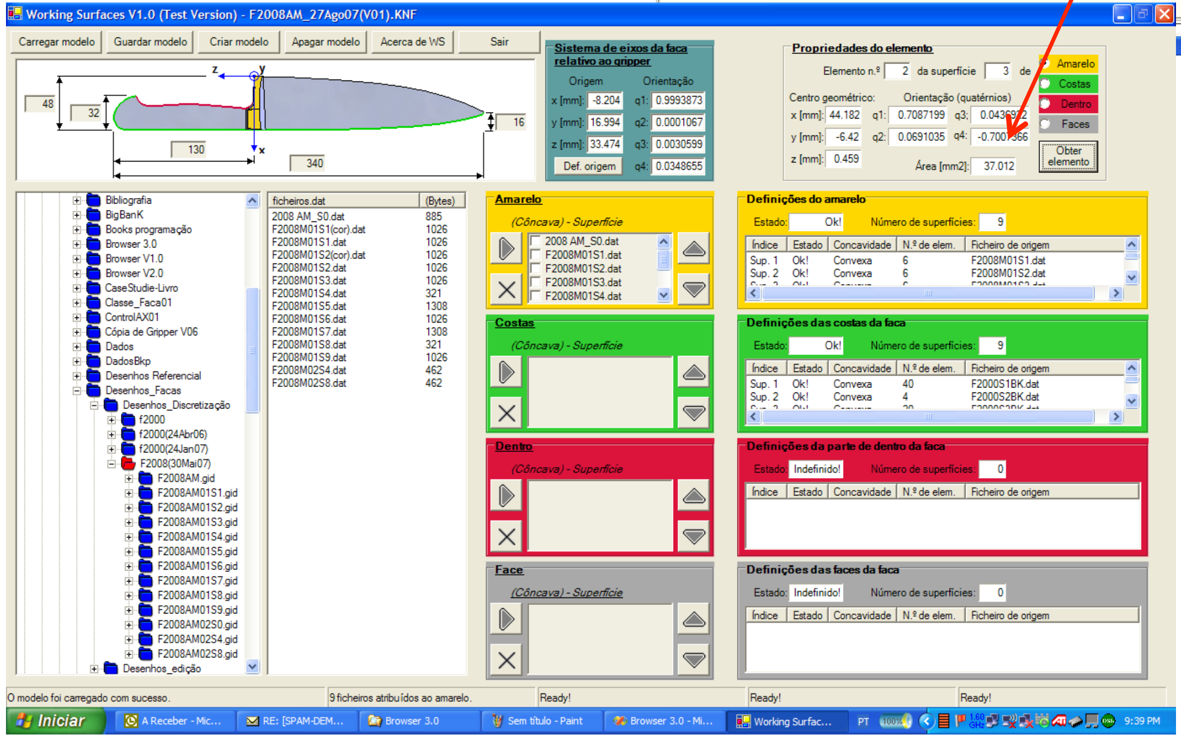
Demo – Prototype - Tool





(if present, mouse wheel zooms dynamically) Enter 1st point

IVO
IVO CUTLERY INTERNATIONAL INC.
None better for price and quality



Sistema de eixos da faca relativo ao gripper

	Origem		Orientação
x [mm]:	-8.204	q1:	0.9993873
y [mm]:	16.994	q2:	0.0001067
z [mm]:	33.474	q3:	0.0030599
		q4:	0.0348655

Propriedades do elemento

Elemento n.º 2 da superfície 3 de

Centro geométrico:		Orientação (quaterniões)			
x [mm]:	44.182	q1:	0.7087199	q3:	0.0436922
y [mm]:	-6.42	q2:	0.0691035	q4:	-0.7001366
z [mm]:	0.459				

Área [mm2]: 37.012

Bibliografia
BigBanK
Books programação
Browser 3.0
Browser V1.0
Browser V2.0
CaseStudie-Livro
Classe_Faca01
ControlAX01
Cópia de Gripper V06
Dados
DadosBkp
Desenhos Referencial
Desenhos_Facas
Desenhos_Discretização
f2000
f2000(24Abr06)
f2000(24Jan07)
F2008(30Mai07)
F2008AM.gid
F2008AM01S1.gid
F2008AM01S2.gid
F2008AM01S3.gid
F2008AM01S4.gid
F2008AM01S5.gid
F2008AM01S6.gid
F2008AM01S7.gid
F2008AM01S8.gid
F2008AM01S9.gid
F2008AM02S0.gid
F2008AM02S4.gid
F2008AM02S8.gid
Desenhos_edição

ficheiros.dat	(Bytes)
2008 AM_S0.dat	885
F2008M01S1(cor).dat	1026
F2008M01S1.dat	1026
F2008M01S2(cor).dat	1026
F2008M01S2.dat	1026
F2008M01S3.dat	1026
F2008M01S4.dat	321
F2008M01S5.dat	1308
F2008M01S6.dat	1026
F2008M01S7.dat	1308
F2008M01S8.dat	321
F2008M01S9.dat	1026
F2008M02S4.dat	462
F2008M02S8.dat	462

Amarelo

(Côncava) - Superfície

2008 AM_S0.dat
F2008M01S1.dat
F2008M01S2.dat
F2008M01S3.dat
F2008M01S4.dat

Definições do amarelo

Estado: Ok! Número de superfícies: 9

Índice	Estado	Concavidade	N.º de elem.	Ficheiro de origem
Sup. 1	Ok!	Convexa	6	F2008M01S1.dat
Sup. 2	Ok!	Convexa	6	F2008M01S2.dat

Costas

(Côncava) - Superfície

Definições das costas da faca

Estado: Ok! Número de superfícies: 9

Índice	Estado	Concavidade	N.º de elem.	Ficheiro de origem
Sup. 1	Ok!	Convexa	40	F2000S1BK.dat
Sup. 2	Ok!	Convexa	4	F2000S2BK.dat

Dentro

(Côncava) - Superfície

Definições da parte de dentro da faca

Estado: Indefinido! Número de superfícies: 0

Índice	Estado	Concavidade	N.º de elem.	Ficheiro de origem

Face

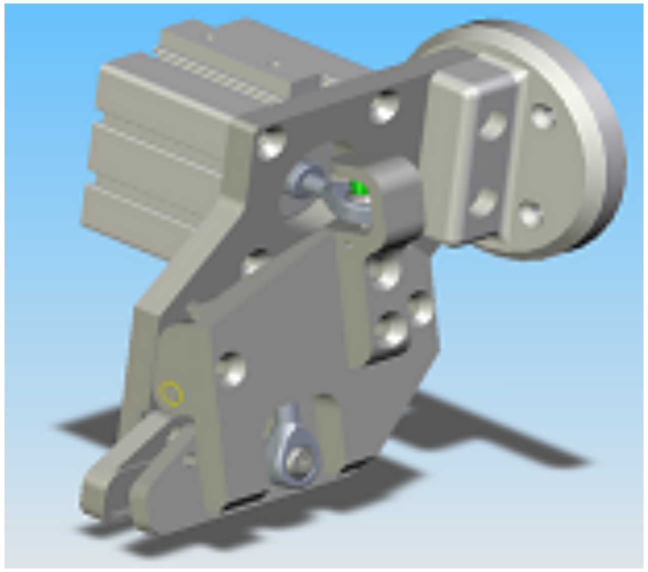
(Côncava) - Superfície

Definições das faces da faca

Estado: Indefinido! Número de superfícies: 0

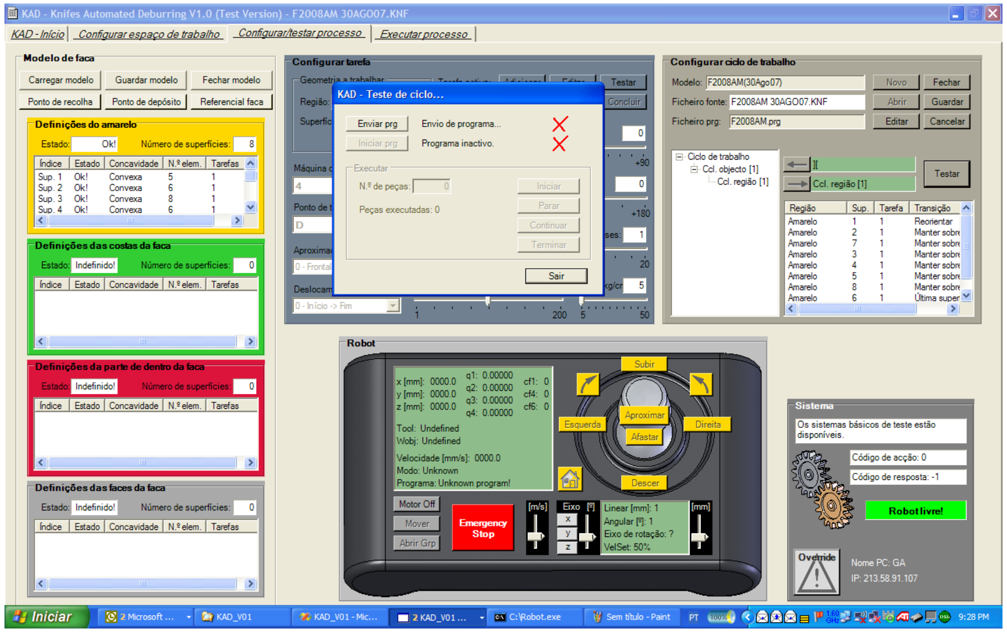
Índice	Estado	Concavidade	N.º de elem.	Ficheiro de origem

O modelo foi carregado com sucesso. | 9 ficheiros atribuídos ao amarelo. | Ready! | Ready! | Ready!



© j. norberto pires 2012

programming – CAD & code generation


KAD - Knifes Automated Deburring V1.0 (Test Version) - F2008AM 30AGO07.KNF

Definições do amarelo — Estado: Ok! — Número de superfícies: 8

Índice	Estado	Concavidade	N.º elem.	Tarefas
Sup. 1	Ok!	Convexa	5	1
Sup. 2	Ok!	Convexa	6	1
Sup. 3	Ok!	Convexa	8	1
Sup. 4	Ok!	Convexa	6	1

Definições das costas da faca — Estado: Indefinido! — Número de superfícies: 0

Definições da parte de dentro da faca — Estado: Indefinido! — Número de superfícies: 0

Definições das faces da faca — Estado: Indefinido! — Número de superfícies: 0

KAD - Teste de ciclo...
Enviar prg — Envio de programa...
Iniciar prg — Programa inactivo.
Executar — N.º de peças: 0 — Peças executadas: 0

Configurar ciclo de trabalho
Modelo: F2008AM(30Ago07)
Ficheiro fonte: F2008AM 30AGO07.KNF
Ficheiro prg: F2008AM.prg

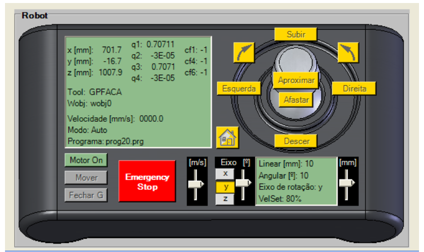
Região	Sup.	Tarefa	Transição
Amarelo	1	1	Reorientar
Amarelo	2	1	Manter sobre
Amarelo	7	1	Manter sobre
Amarelo	3	1	Manter sobre
Amarelo	4	1	Manter sobre
Amarelo	5	1	Manter sobre
Amarelo	8	1	Manter sobre
Amarelo	6	1	Última super

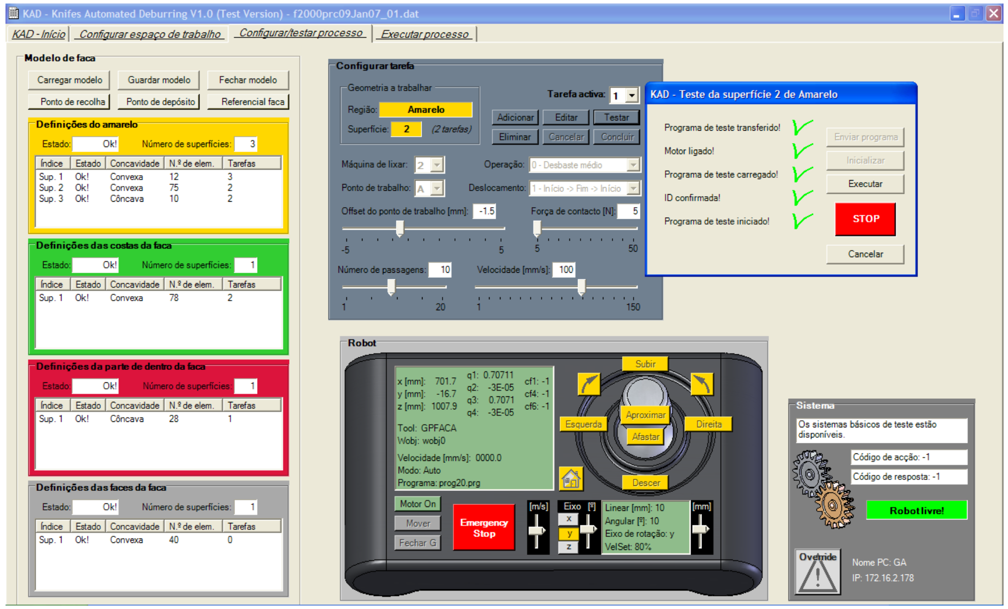
Robot
x [mm]: 0000.0 y [mm]: 0000.0 z [mm]: 0000.0
q1: 0.00000 q2: 0.00000 q3: 0.00000 q4: 0.00000
cf1: 0 cf4: 0 cf6: 0
Tool: Undefined
Wobj: Undefined
Velocidade [mm/s]: 0000.0
Modo: Unknown
Programa: Unknown program!
Linear [mm]: 1
Angular [º]: 1
Eixo de rotação: ?
VelSet: 50%

Sistema
Os sistemas básicos de teste estão disponíveis.
Código de acção: 0
Código de resposta: -1
Robot livre!
Nome PC: GA
IP: 213.58.91.107

Features:

- CAD Model: *Solidworks* and *Inventor*
- Surface modeling defining where to debur and grind
- Import surfaces and code generation
- Code adjusted using a graphical HMI
- Using a FT sensor to adjust the code
- A new model takes 5 to 10 minutes to program.
- Demonstrated using an industrial prototype.


Robot
x [mm]: 701.7 y [mm]: -16.7 z [mm]: 1007.9
q1: 0.70711 q2: -3E-05 q3: 0.7071 q4: -3E-05
cf1: -1 cf4: -1 cf6: -1
Tool: GPFACA
Wobj: wobj0
Velocidade [mm/s]: 0000.0
Modo: Auto
Programa: prog20.prg
Linear [mm]: 10
Angular [º]: 10
Eixo de rotação: y
VelSet: 80%


KAD - Knifes Automated Deburring V1.0 (Test Version) - f2000prc09Jan07_01.dat

Definições do amarelo — Estado: Ok! — Número de superfícies: 3

Índice	Estado	Concavidade	N.º de elem.	Tarefas
Sup. 1	Ok!	Convexa	12	3
Sup. 2	Ok!	Convexa	75	2
Sup. 3	Ok!	Côncava	10	2

Definições das costas da faca — Estado: Ok! — Número de superfícies: 1

Índice	Estado	Concavidade	N.º de elem.	Tarefas
Sup. 1	Ok!	Convexa	78	2

Definições da parte de dentro da faca — Estado: Ok! — Número de superfícies: 1

Índice	Estado	Concavidade	N.º de elem.	Tarefas
Sup. 1	Ok!	Côncava	28	1

Definições das faces da faca — Estado: Ok! — Número de superfícies: 1

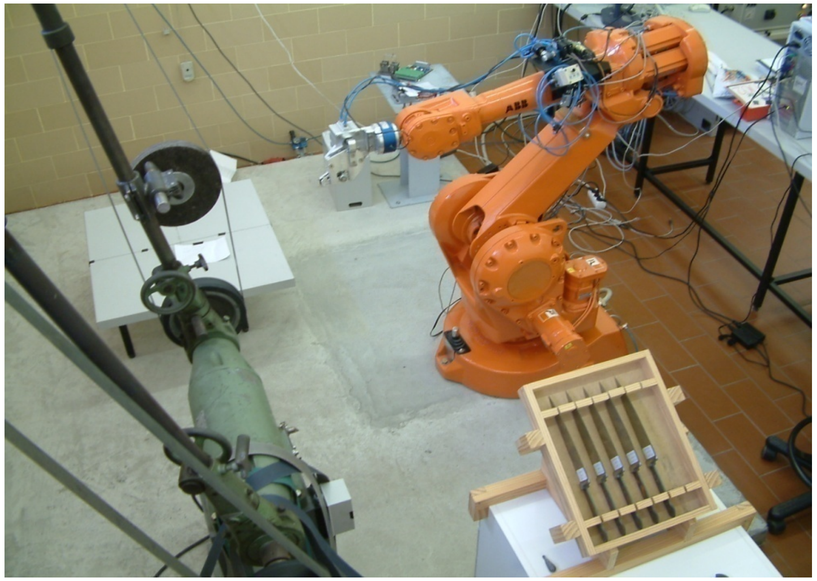
Índice	Estado	Concavidade	N.º de elem.	Tarefas
Sup. 1	Ok!	Convexa	40	0

Configurar tarefa
Geometria a trabalhar — Tarefa activa: 1
Região: Amarelo
Superfície: 2 (2 tarefas)
Máquina de lixar: 2
Operação: 0 - Desbaste médio
Ponto de trabalho: A
Deslocamento: 1 - Início -> Fim -> Início
Offset do ponto de trabalho [mm]: -1.5
Força de contacto [N]: 5
Número de passagens: 10
Velocidade [mm/s]: 100

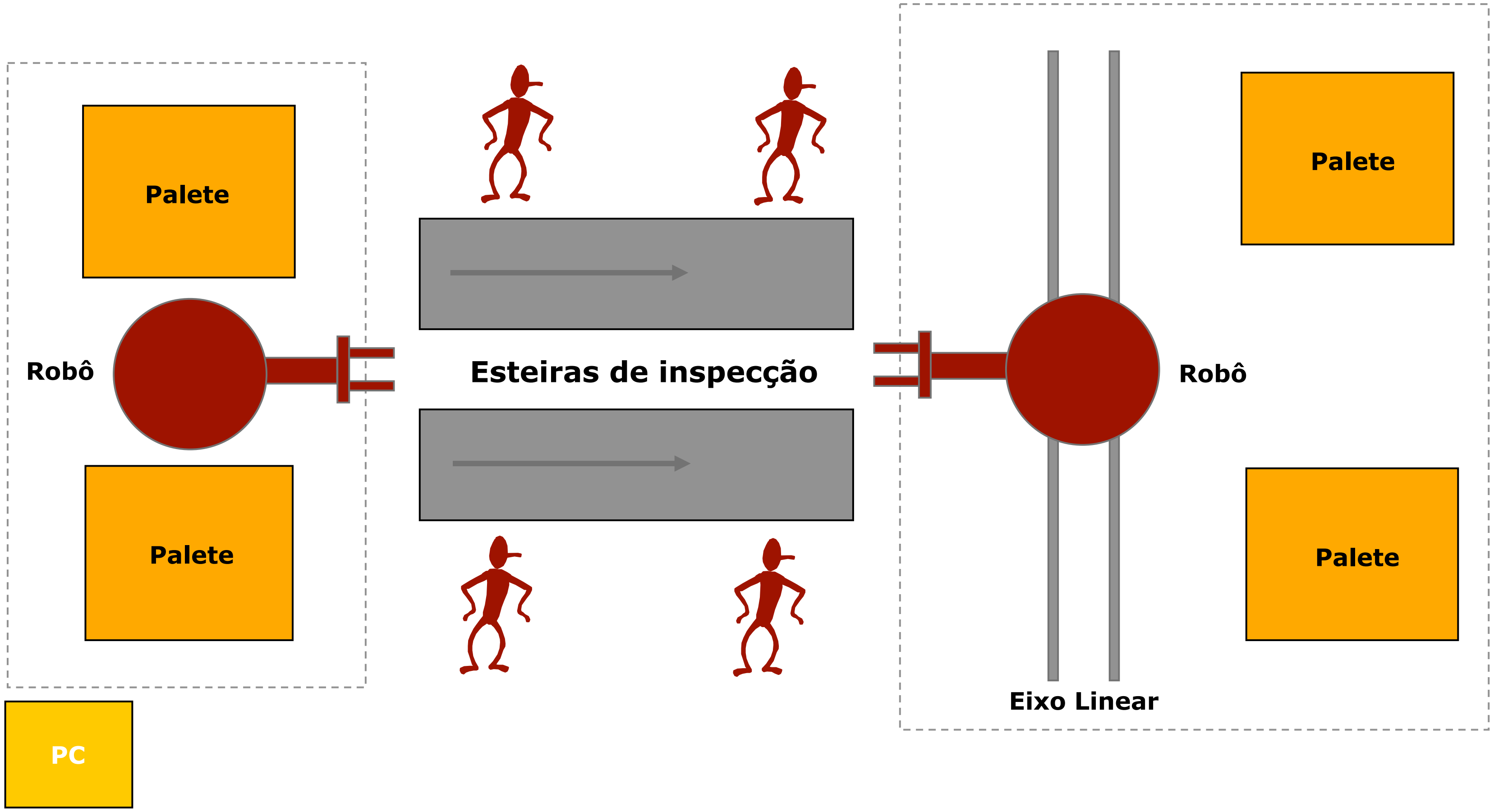
KAD - Teste da superfície 2 de Amarelo
Programa de teste transferido! ✓
Motor ligado! ✓
Programa de teste carregado! ✓
ID confirmada! ✓
Programa de teste iniciado! ✓

Robot
x [mm]: 701.7 y [mm]: -16.7 z [mm]: 1007.9
q1: 0.70711 q2: -3E-05 q3: 0.7071 q4: -3E-05
cf1: -1 cf4: -1 cf6: -1
Tool: GPFACA
Wobj: wobj0
Velocidade [mm/s]: 0000.0
Modo: Auto
Programa: prog20.prg
Linear [mm]: 10
Angular [º]: 10
Eixo de rotação: y
VelSet: 80%

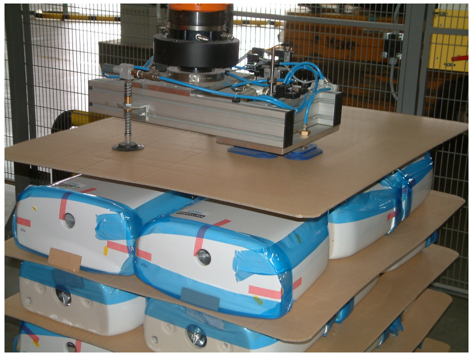
Sistema
Os sistemas básicos de teste estão disponíveis.
Código de acção: -1
Código de resposta: -1
Robot livre!
Nome PC: GA
IP: 172.16.2.178



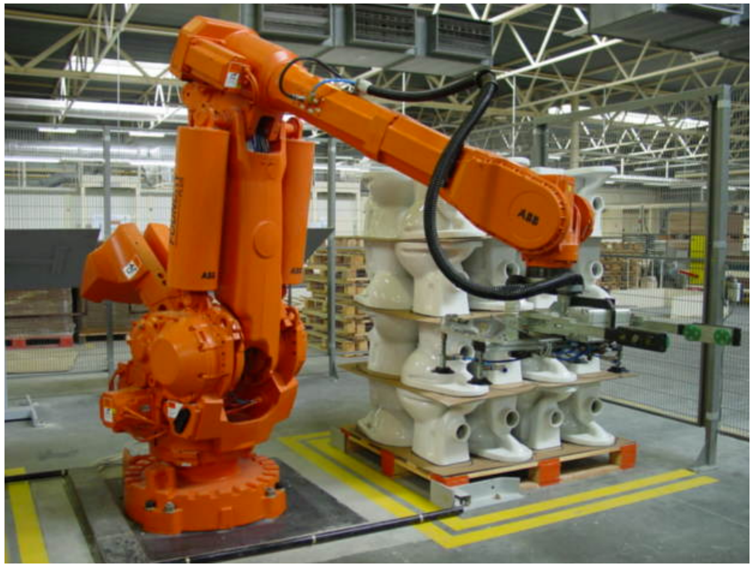
programming – PC interfaces & databases & teaching

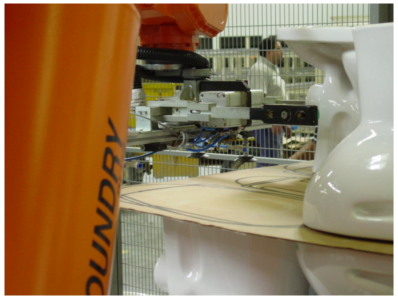


programming – PC interfaces & databases & teaching



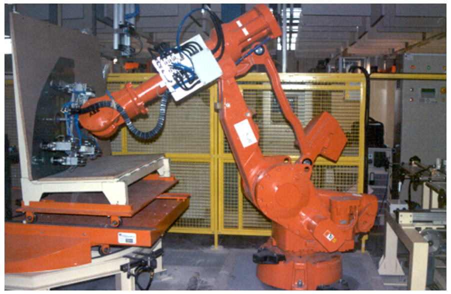
programming – PC interfaces & databases & teaching

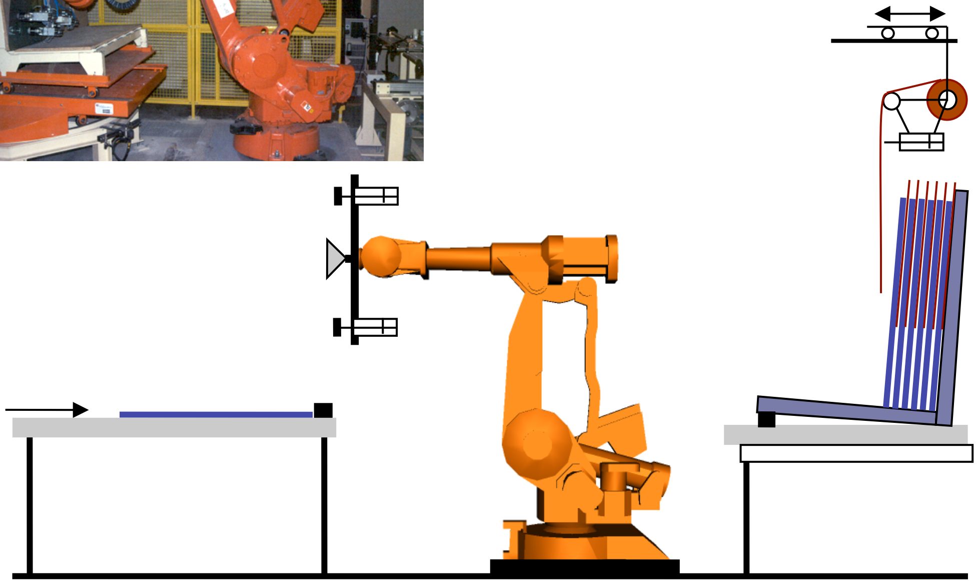


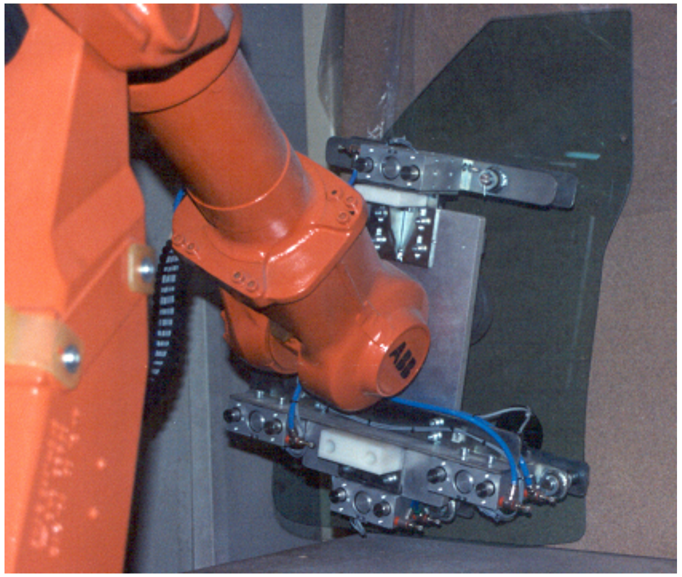


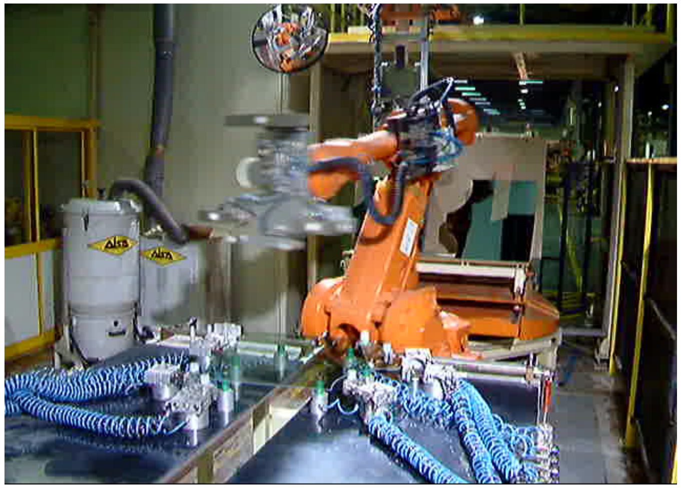
© j. norberto pires 2012

programming – PC interfaces & databases & teaching



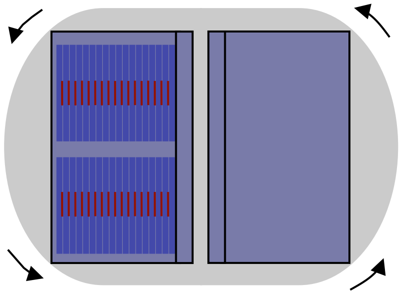




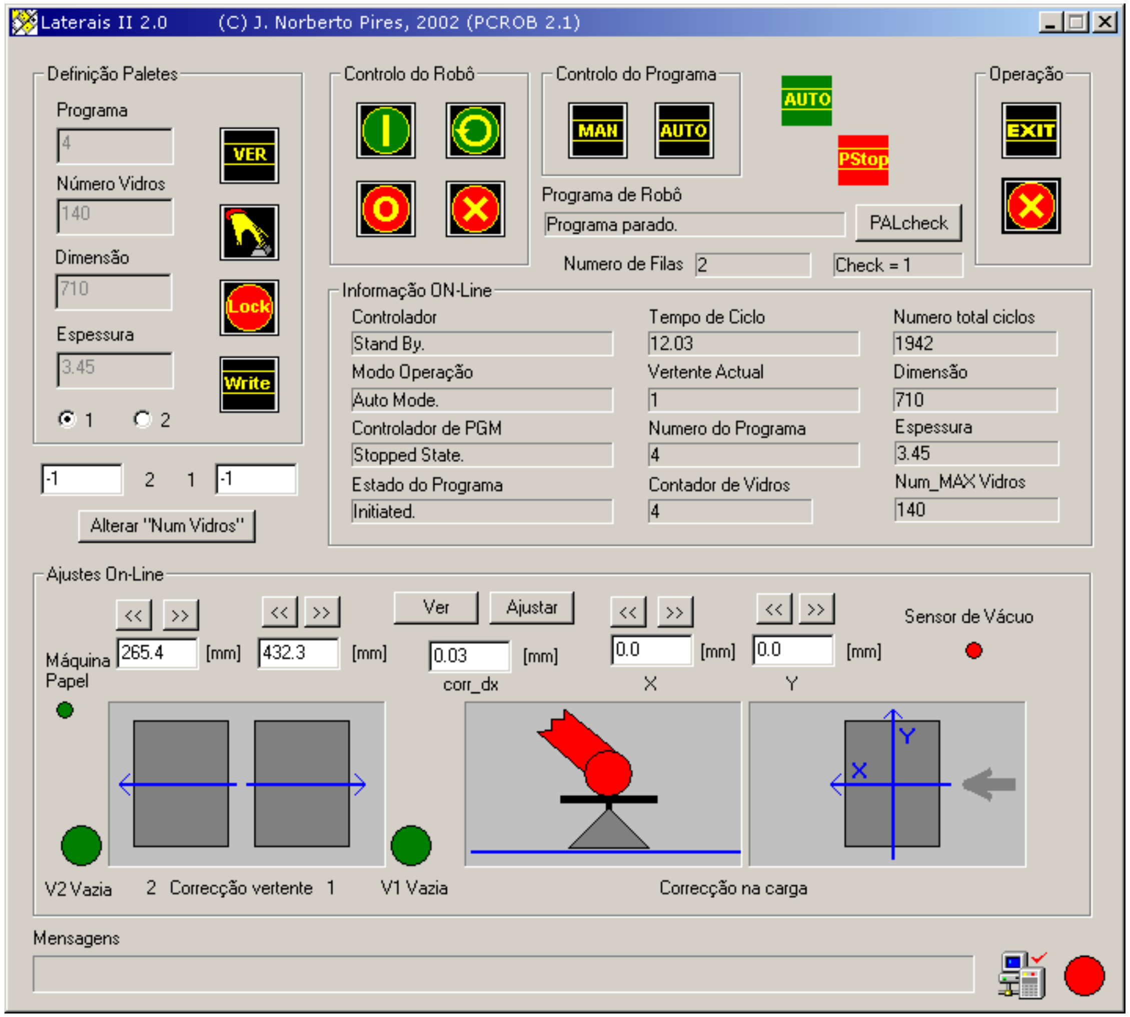


Basic requirements here:

- programming: adding new models
- human-machine interfaces
- on-line adjustment
- on-line monitoring
- integration
- multi-model production



programming – PC interfaces & databases & teaching



Laterais II 2.0 (C) J. Norberto Pires, 2002 (PCROB 2.1)

Definição Paletes

Programa	4
Número Vidros	140
Dimensão	710
Espessura	3.45

[✓ 1] 2

-1 2 1 -1

Controlo do Programa

Programa de Robô: Programa parado.

Numero de Filas: 2

Check = 1

Informação ON-Line

Controlador	Tempo de Ciclo	Numero total ciclos
Stand By.	12.03	1942
Modo Operação	Vertente Actual	Dimensão
Auto Mode.	1	710
Controlador de PGM	Numero do Programa	Espessura
Stopped State.	4	3.45
Estado do Programa	Contador de Vidros	Num_MAX Vidros
Initiated.	4	140

Ajustes On-Line

Máquina Papel: 265.4 [mm] 432.3 [mm]

corr_dx: 0.03 [mm]

X: 0.0 [mm]

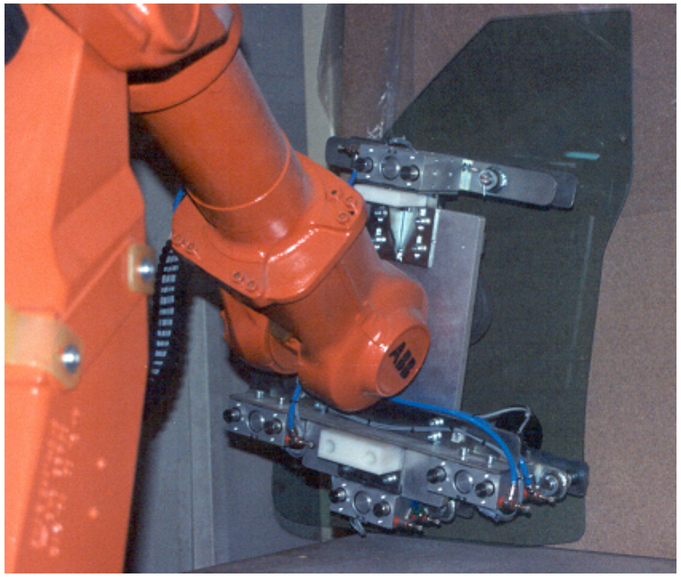
Y: 0.0 [mm]

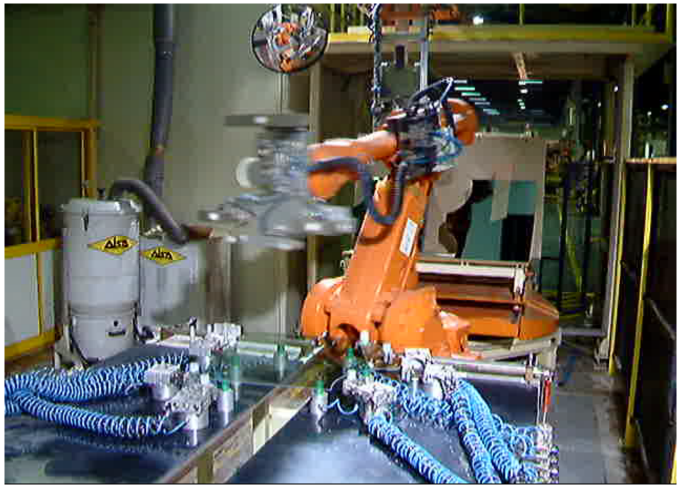
Sensor de Vácuo

V2 Vazia — 2 Correcção vertente 1 — V1 Vazia

Correcção na carga

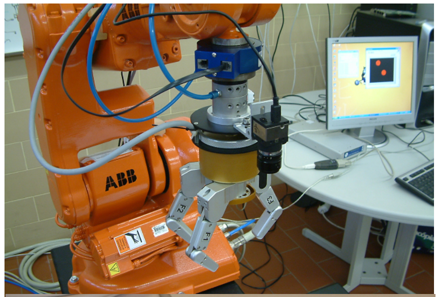
Mensagens

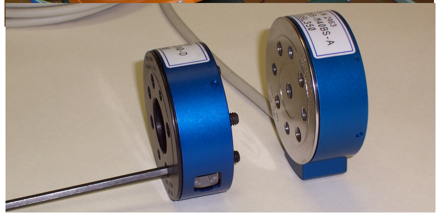
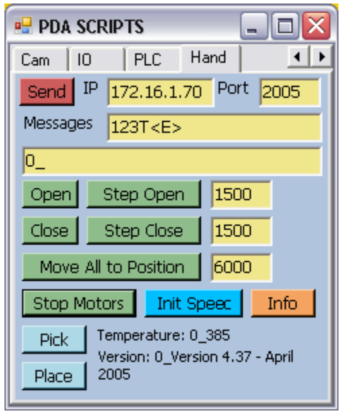




> Go to slide with IT infrastructure

programming – sensors & IO devices



IP 172.16.1.70 Port 2005

Messages 123T<E>

0_

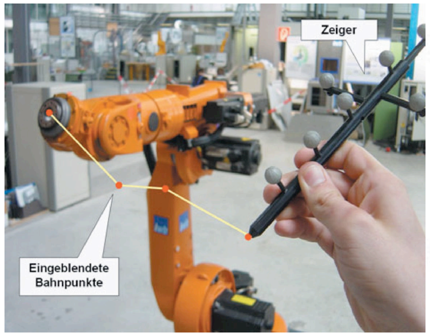
1500

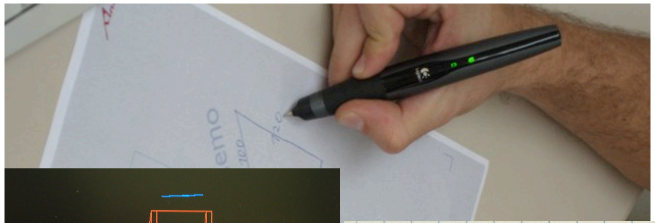
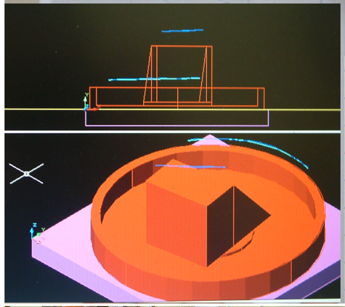
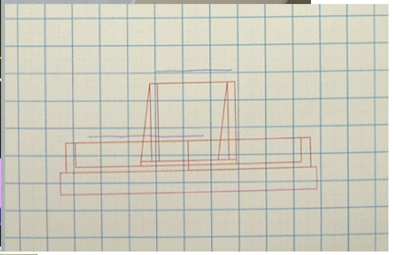
1500

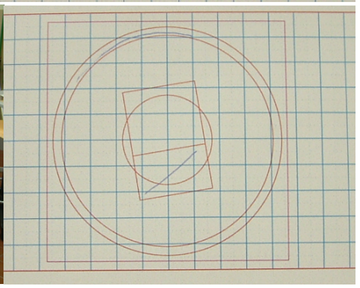
6000

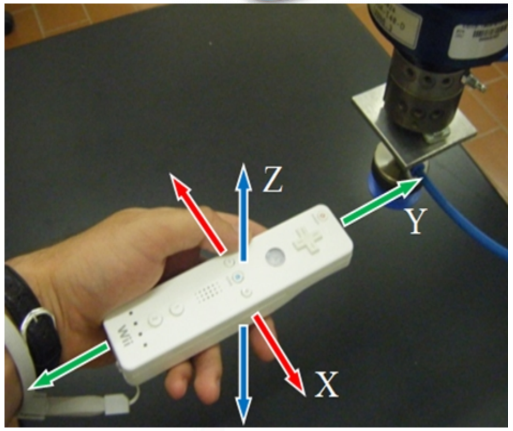
Temperature: 0_385

Version: 0_Version 4.37 - April 2005



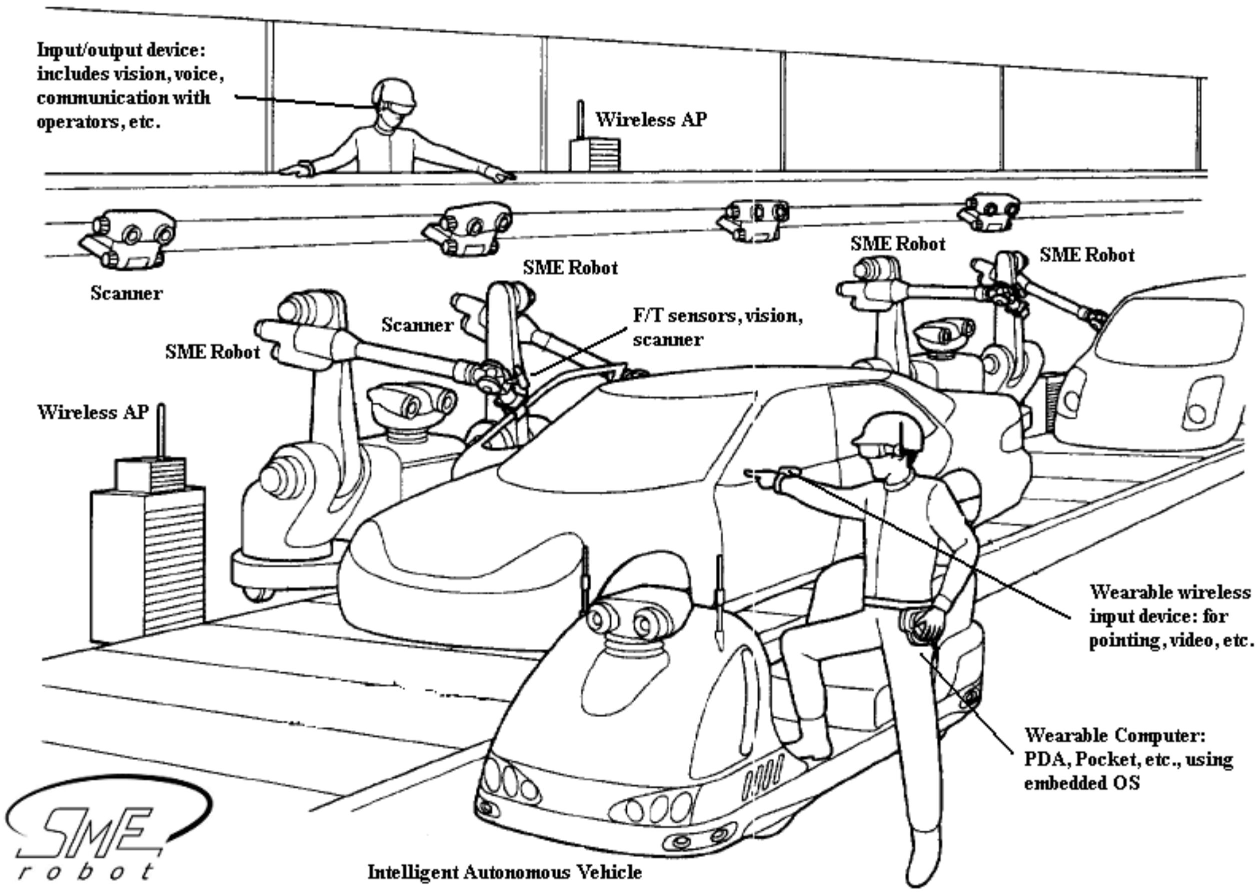








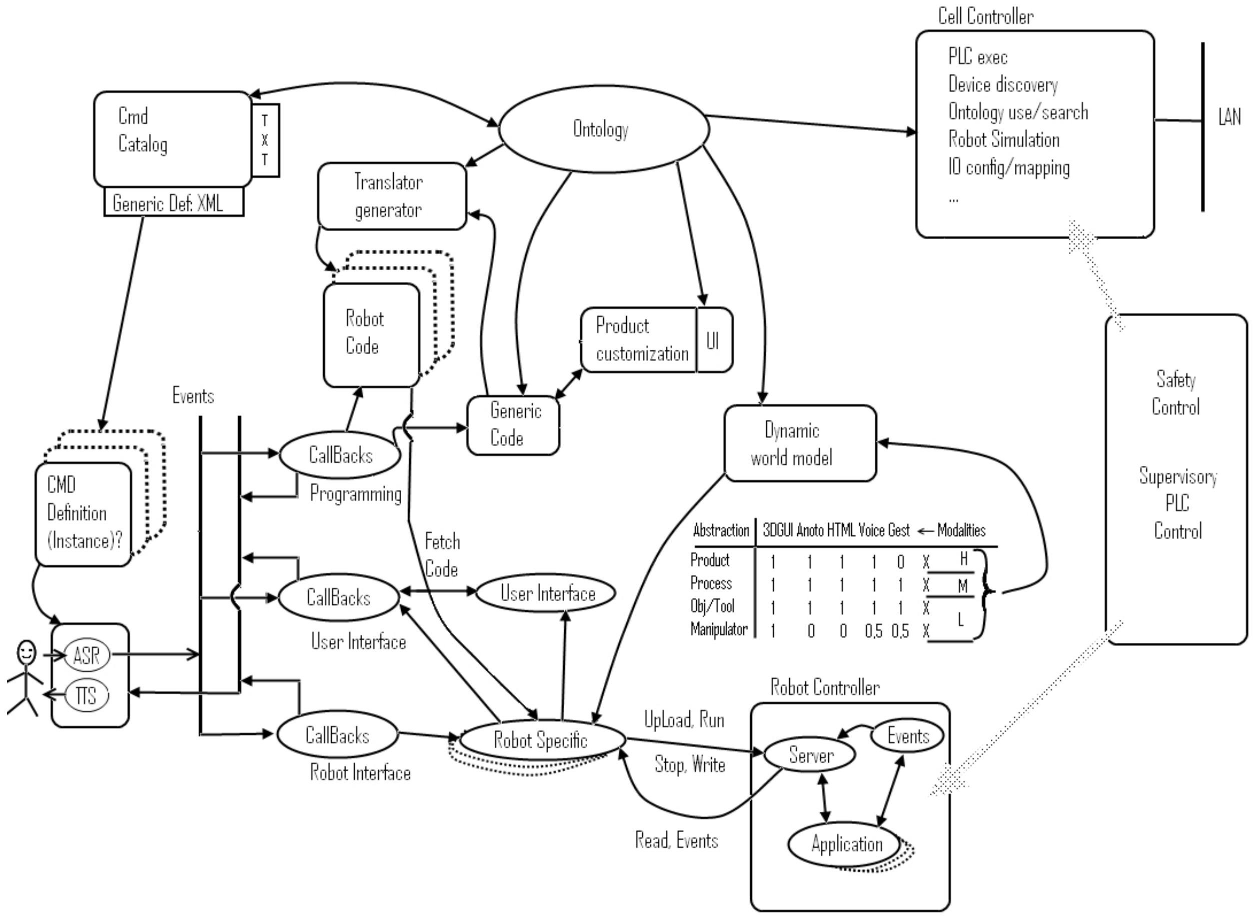
SMErobot HLP framework

The ***high-level programming*** *(HLP) concept applied to industrial robots* ***<u>means</u> programming by any type of user*** *(not restricted to specialists),* ***independence of the platform and higher levels of abstraction****.*


Input/output device: includes vision, voice, communication with operators, etc.
Wireless AP
Scanner
SME Robot
Scanner
SME Robot
F/T sensors, vision, scanner
SME Robot
SME Robot
Wireless AP
Wearable wireless input device: for pointing, video, etc.
Wearable Computer: PDA, Pocket, etc., using embedded OS
Intelligent Autonomous Vehicle
SME robot

HLP can be understood as a set of techniques *(<u>offered probably has reusable software components</u>) used to program SME (small and medium enterprises) based manufacturing cells,* ***which hide from the programmer the tricky details about how to obtain the machine code necessary to implement agile manufacturing****.*

SMErobot HLP framework



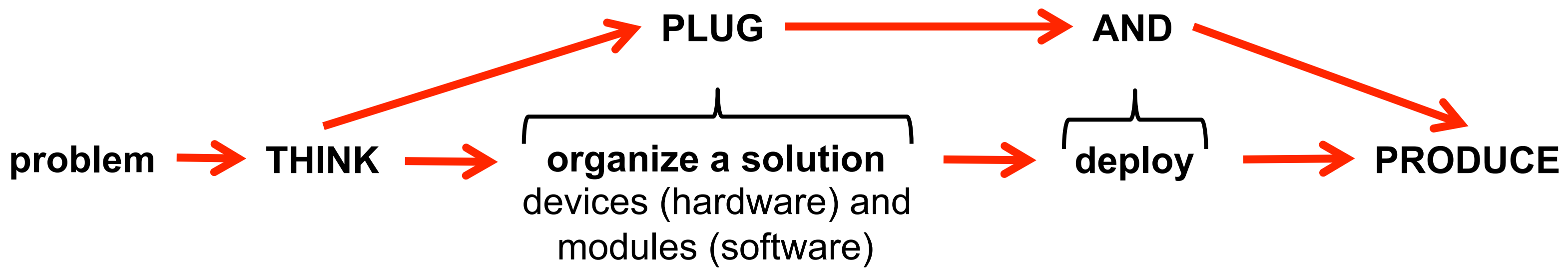
Abstraction	3DGUI	Anoto	HTML	Voice	Gest	← Modalities	
Product	1	1	1	1	0	X	H
Process	1	1	1	1	1	X	M
Obj/Tool	1	1	1	1	1	X	
Manipulator	1	0	0	0,5	0,5	X	L

*The **high-level programming** (HLP) concept applied to industrial robots __**means**__ **programming by any type of user** (not restricted to specialists), **independence of the platform and higher levels of abstraction**.*

HLP can be understood as a set of techniques *(<u>offered probably has reusable software components</u>) used to program SME (small and medium enterprises) based manufacturing cells,* ***which hide from the programmer the tricky details about how to obtain the machine code necessary to implement agile manufacturing****.*

HLP - discussion

*The **high-level programming (HLP)** concept applied to industrial robots means programming by any type of user (not restricted to specialists), independence of the platform and higher levels of abstraction. In fact, **HLP** can be understood **as a set of techniques (offered probably has reusable software components)** used to program SME (small and medium enterprises) based manufacturing cells, **which hide from the programmer the tricky details about how to obtain the machine code necessary to implement agile manufacturing.***



HLP - discussion

- **Extend the programming task** to inexperienced users, which don’t want to have special and detailed robotics knowledge to use a robot system, and don’t want to get into the lower programming levels to be able use a robot;

- **Command and configure robot systems,** instead of programming and setting them up on a lower level. Actions shall be defined on a high abstraction level as e.g. “Move object A from B to C” without defining all intermediate positions and including automatic collision avoidance strategies and automatic setting of movement parameters (e.g. speed, acceleration or interpolation mode);

- **Make programming an intuitive procedure** capable of teaching and commanding a robot system. This includes the use of graphical input/representation of user commands and program structures. The command flow and the data input shall be supported via graphical elements using standards, e.g. a style guide to arrange and position the command elements;

HLP - discussion

- **Integrate robot programming with cell programming**. This means introducing process information into the programming effort, extracting from there the detailed system programming necessary to achieve the required operation;

- **Integrate user interface devices** making the maximum out of them and integrating easily their functionality: plug-and-produce concepts are here fundamental;

- **Facilitate the re-programming task.** The re-programming of industrial robotic systems is still a difficult, costly, and time consuming operation. In order to increase flexibility, a common approach is to consider the work-cell programming at a high level of abstraction, which enables a description of the sequence of actions at a task-level.

HLP - discussion

The questions are:
How will the “programming” be done (the focus is on user-machine interfaces, input programming devices, command catalogs, etc., to avoid direct programming)?

How will the robot code be generated (adoption of a common framework independent of any specific robot programming language)?

How will the process knowledge be integrated (production sketches based on ontologies are a possibility, but also service-oriented architectures (SOA) and programming based on CAD packages)?

How will this work with any robot controller (platform independence)?

still open issues!!!

HLP - discussion

To address this problem several possibilities can be considered:

Improved task-level programming, using a high-level programming language: less technical, easier to use, generic, multi-robot. Such a language would provide a structure, constructs, and functions suited for manufacturing control. This results in several advantages with respect to the speed and ease of software development, its modifiability and maintainability, and its cost. Software quality can also be improved by enabling the developer to focus on the tasks to be performed rather than on the way in which they are performed.

Using a classical client-server approach, where servers are designed to run on robot controllers and implemented to be accessed using standard remote procedure calling or messaging protocols. The client applications are designed to handle the required user interfaces and the cell functionality, making calls to server functions (usually complex routines, or macros).

HLP - discussion

Using XML forms, including grammars, state charts, etc.. XML is a standard that can be used to specify manufacturing actions for the complete work cell without writing code, and without generating binary code.

Using CAD packages to program robotic cells. This solution is an interesting approach since CAD packages are available at SMEs and users are usually prepared to use it. Code generation can be available, but it could also be based on services, or RPC/ XIRP calls.

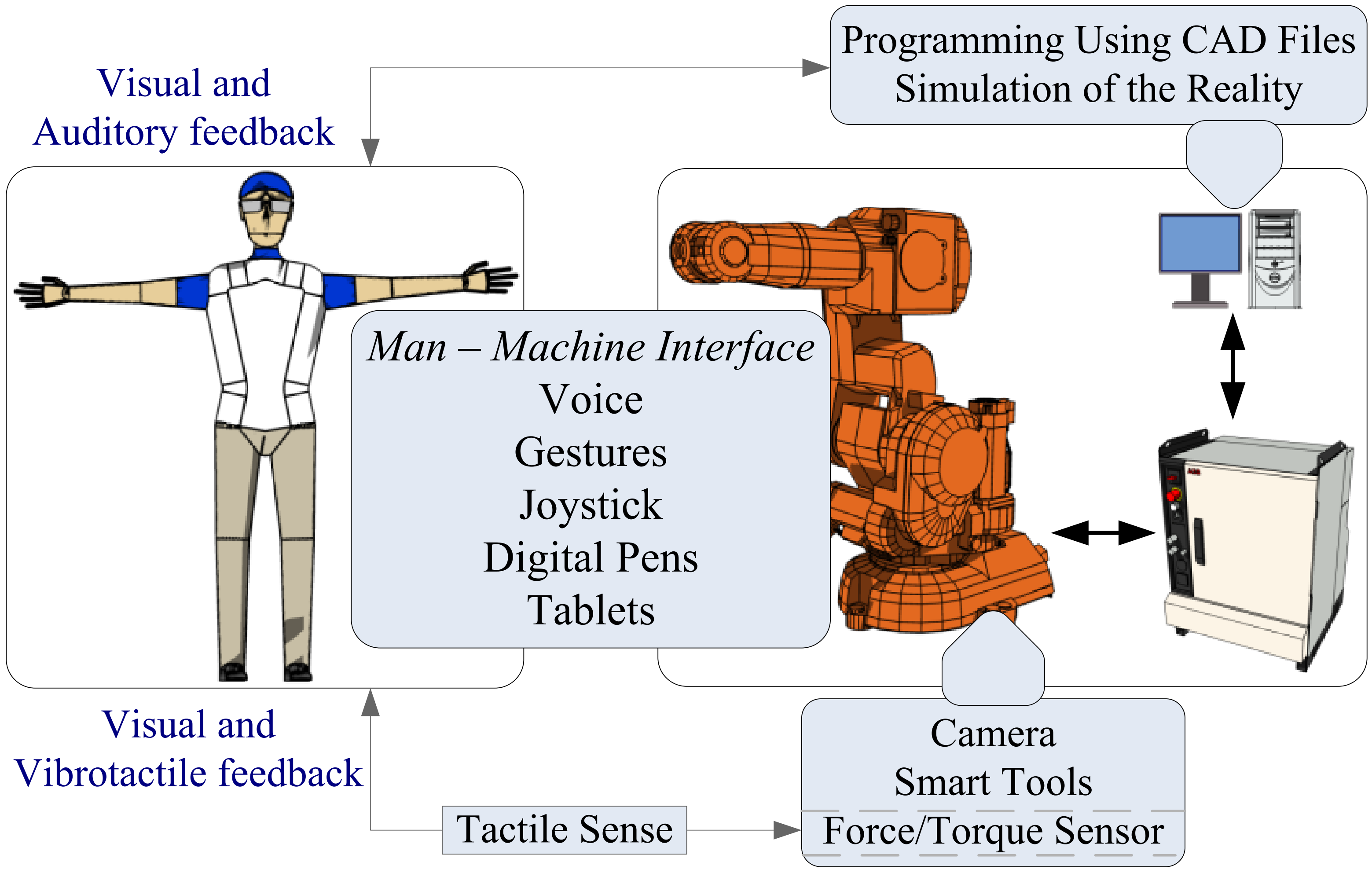
Using Service oriented architectures (SOA). One of the things that a SOA should be is a suitable platform for the development of HLP features, which can occur during orchestration of services when assembling a program.. One of the keys for HLP is environment sensibility which is easily reached with a “hot plug-and-play” discovery as we found in UPnP. In this scenario UPnP -discovery (peer-to-peer) takes advantage.

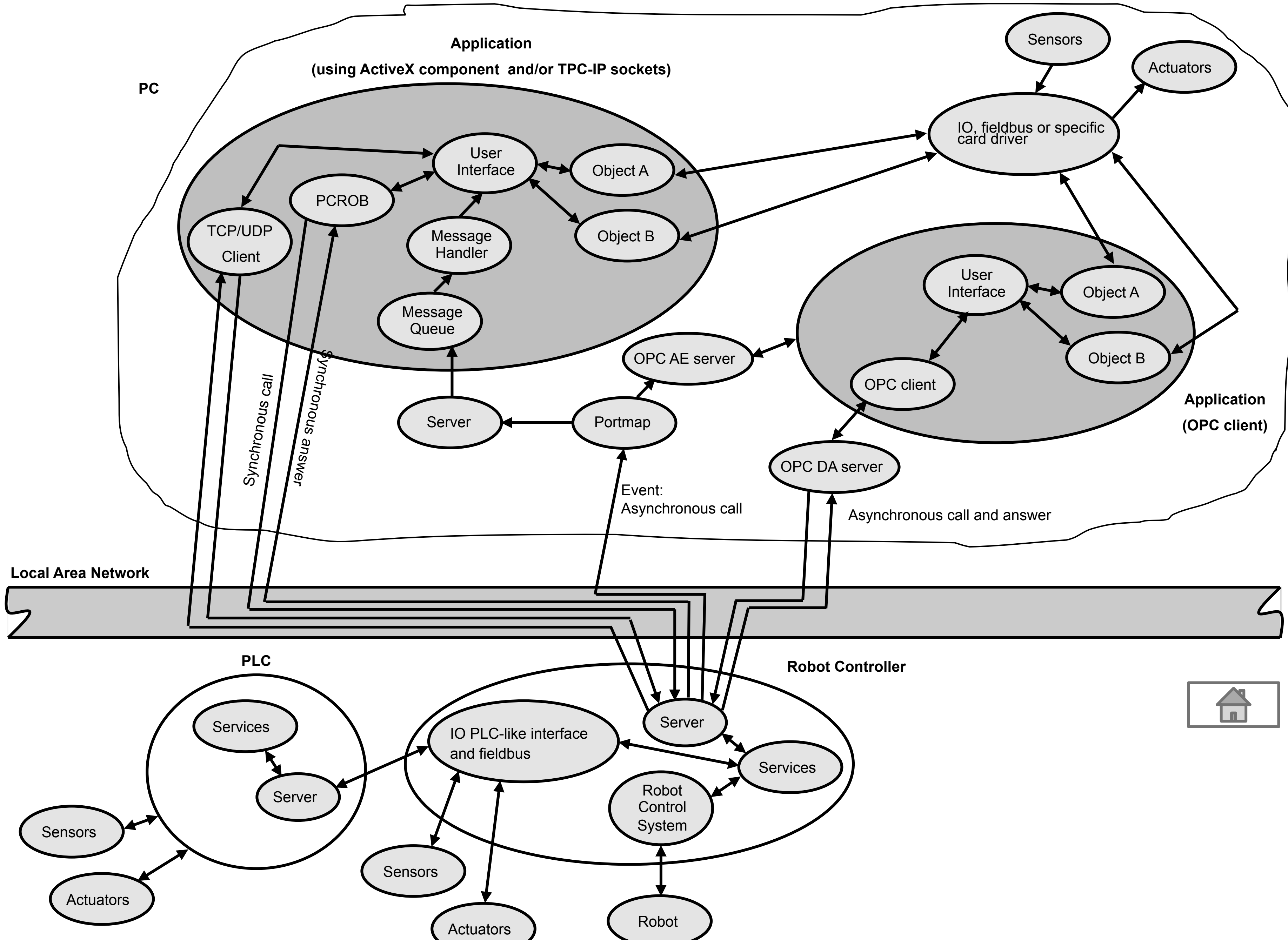
HLP - discussion

Using programming-by-demonstration techniques. This approach includes all the techniques that require the user to program the robot starting by teaching the motions and dealing with the obtained results at a higher level. In fact, while it is possible to record position and orientation data directly in the working scenario, all other interactions shall be defined on a high level: integration of peripherals, increasing the quality of the trajectory and adopting the program to the specific process. These interactions shall take place on a high level of autonomy or interaction with the user.

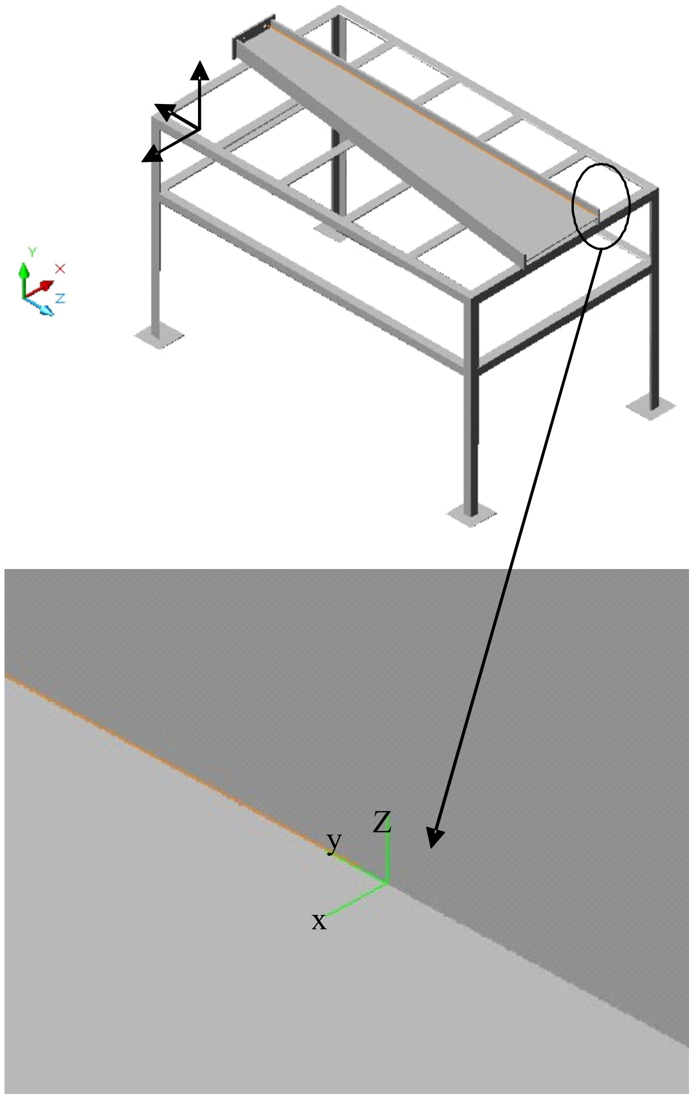
Integrating IO devices and systems: portable devices, speech interfaces, etc. This means the possibility to include these devices into the same high-level framework without going into the tricky details about how to program them, which can be done just by adopting SOA on a plug-and-produce environment.

HLP - discussion

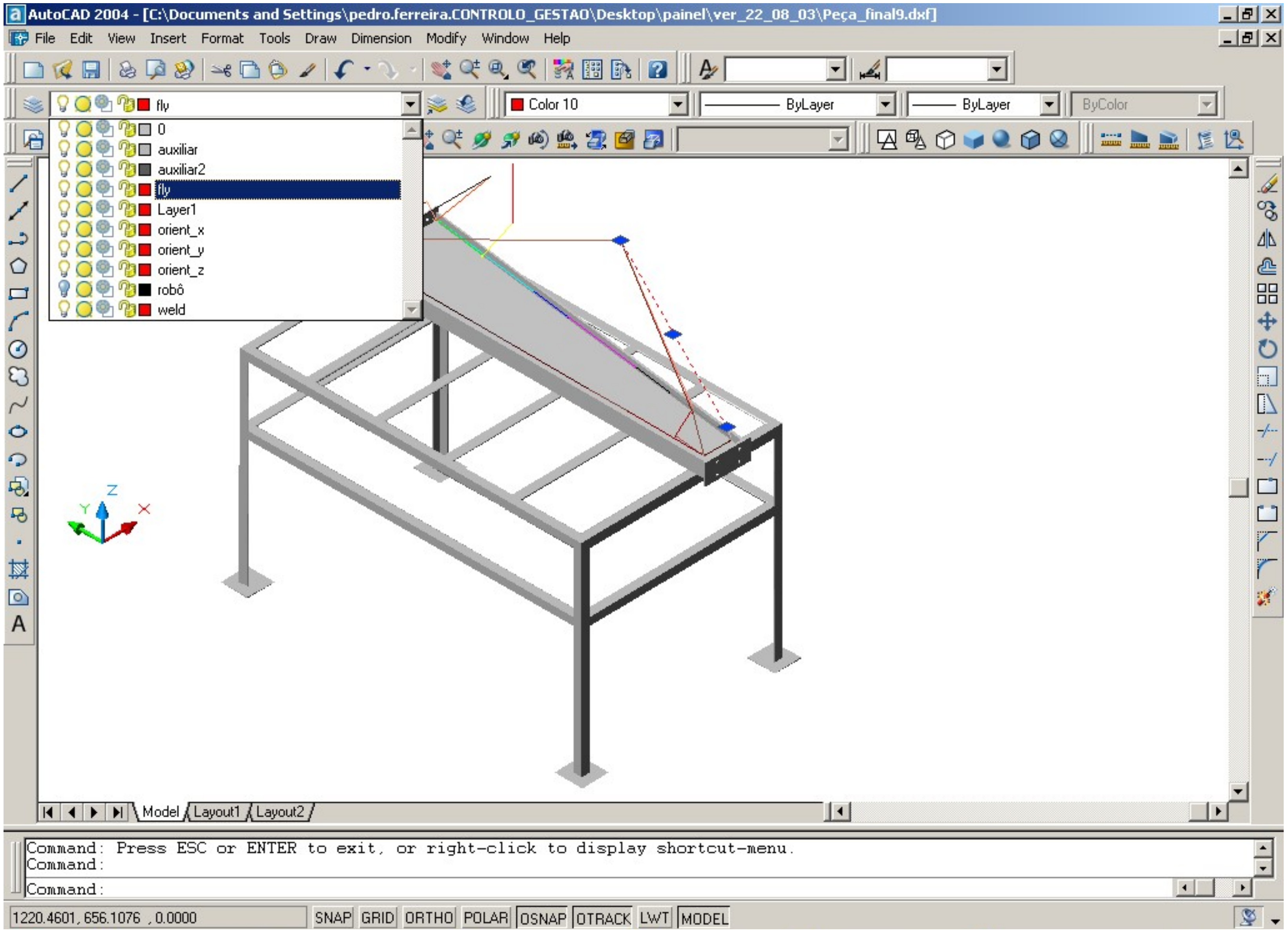



Application
(using ActiveX component and/or TPC-IP sockets)
PC
Sensors
Actuators
IO, fieldbus or specific card driver
User Interface
Object A
Object B
PCROB
TCP/UDP Client
Message Handler
Message Queue
Synchronous call
Synchronous answer
Server
Portmap
OPC AE server
User Interface
Object A
Object B
OPC client
Application
(OPC client)
OPC DA server
Event: Asynchronous call
Asynchronous call and answer
Local Area Network
PLC
Robot Controller
Services
Server
Sensors
Actuators
IO PLC-like interface and fieldbus
Server
Services
Robot Control System
Sensors
Actuators
Robot

welding application – CAD interface



welding application – CAD interface



AutoCAD 2004 - [C:\Documents and Settings\pedro.ferreira.CONTROLO_GESTAO\Desktop\painel\ver_22_08_03\Peça_final9.dxf]

```
Command: Press ESC or ENTER to exit, or right-click to display shortcut-menu.
Command:
Command:
```

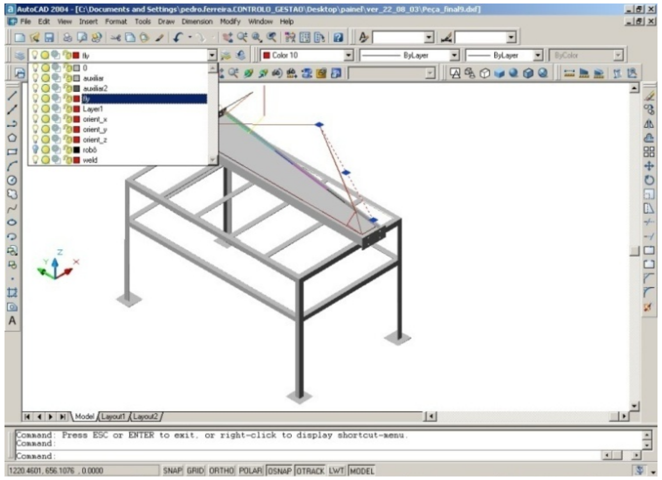
1220.4601, 656.1076 , 0.0000

welding application – CAD interface

Welding point definition

Welding_Test	Name of the file
4	Number of points
Point 1	Point number X
Origin	Name of the point
1	Type of point welding – 0, fly-by - 1)
656.419922	x value
-444.451813	y value
730.853149	z value
0.091980	Quaternion q1
0.001690	Quaternion q2
0.995760	Quaternion q3
0.002070	Quaternion q4
0	cf1
-1	cf4
0	cf6
0	cfx
8999999488	External joint 1
8999999488	External joint 1
8999999488	External joint 1
8999999488	External joint 1
8999999488	External joint 1
8999999488	External joint 1
0.00	Current [A]
0.00	Voltage [V]
100	Velocity [mm/s]
5	Precision
0	Type of motion (linear – 0, circular – 1, joints - 2)

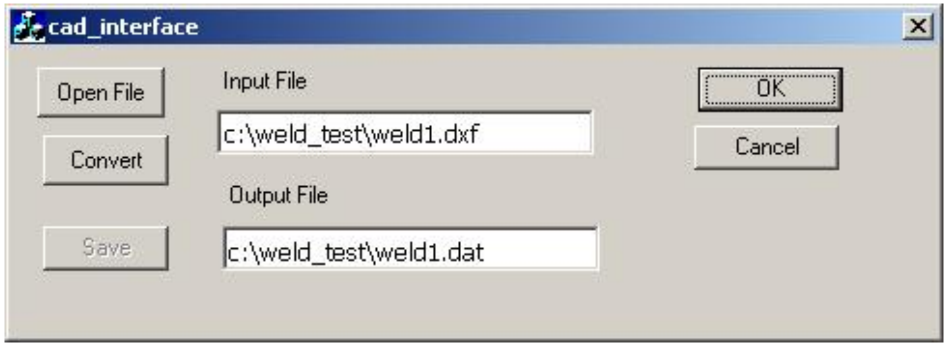
welding application – CAD interface


AutoCAD 2004 - [C:\Documents and Settings\pedro.ferreira.CONTROLO_GESTAO\Desktop\painel\ver_22_08_03\Peca_final9.dxf]
File Edit View Insert Format Tools Draw Dimension Modify Window Help
0
auxiliar
auxiliar2
lly
Layer1
orient_x
orient_y
orient_z
robô
weld
Color 10
ByLayer
ByColor
Model
Layout1
Layout2
SNAP GRID ORTHO POLAR OSNAP OTRACK LWT MODEL

```
Command: Press ESC or ENTER to exit, or right-click to display shortcut-menu.
Command:
Command:
```

1220.4601, 656.1076 , 0.0000

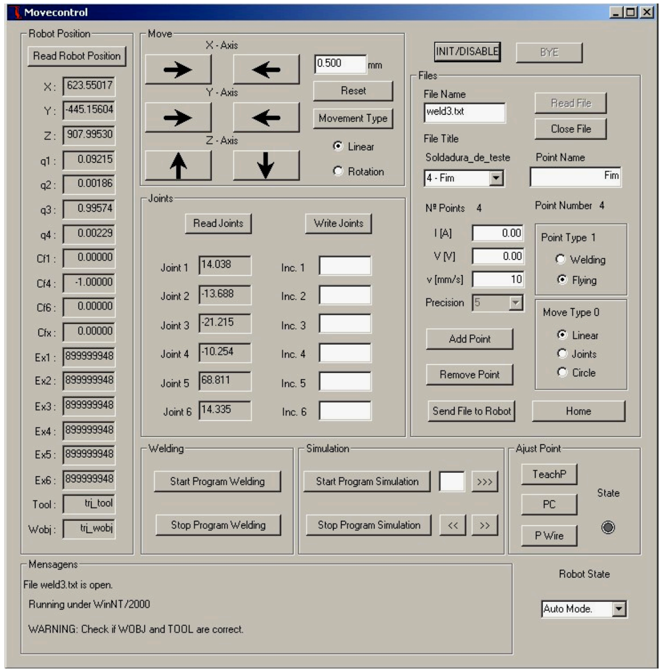
AutoCAD 2004



Input File: c:\weld_test\weld1.dxf

Output File: c:\weld_test\weld1.dat

Visual Basic .NET



X :	623.55017
Y :	-445.15604
Z :	907.99530
q1 :	0.09215
q2 :	0.00186
q3 :	0.99574
q4 :	0.00229
Cf1 :	0.00000
Cf4 :	-1.00000
Cf6 :	0.00000
Cfx :	0.00000
Ex1 :	899999948
Ex2 :	899999948
Ex3 :	899999948
Ex4 :	899999948
Ex5 :	899999948
Ex6 :	899999948
Tool :	trj_tool
Wobj :	trj_wobj

Move step: 0.500 mm

Joint 1	14.038	Inc. 1	
Joint 2	-13.688	Inc. 2	
Joint 3	-21.215	Inc. 3	
Joint 4	-10.254	Inc. 4	
Joint 5	68.811	Inc. 5	
Joint 6	14.335	Inc. 6	

File Name: weld3.txt

File Title: Soldadura_de_teste

4 - Fim

Point Name: Fim

Nº Points 4

Point Number 4

I [A]: 0.00

V [V]: 0.00

v [mm/s]: 10

Precision: 5

Point Type 1: Welding [✓ Flying]

Move Type 0: [✓ Linear] Joints Circle

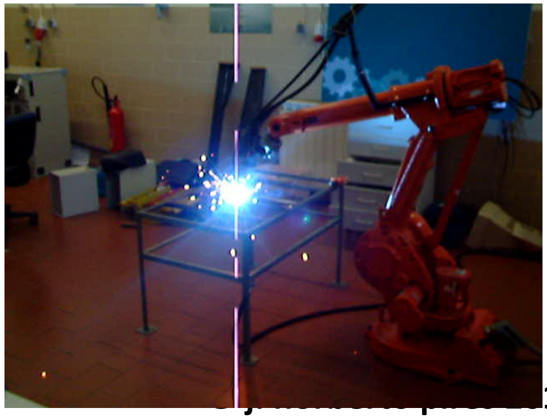
File weld3.txt is open.

Running under WinNT/2000

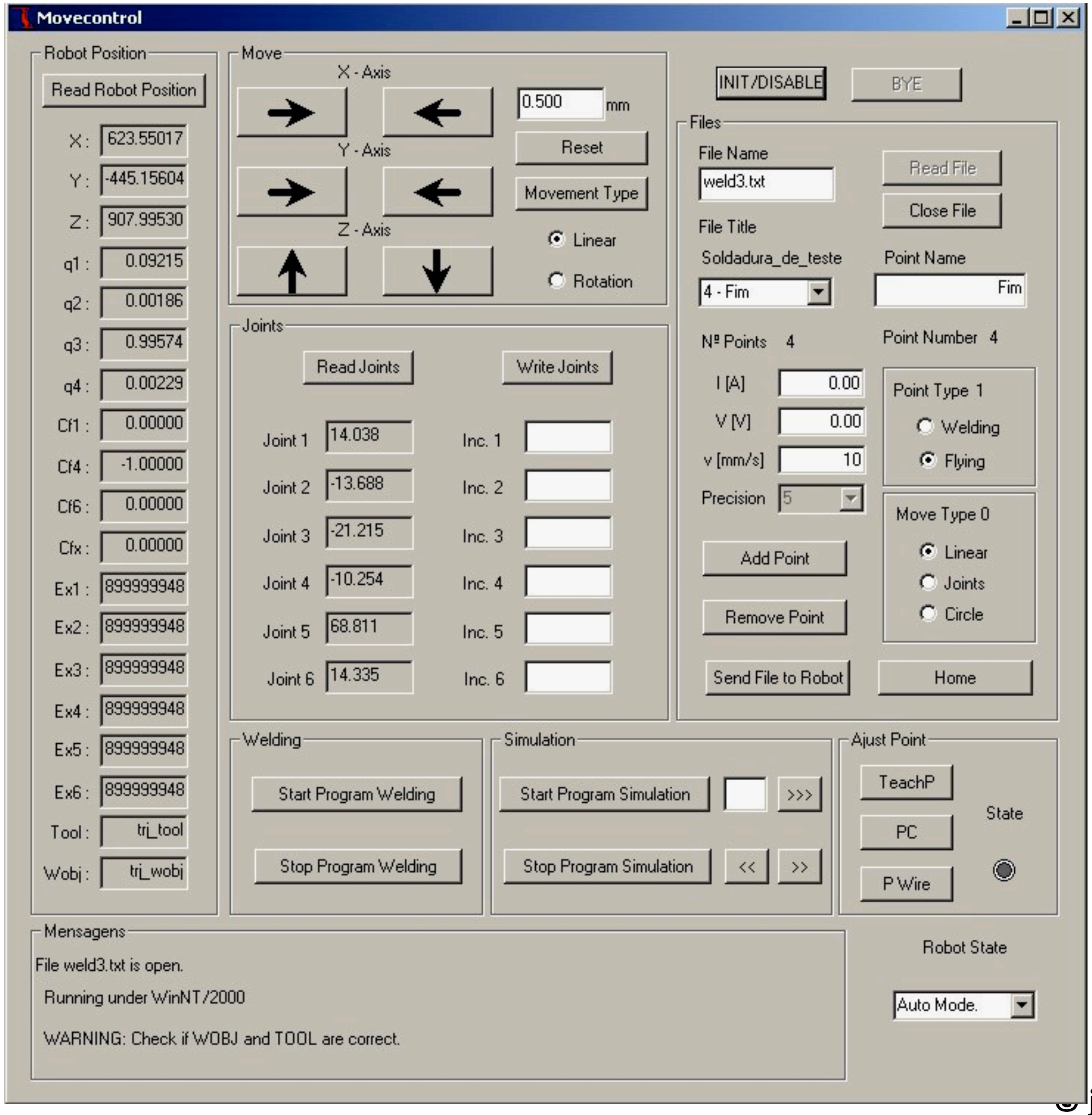
WARNING: Check if WOBJ and TOOL are correct.

Robot State: Auto Mode.

Visual C++ .NET



welding application – CAD interface



X : 623.55017
Y : -445.15604
Z : 907.99530
q1 : 0.09215
q2 : 0.00186
q3 : 0.99574
q4 : 0.00229
Cf1 : 0.00000
Cf4 : -1.00000
Cf6 : 0.00000
Cfx : 0.00000
Ex1 : 899999948
Ex2 : 899999948
Ex3 : 899999948
Ex4 : 899999948
Ex5 : 899999948
Ex6 : 899999948
Tool : tri_tool
Wobj : tri_wobj

0.500 mm

[✓ Linear] Rotation

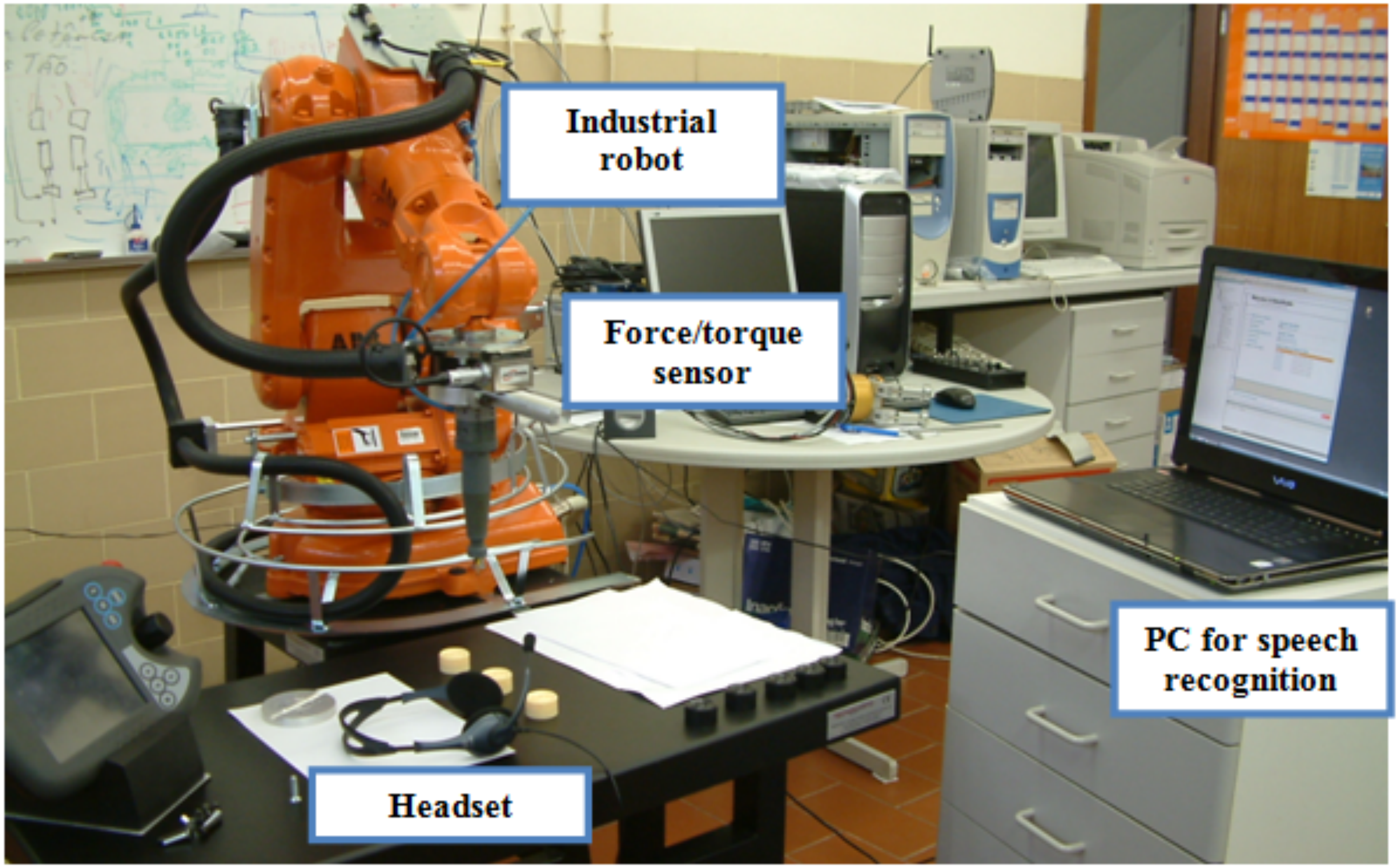
Joint 1 14.038 Inc. 1
Joint 2 -13.688 Inc. 2
Joint 3 -21.215 Inc. 3
Joint 4 -10.254 Inc. 4
Joint 5 68.811 Inc. 5
Joint 6 14.335 Inc. 6

File Name: weld3.txt
File Title: Soldadura_de_teste
4 - Fim
Point Name: Fim
Nº Points 4
Point Number 4
I [A] 0.00
V [V] 0.00
v [mm/s] 10
Precision 5
Point Type 1: Welding [✓ Flying]
Move Type 0: [✓ Linear] Joints Circle

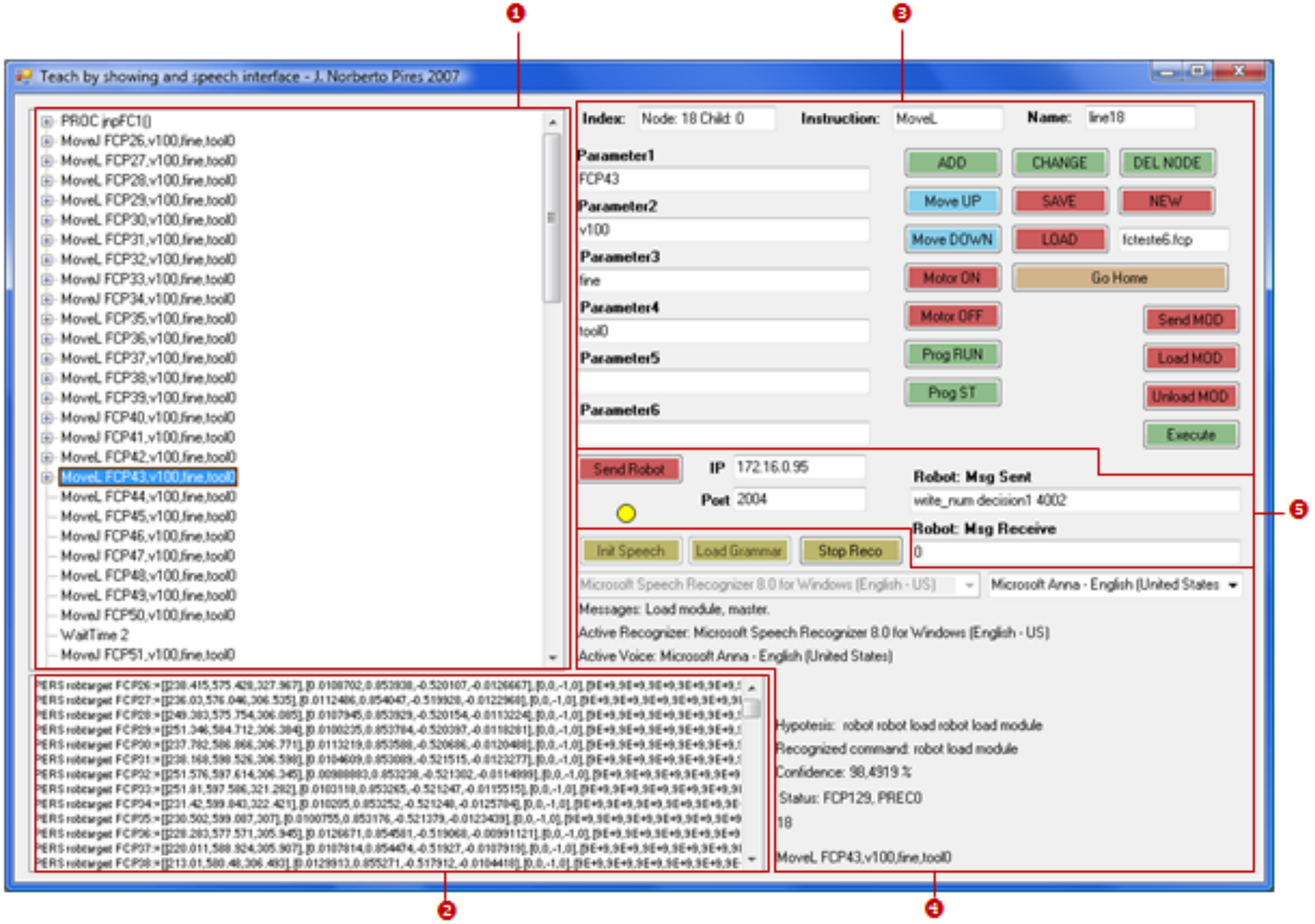
File weld3.txt is open.
Running under WinNT/2000
WARNING: Check if WOBJ and TOOL are correct.

Auto Mode.

using speech – programming-by demonstration



using speech – programming-by demonstration



```
PROC jnpFC1()
MoveJ FCP26,v100,fine,tool0
MoveL FCP27,v100,fine,tool0
MoveL FCP28,v100,fine,tool0
MoveL FCP29,v100,fine,tool0
MoveL FCP30,v100,fine,tool0
MoveL FCP31,v100,fine,tool0
MoveL FCP32,v100,fine,tool0
MoveJ FCP33,v100,fine,tool0
MoveJ FCP34,v100,fine,tool0
MoveL FCP35,v100,fine,tool0
MoveL FCP36,v100,fine,tool0
MoveL FCP37,v100,fine,tool0
MoveL FCP38,v100,fine,tool0
MoveL FCP39,v100,fine,tool0
MoveJ FCP40,v100,fine,tool0
MoveJ FCP41,v100,fine,tool0
MoveL FCP42,v100,fine,tool0
MoveL FCP43,v100,fine,tool0
MoveL FCP44,v100,fine,tool0
MoveL FCP45,v100,fine,tool0
MoveJ FCP46,v100,fine,tool0
MoveJ FCP47,v100,fine,tool0
MoveL FCP48,v100,fine,tool0
MoveL FCP49,v100,fine,tool0
MoveJ FCP50,v100,fine,tool0
WaitTime 2
MoveJ FCP51,v100,fine,tool0
```

Index: Node: 18 Child: 0 Instruction: MoveL Name: line18

Parameter1: FCP43
Parameter2: v100
Parameter3: fine
Parameter4: tool0
Parameter5:
Parameter6:

lcteste6.fcp

IP: 172.16.0.95
Port: 2004

Robot: Msg Sent: write_num decision1 4002
Robot: Msg Receive: 0

Microsoft Speech Recognizer 8.0 for Windows (English - US)
Microsoft Anna - English (United States)

```
Messages: Load module, master.
Active Recognizer: Microsoft Speech Recognizer 8.0 for Windows (English - US)
Active Voice: Microsoft Anna - English (United States)
```

```
PERS robtarget FCP26:=[[238.415,575.428,327.967],[0.0108702,0.853938,-0.520107,-0.0126667],[0,0,-1,0],[9E+9,9E+9,9E+9,9E+9,9E+9,
PERS robtarget FCP27:=[[236.03,576.046,306.535],[0.0112486,0.854047,-0.519928,-0.0122968],[0,0,-1,0],[9E+9,9E+9,9E+9,9E+9,9E+9,9
PERS robtarget FCP28:=[[249.383,575.754,306.085],[0.0107945,0.853929,-0.520154,-0.0113224],[0,0,-1,0],[9E+9,9E+9,9E+9,9E+9,9E+9,
PERS robtarget FCP29:=[[251.346,584.712,306.384],[0.0100235,0.853784,-0.520397,-0.0118281],[0,0,-1,0],[9E+9,9E+9,9E+9,9E+9,9E+9,
PERS robtarget FCP30:=[[237.782,586.866,306.771],[0.0113219,0.853588,-0.520686,-0.0120488],[0,0,-1,0],[9E+9,9E+9,9E+9,9E+9,9E+9,
PERS robtarget FCP31:=[[238.168,598.526,306.598],[0.0104609,0.853089,-0.521515,-0.0123277],[0,0,-1,0],[9E+9,9E+9,9E+9,9E+9,9E+9,
PERS robtarget FCP32:=[[251.576,597.614,306.345],[0.00988883,0.853238,-0.521302,-0.0114899],[0,0,-1,0],[9E+9,9E+9,9E+9,9E+9,9E+9
PERS robtarget FCP33:=[[251.81,597.586,321.282],[0.0103118,0.853265,-0.521247,-0.0115515],[0,0,-1,0],[9E+9,9E+9,9E+9,9E+9,9E+9,9
PERS robtarget FCP34:=[[231.42,589.043,322.421],[0.010205,0.853252,-0.521248,-0.0125784],[0,0,-1,0],[9E+9,9E+9,9E+9,9E+9,9E+9,9E
PERS robtarget FCP35:=[[230.502,599.087,307],[0.0100755,0.853176,-0.521379,-0.0123439],[0,0,-1,0],[9E+9,9E+9,9E+9,9E+9,9E+9,9E+9
PERS robtarget FCP36:=[[228.283,577.571,305.945],[0.0126671,0.854581,-0.519068,-0.00991121],[0,0,-1,0],[9E+9,9E+9,9E+9,9E+9,9E+9
PERS robtarget FCP37:=[[220.011,588.924,305.907],[0.0107814,0.854474,-0.51927,-0.0107919],[0,0,-1,0],[9E+9,9E+9,9E+9,9E+9,9E+9,9
PERS robtarget FCP38:=[[213.01,580.48,306.483],[0.0129913,0.855271,-0.517912,-0.0104418],[0,0,-1,0],[9E+9,9E+9,9E+9,9E+9,9E+9,9E-
```

```
Hypotesis: robot robot load robot load module
Recognized command: robot load module
Confidence: 98,4919 %
Status: FCP129, PRECO
18
MoveL FCP43,v100,fine,tool0
```

using speech – programming-by demonstration

- Working based on XML grammars

- Actually using Microsoft SAPI 5.3, but it's independent of SR and TTS engines.

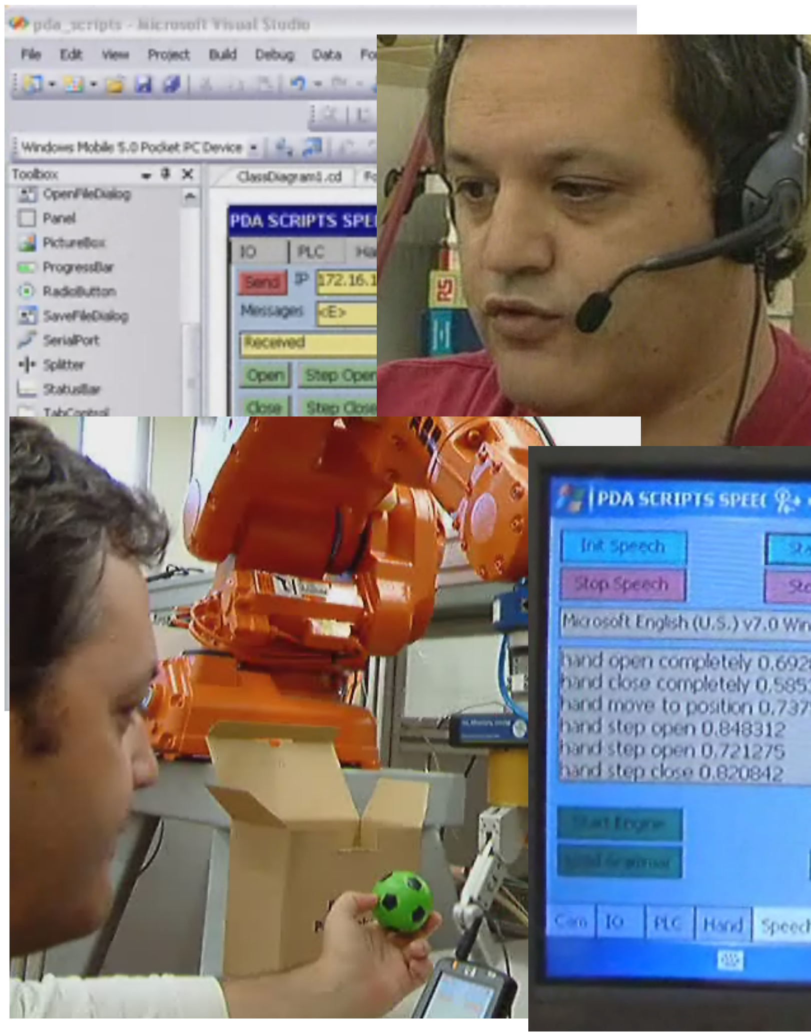
- Fully functional for English and Portuguese

- Fully tested at laboratory, using 80dB noise levels

- Fully tested under Windows 7/Vista/Xp, Windows CE and Windows Mobile 2003/2005

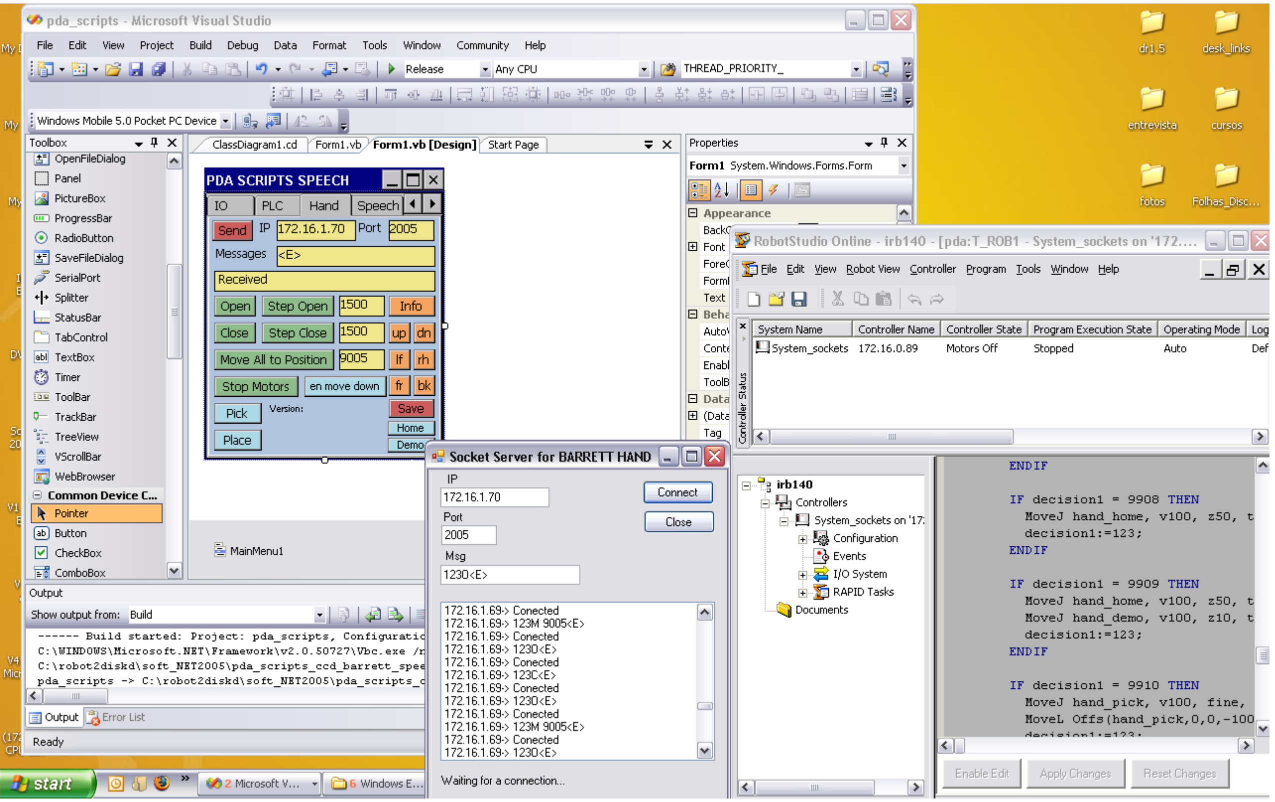
- Fully tested using ABB robots (S4 and IRC5 controllers) and a few MOTOMAN robots (latest controller only)

- Cooperation established with **MLDC** – ***Microsoft Language Development Centre*** to add dictation features to upgrade the actual command based approach.



```
hand open completely 0.692
hand close completely 0.585
hand move to position 0.737
hand step open 0.848312
hand step open 0.721275
hand step close 0.820842
```

PDA interfaces



```
------ Build started: Project: pda_scripts, Configuratic
C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\Vbc.exe /r
C:\robot2diskd\soft_NET2005\pda_scripts_ccd_barrett_spee
pda_scripts -> C:\robot2diskd\soft_NET2005\pda_scripts_c
```

System Name	Controller Name	Controller State	Program Execution State	Operating Mode	Log
System_sockets	172.16.0.89	Motors Off	Stopped	Auto	Def

Socket Server for BARRETT HAND

IP: 172.16.1.70
Port: 2005
Msg: 1230<E>

```
172.16.1.69-> Conected
172.16.1.69-> 123M 9005<E>
172.16.1.69-> Conected
172.16.1.69-> 1230<E>
172.16.1.69-> Conected
172.16.1.69-> 123C<E>
172.16.1.69-> Conected
172.16.1.69-> 1230<E>
172.16.1.69-> Conected
172.16.1.69-> 123M 9005<E>
172.16.1.69-> Conected
172.16.1.69-> 1230<E>
```

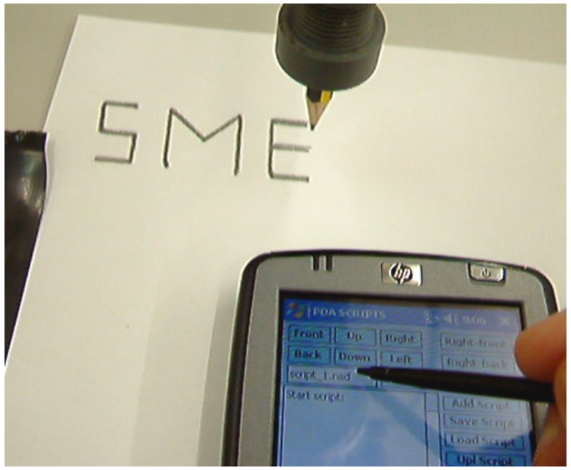
Waiting for a connection...

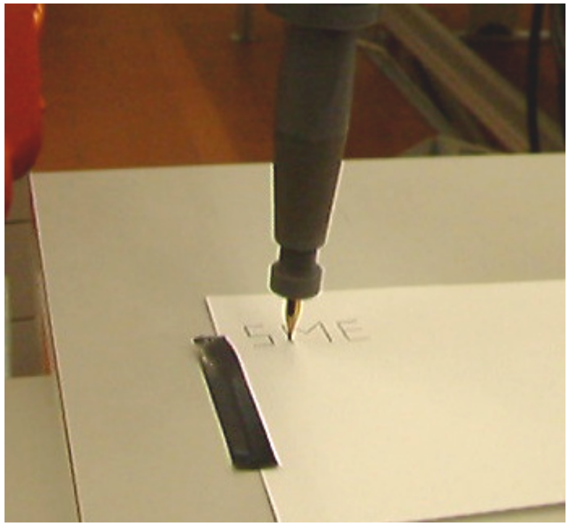
```
    ENDIF

    IF decision1 = 9908 THEN
      MoveJ hand_home, v100, z50, t
      decision1:=123;
    ENDIF

    IF decision1 = 9909 THEN
      MoveJ hand_home, v100, z50, t
      MoveJ hand_demo, v100, z10, t
      decision1:=123;
    ENDIF

    IF decision1 = 9910 THEN
      MoveJ hand_pick, v100, fine,
      MoveL Offs(hand_pick,0,0,-100
```





- PDA interfaces based on TCP, UDP and RPC sockets.
- Work with any robot controller
- Fully programmable, using regular programming tools: VB, C#, C++
- Fully tested on PDAs running WinCE
- Developments can be exported to any WinCE device.
- Useful for robot programming: teach and program

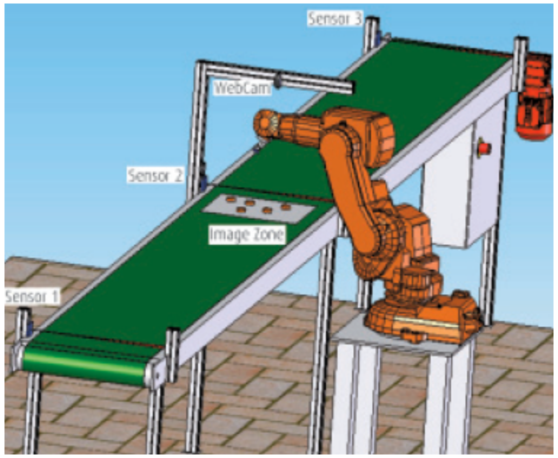
service oriented architectures

Service Oriented Architectures:

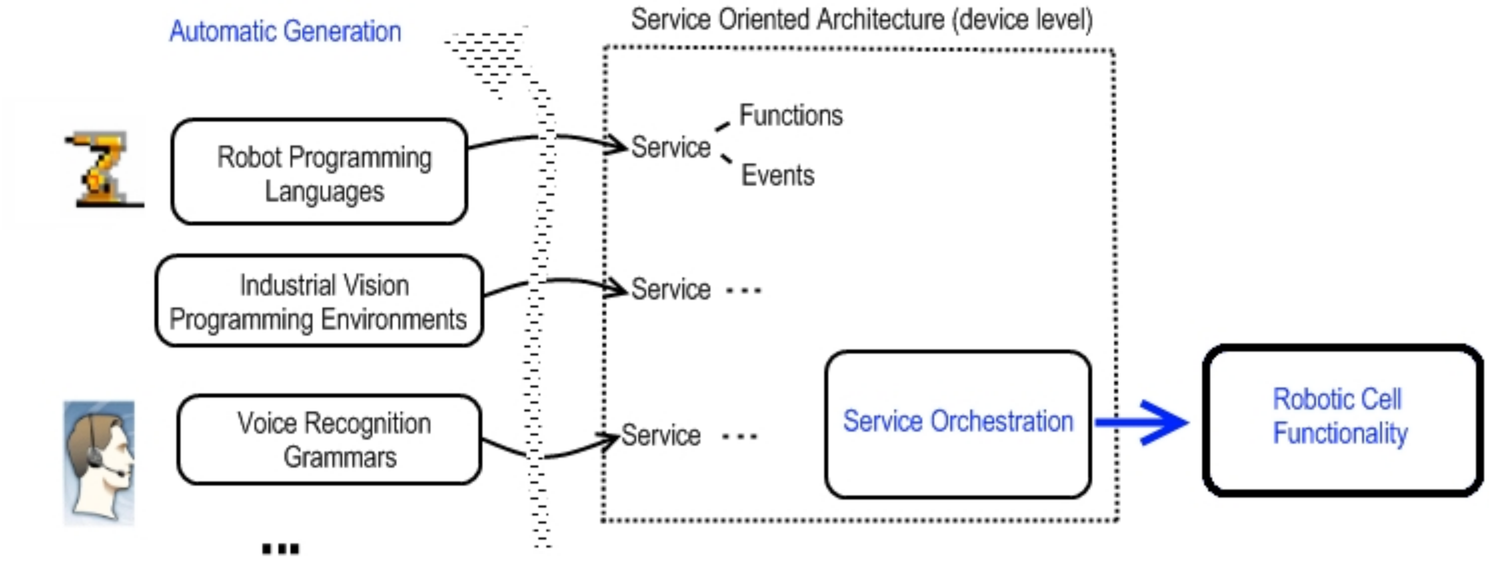
- Cameras, PLC's, RFID systems, …
- Easy to program hard to integrate.

Robotic work-cell programming:

- Traditional programming languages (C#, Java,C++)
- Problem with the separation of concerns, computation and coordination, limits reusability.

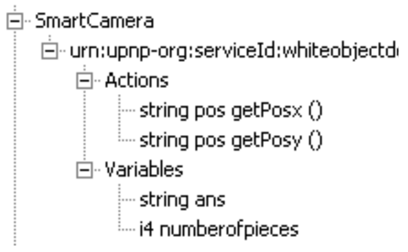


Test platform



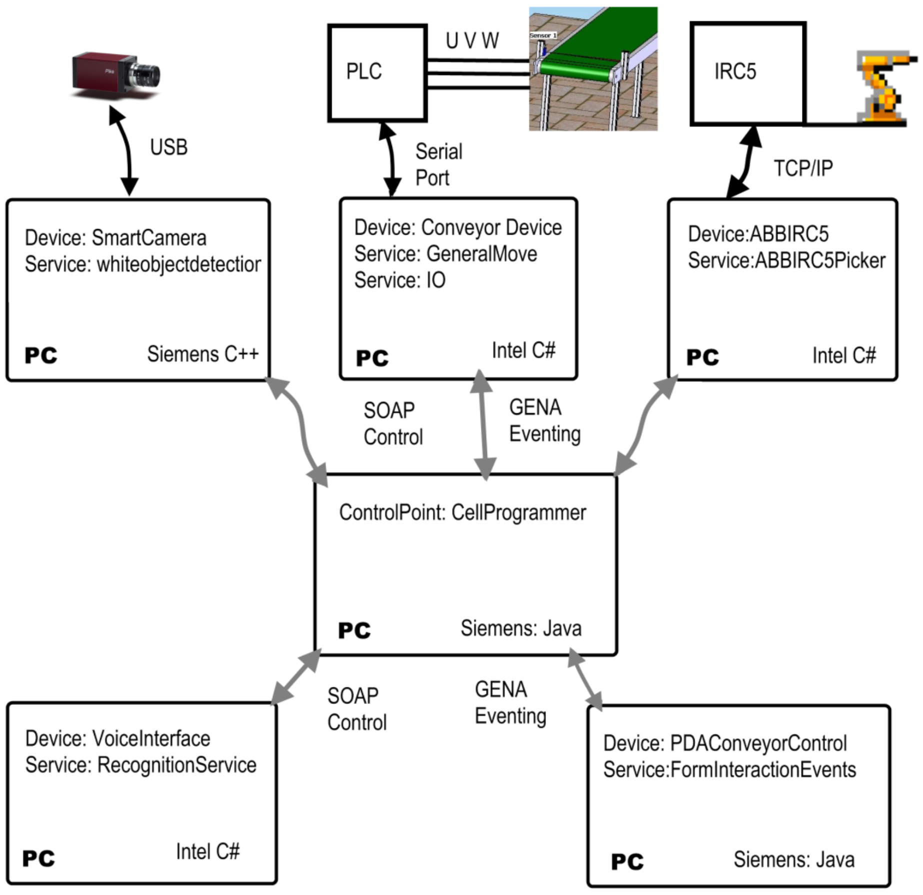
Automatic Generation
Service Oriented Architecture (device level)
Robot Programming Languages
Industrial Vision Programming Environments
Voice Recognition Grammars
...
Service
Functions
Events
Service ...
Service ...
Service Orchestration
Robotic Cell Functionality

service oriented architectures

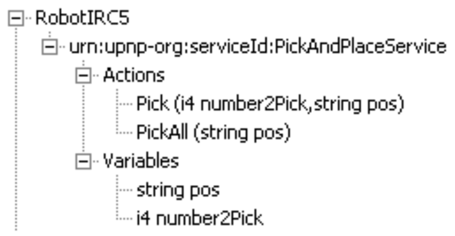


```
⊟ SmartCamera
   ⊟ urn:upnp-org:serviceId:whiteobjectd
      ⊟ Actions
           string pos getPosx ()
           string pos getPosy ()
      ⊟ Variables
           string ans
           i4 numberofpieces
```

SmartCamera UPnP device

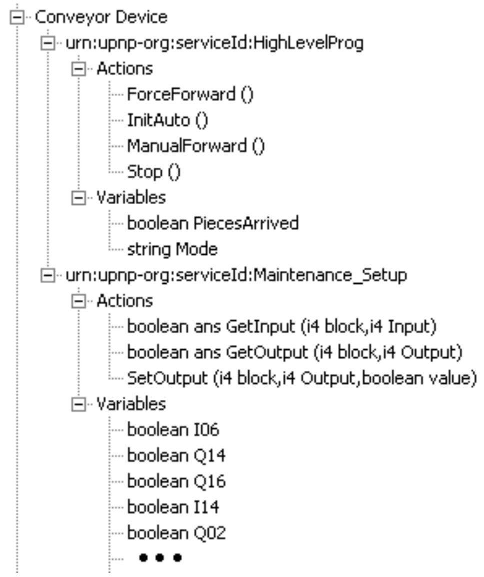


UPnP network



```
⊟ RobotIRC5
   ⊟ urn:upnp-org:serviceId:PickAndPlaceService
      ⊟ Actions
           Pick (i4 number2Pick,string pos)
           PickAll (string pos)
      ⊟ Variables
           string pos
           i4 number2Pick
```

Robot IRC5 UPnP device



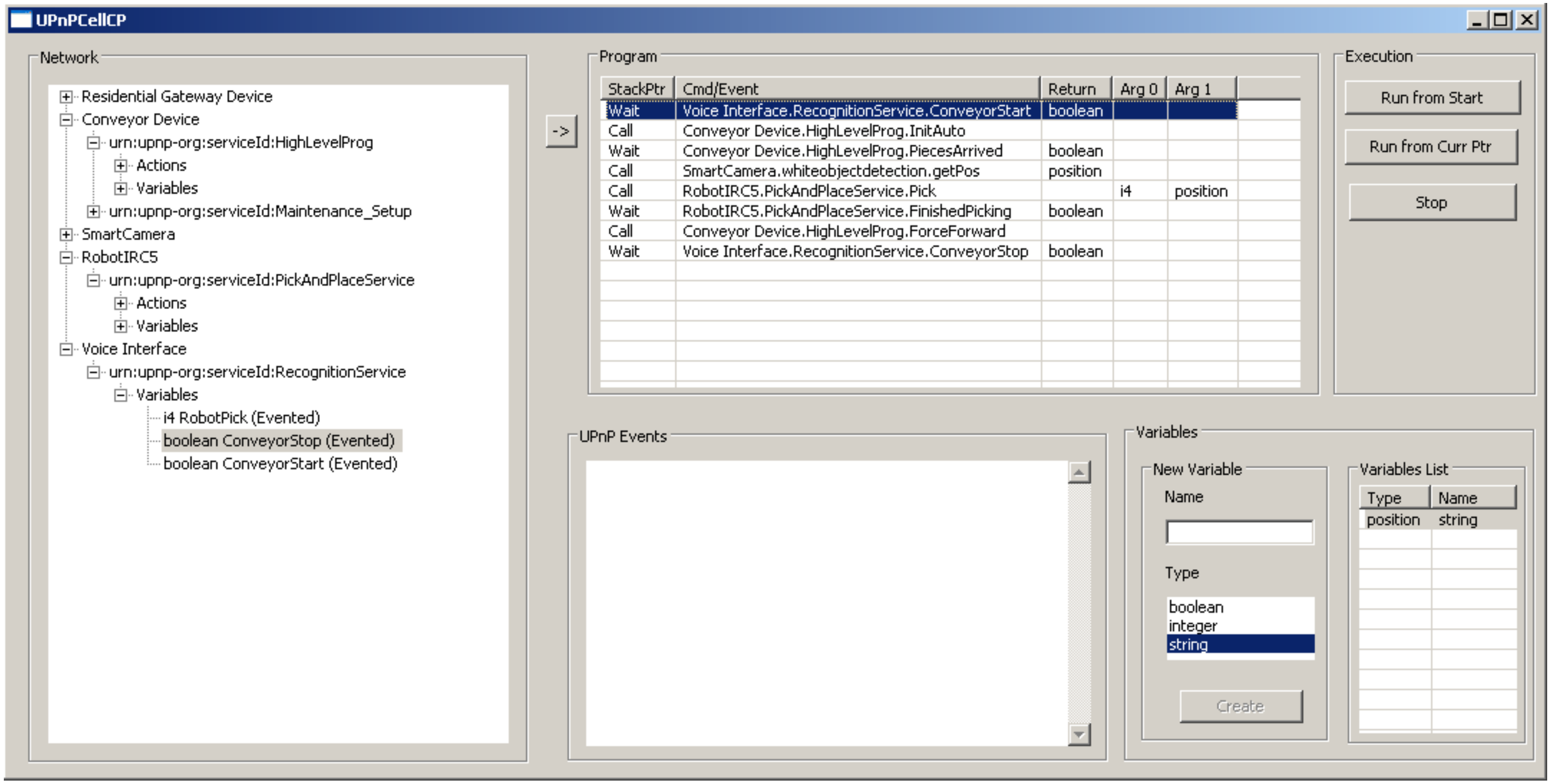
```
⊟ Conveyor Device
   ⊟ urn:upnp-org:serviceId:HighLevelProg
      ⊟ Actions
           ForceForward ()
           InitAuto ()
           ManualForward ()
           Stop ()
      ⊟ Variables
           boolean PiecesArrived
           string Mode
   ⊟ urn:upnp-org:serviceId:Maintenance_Setup
      ⊟ Actions
           boolean ans GetInput (i4 block,i4 Input)
           boolean ans GetOutput (i4 block,i4 Output)
           SetOutput (i4 block,i4 Output,boolean value)
      ⊟ Variables
           boolean I06
           boolean Q14
           boolean Q16
           boolean I14
           boolean Q02
           • • •
```

Conveyor UPnP device

Robot Programs to service contracts

- Step forward to the dissemination of SOA in industrial environment
- Complete compatibility with the existing robot technology
- Significant reduction on cell setup time
- Gives the robot programmer power to expose fuctionality

service oriented architectures



UPnPCellCP

Network

- Residential Gateway Device
- Conveyor Device
  - urn:upnp-org:serviceId:HighLevelProg
    - Actions
    - Variables
  - urn:upnp-org:serviceId:Maintenance_Setup
- SmartCamera
- RobotIRC5
  - urn:upnp-org:serviceId:PickAndPlaceService
    - Actions
    - Variables
- Voice Interface
  - urn:upnp-org:serviceId:RecognitionService
    - Variables
      - i4 RobotPick (Evented)
      - boolean ConveyorStop (Evented)
      - boolean ConveyorStart (Evented)

Program

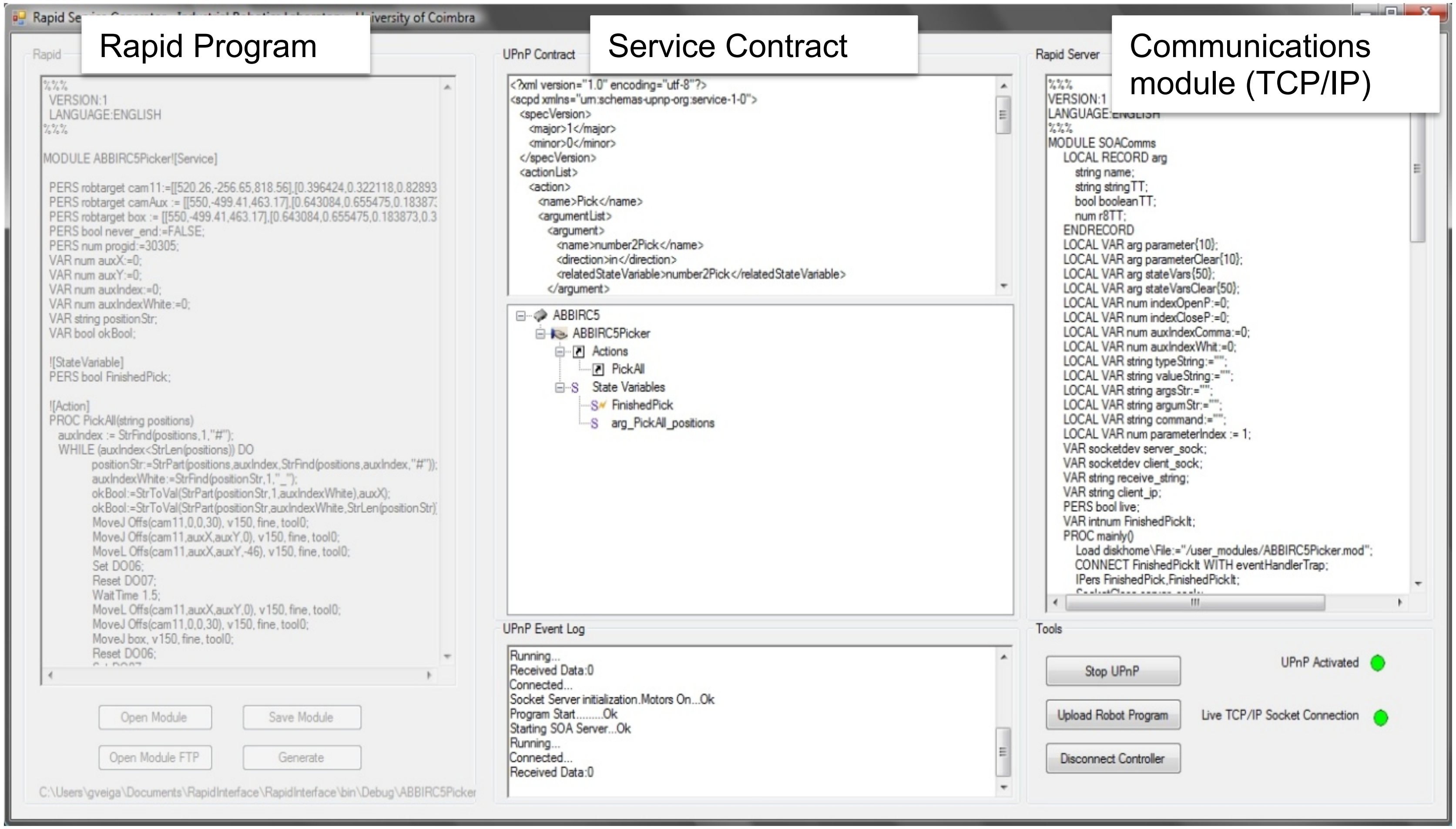
StackPtr	Cmd/Event	Return	Arg 0	Arg 1
Wait	Voice Interface.RecognitionService.ConveyorStart	boolean		
Call	Conveyor Device.HighLevelProg.InitAuto			
Wait	Conveyor Device.HighLevelProg.PiecesArrived	boolean		
Call	SmartCamera.whiteobjectdetection.getPos	position		
Call	RobotIRC5.PickAndPlaceService.Pick		i4	position
Wait	RobotIRC5.PickAndPlaceService.FinishedPicking	boolean		
Call	Conveyor Device.HighLevelProg.ForceForward			
Wait	Voice Interface.RecognitionService.ConveyorStop	boolean		

Execution

Type	Name
position	string

Cell programmer

service oriented architectures



```
%%%
 VERSION:1
 LANGUAGE:ENGLISH
%%%

MODULE ABBIRC5Picker![Service]

 PERS robtarget cam11:=[[520.26,-256.65,818.56],[0.396424,0.322118,0.82893
 PERS robtarget camAux := [[550,-499.41,463.17],[0.643084,0.655475,0.18387
 PERS robtarget box := [[550,-499.41,463.17],[0.643084,0.655475,0.183873,0.3
 PERS bool never_end:=FALSE;
 PERS num progid:=30305;
 VAR num auxX:=0;
 VAR num auxY:=0;
 VAR num auxIndex:=0;
 VAR num auxIndexWhite:=0;
 VAR string positionStr;
 VAR bool okBool;

 ![StateVariable]
 PERS bool FinishedPick;

 ![Action]
 PROC PickAll(string positions)
   auxIndex := StrFind(positions,1,"#");
   WHILE (auxIndex<StrLen(positions)) DO
          positionStr:=StrPart(positions,auxIndex,StrFind(positions,auxIndex,"#"));
          auxIndexWhite:=StrFind(positionStr,1,"_");
          okBool:=StrToVal(StrPart(positionStr,1,auxIndexWhite),auxX);
          okBool:=StrToVal(StrPart(positionStr,auxIndexWhite,StrLen(positionStr)
          MoveJ Offs(cam11,0,0,30), v150, fine, tool0;
          MoveJ Offs(cam11,auxX,auxY,0), v150, fine, tool0;
          MoveL Offs(cam11,auxX,auxY,-46), v150, fine, tool0;
          Set DO06;
          Reset DO07;
          WaitTime 1.5;
          MoveL Offs(cam11,auxX,auxY,0), v150, fine, tool0;
          MoveJ Offs(cam11,0,0,30), v150, fine, tool0;
          MoveJ box, v150, fine, tool0;
          Reset DO06;
```

C:\Users\gveiga\Documents\RapidInterface\RapidInterface\bin\Debug\ABBIRC5Picker

```
<?xml version="1.0" encoding="utf-8"?>
<scpd xmlns="urn:schemas-upnp-org:service-1-0">
  <specVersion>
    <major>1</major>
    <minor>0</minor>
  </specVersion>
  <actionList>
    <action>
      <name>Pick</name>
      <argumentList>
        <argument>
          <name>number2Pick</name>
          <direction>in</direction>
          <relatedStateVariable>number2Pick</relatedStateVariable>
        </argument>
```

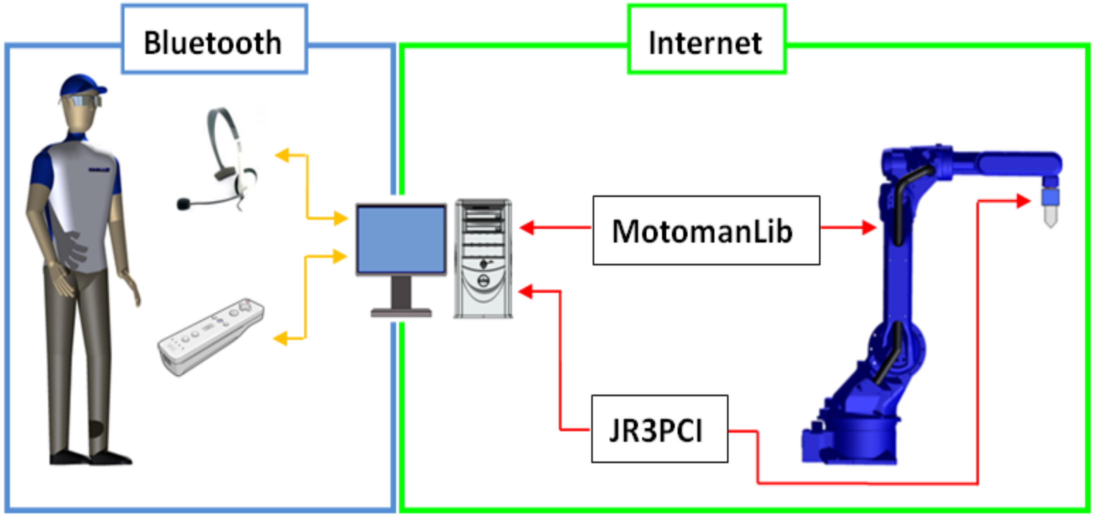
```
ABBIRC5
  ABBIRC5Picker
    Actions
      PickAll
    State Variables
      FinishedPick
      arg_PickAll_positions
```

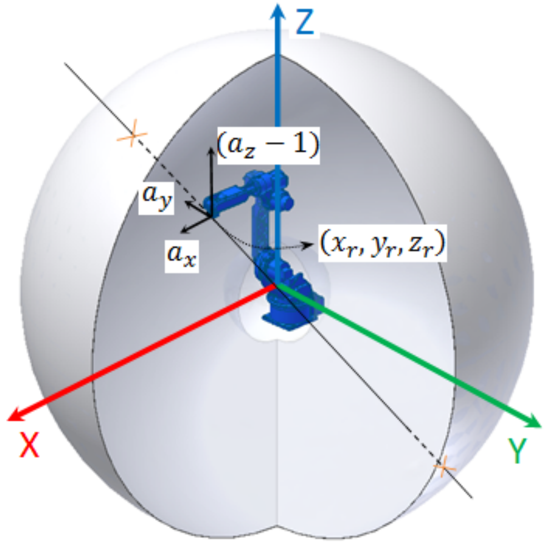
```
Running...
Received Data:0
Connected...
Socket Server initialization.Motors On...Ok
Program Start.........Ok
Starting SOA Server...Ok
Running...
Connected...
Received Data:0
```

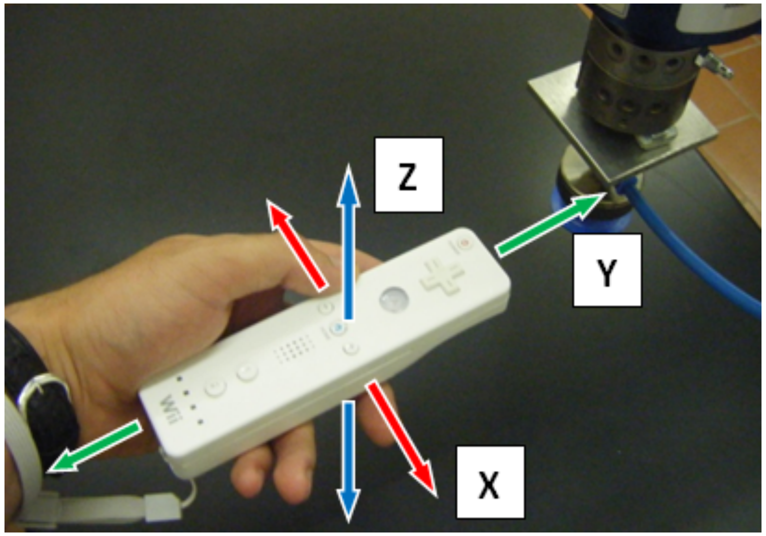
```
%%%
VERSION:1
LANGUAGE:ENGLISH
%%%
MODULE SOAComms
   LOCAL RECORD arg
      string name;
      string stringTT;
      bool booleanTT;
      num r8TT;
   ENDRECORD
   LOCAL VAR arg parameter{10};
   LOCAL VAR arg parameterClear{10};
   LOCAL VAR arg stateVars{50};
   LOCAL VAR arg stateVarsClear{50};
   LOCAL VAR num indexOpenP:=0;
   LOCAL VAR num indexCloseP:=0;
   LOCAL VAR num auxIndexComma:=0;
   LOCAL VAR num auxIndexWhit:=0;
   LOCAL VAR string typeString:="";
   LOCAL VAR string valueString:="";
   LOCAL VAR string argsStr:="";
   LOCAL VAR string argumStr:="";
   LOCAL VAR string command:="";
   LOCAL VAR num parameterIndex := 1;
   VAR socketdev server_sock;
   VAR socketdev client_sock;
   VAR string receive_string;
   VAR string client_ip;
   PERS bool live;
   VAR intnum FinishedPickIt;
   PROC mainly()
      Load diskhome\File:="/user_modules/ABBIRC5Picker.mod";
      CONNECT FinishedPickIt WITH eventHandlerTrap;
      IPers FinishedPick,FinishedPickIt;
```

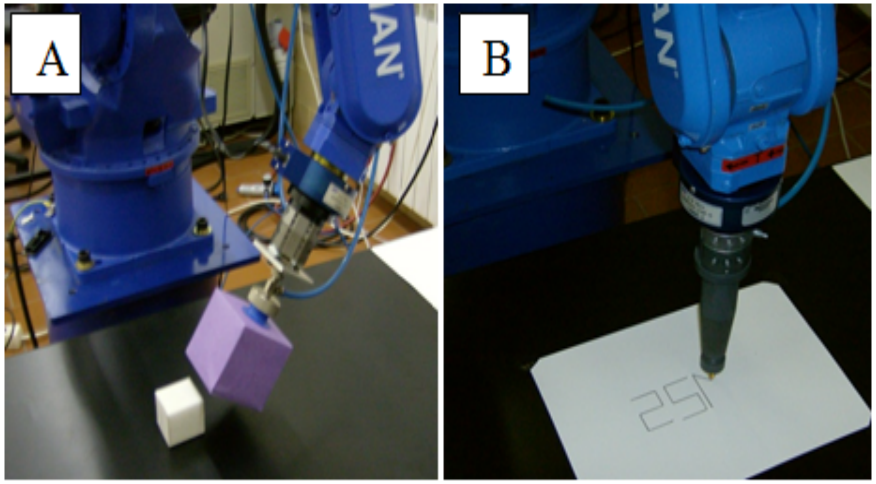
RAPID service generator

devices that capture gestures

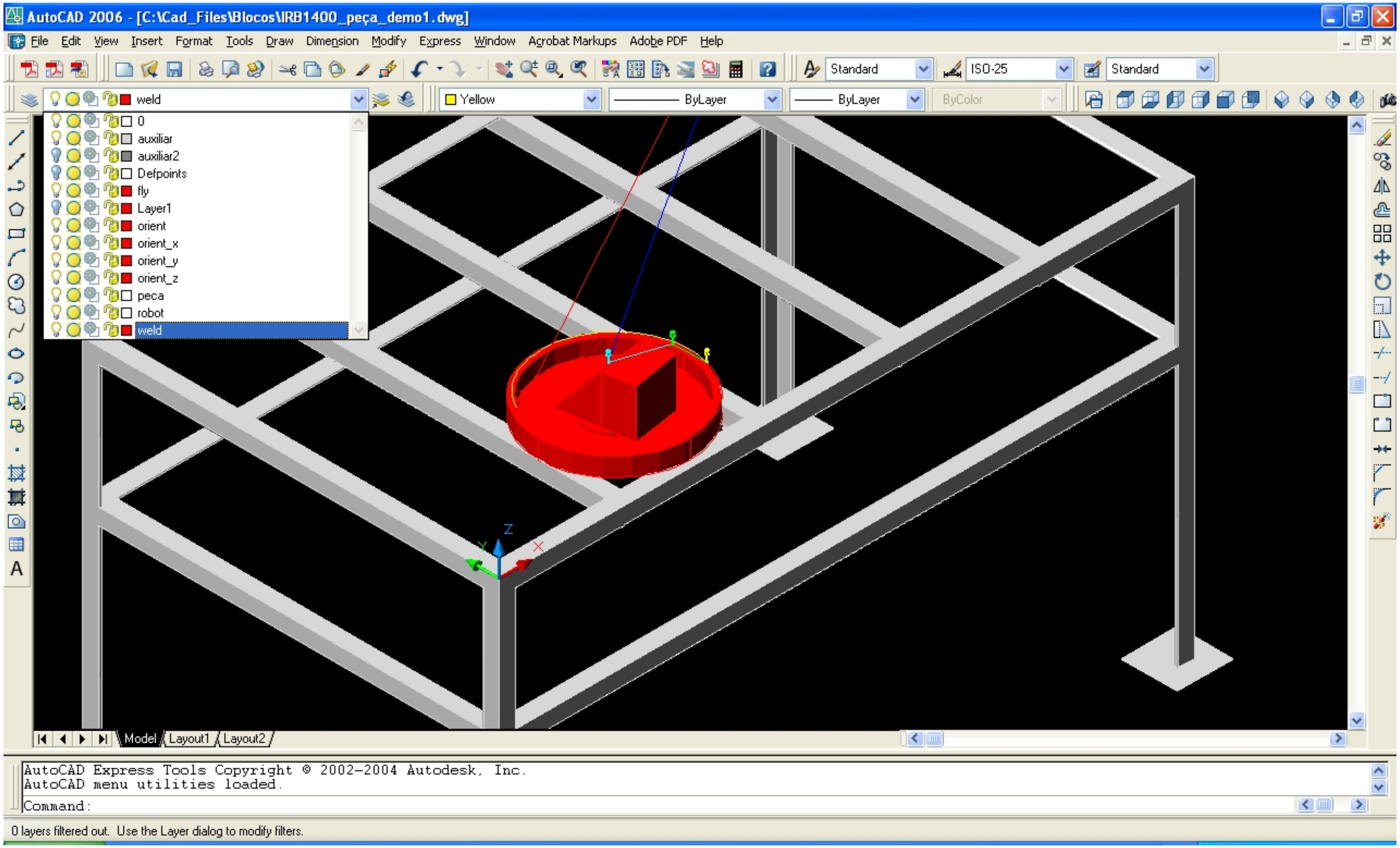








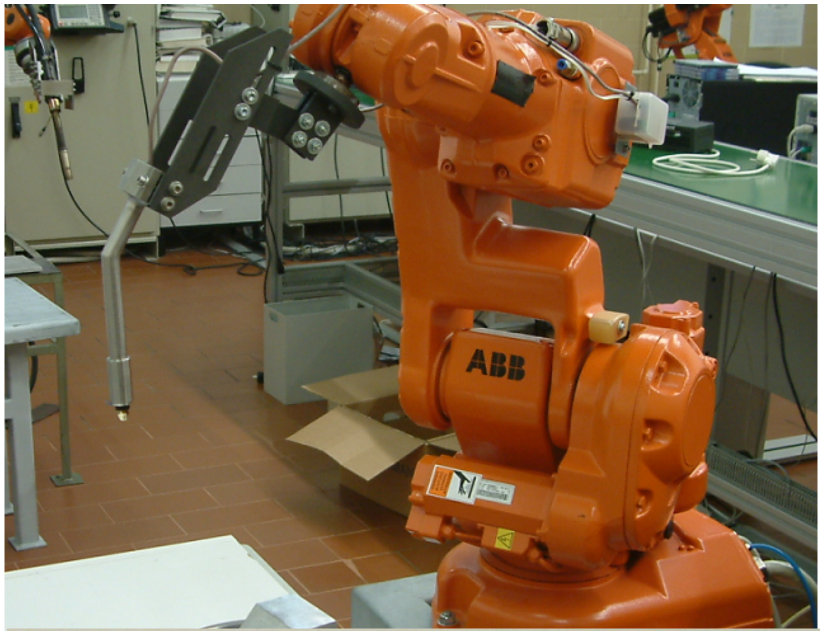
devices for programming

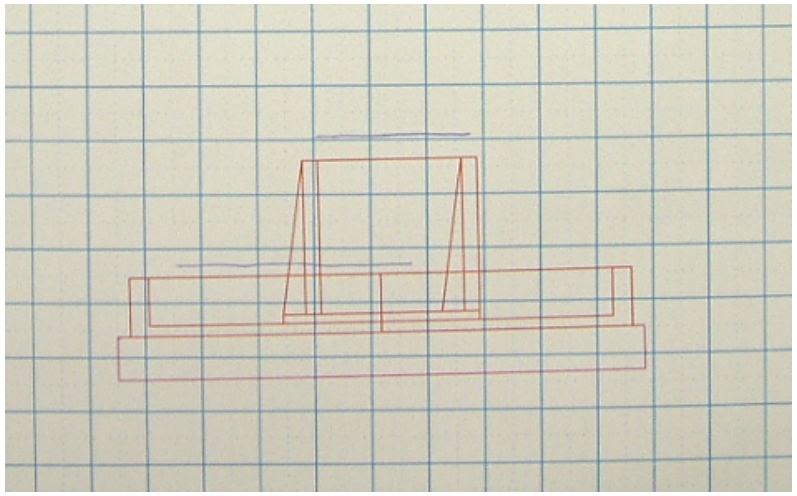

AutoCAD 2006 - [C:\Cad_Files\Blocos\IRB1400_peça_demo1.dwg]
File Edit View Insert Format Tools Draw Dimension Modify Express Window Acrobat Markups Adobe PDF Help
Standard
ISO-25
Standard
weld
0
auxiliar
auxiliar2
Defpoints
fly
Layer1
orient
orient_x
orient_y
orient_z
peca
robot
weld
Yellow
ByLayer
ByLayer
ByColor
Model
Layout1
Layout2

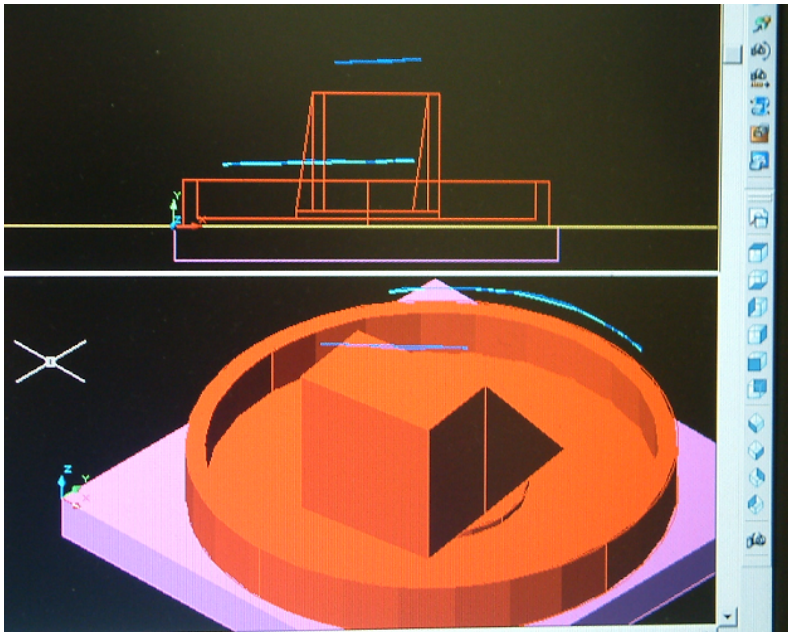
```
AutoCAD Express Tools Copyright © 2002-2004 Autodesk, Inc.
AutoCAD menu utilities loaded.
Command:
```

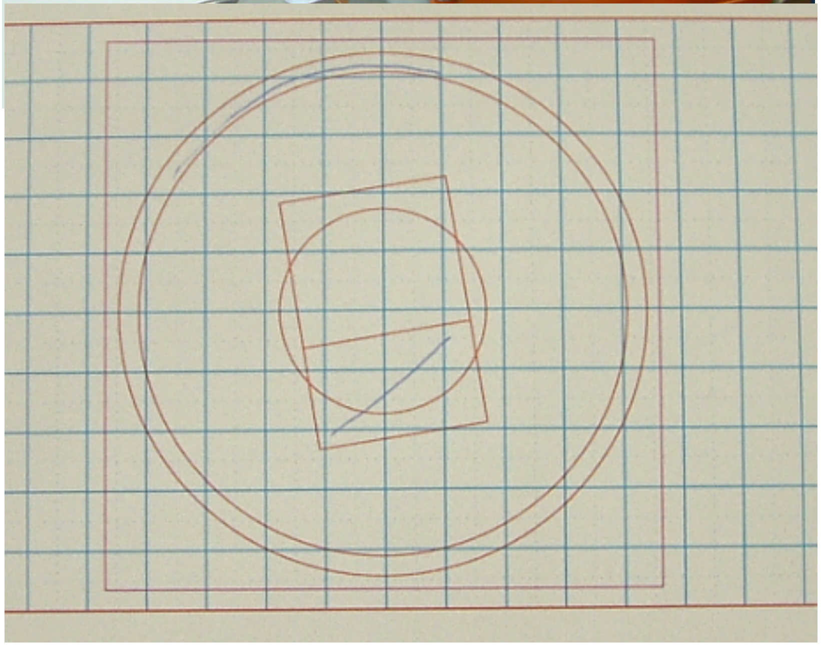
0 layers filtered out. Use the Layer dialog to modify filters.

devices for programming





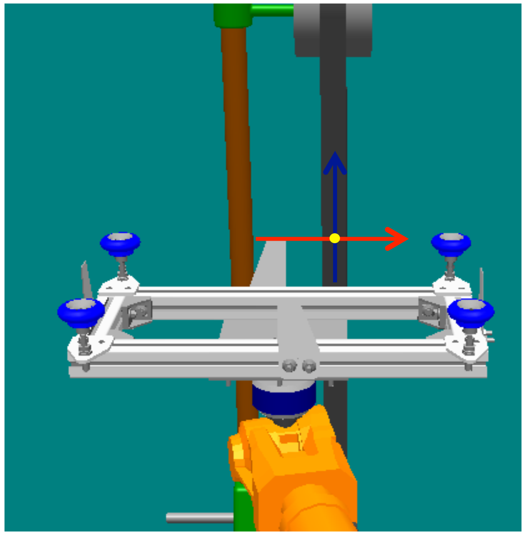




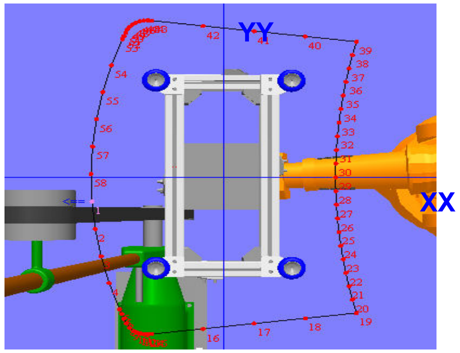
videos

The perspective here is the *coworker scenario*.

Soporcel PM2 – PM1
SSGP Rebarbagem – Modelo – Real
Albert - reconfiguração
SMErobot video
Speech with PDA
Speech Welding - vídeo
Rita Catita -vídeo
Rita Catita Mobile
Speech with noise – industrial
Demo at Stuttgart - vídeo
PDA Scripts - vídeo
Demo Biomédica - vídeo
Voice4Robotics – vídeo
IA20 - demo
"Humanoide" – vídeo
"HFeira" - vídeo



Roca tanques
Roca caixas
SSGP rebarbagem
SSGP final
SSGP Intermédio
SSGP Jardel
SOPORCEL nova
SOPORCEL Velha



team

Talent wins games, but teamwork and intelligence wins championships

Team members
J. Norberto Pires, Ph.D.
Germano Veiga, Ph.D.
Ricardo Araújo, M.Sc.
Pedro Neto, M.Sc., Ph.D. student
Nuno Mendes, M.Sc. , Ph.D. student
Gabriel Afonso, M.Sc., Ph.D. student
Bruno Vasconcelos, student
Pedro Brito, M.Sc.
Francisco Caramelo, Ph.D.
Filomena Botelho, Ph.D.
Pedro Malaca, M.Sc.

Past members
Sérgio Paulo, Ph.D.
Tiago Godinho, M.Sc.
Pedro Ferreira, M.Sc.
Jian Song, M.Sc.
Francisco Lopes, M.Sc.
Edson Citó, M.Sc.
Urbano Freitas, Student
Nélio Mourato, M.Sc.
Dário Pereira, M.Sc.

Cooperation
Miguel Dias and MLDC team, Ph.D., Microsoft
John Ramming, Ph.D., JR3 Inc.
Altino Loureiro, Ph.D., FCTUC
A. Paulo Moreira, Ph.D., FEUP

books & papers

conclusions & long term challenges

- human-machine interfaces: simpler, natural and human-like
- human-machine cooperation: coworker, hyperflexible, cognitive
- plug-and-play
- plug-and-produce
- modular design
- less expensive
- simple and natural way of programming

agility

stronger, flexible and easier automation

teamwork

real cooperative work, the robot as a human coworker

better jobs

Makes manufacturing more attractive for humans

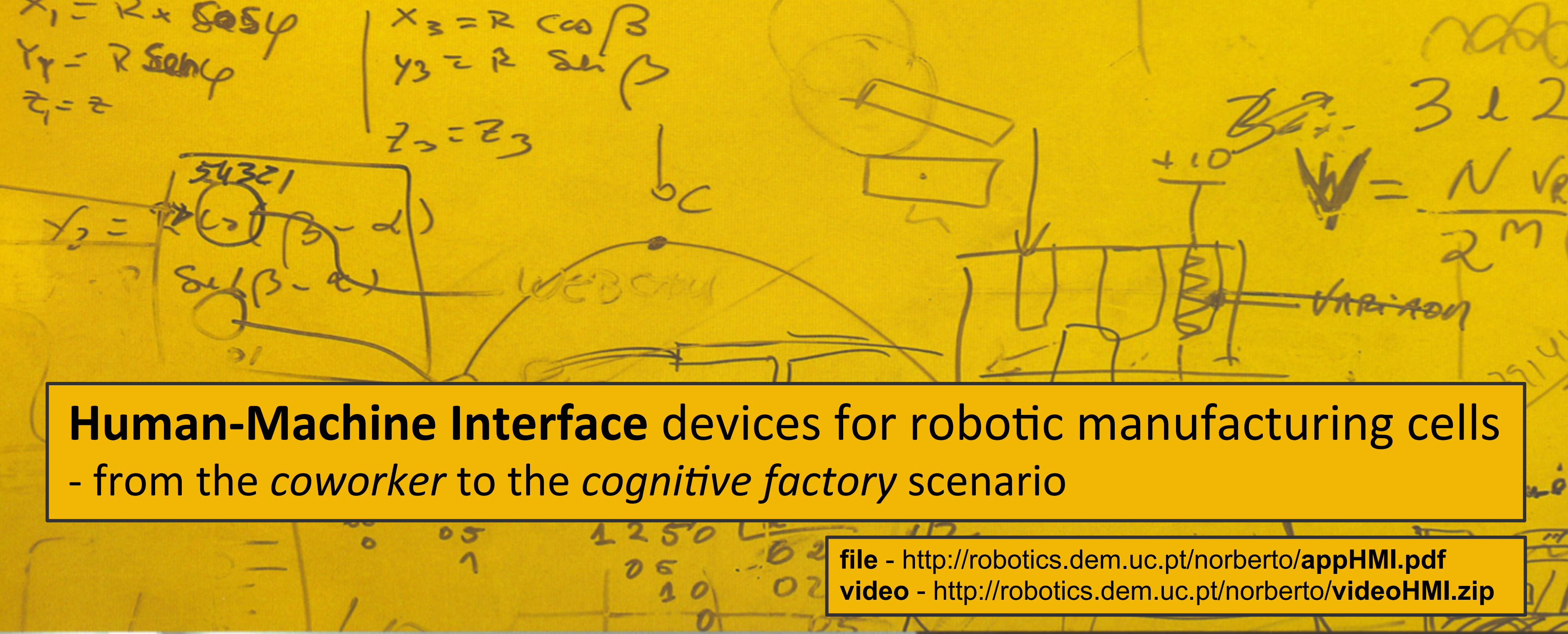
instruct a robot in a way similar to the way we “instruct” other humans

- **more software**
- **less hardware**
- **more intelligence**

flexibility

agility

think & produce



Human-Machine Interface devices for robotic manufacturing cells
- from the *coworker* to the *cognitive factory* scenario

file - http://robotics.dem.uc.pt/norberto/**appHMI.pdf**
video - http://robotics.dem.uc.pt/norberto/**videoHMI.zip**

J. Norberto Pires
Departamento de Engenharia Mecânica
Faculdade de Ciências e Tecnologia
Universidade de Coimbra

email: jnp@robotics.dem.uc.pt
web: robotics.dem.uc.pt/norberto

FCTUC FACULDADE DE CIÊNCIAS E TECNOLOGIA UC

JAI 2012
Universidade de Vigo
16 Novembro de 2012