COOPERATION IN SCIENCE AND TECHNOLOGY WITH CENTRAL AND EAST EUROPEAN COUNTRIES

PROARC

No. 7831

CAD-Based **Pro**gramming System for **Arc** Welding Robots in One-Off Production Runs

Final **R**eport

Computer & Automation Institute

University of Veszprém

WZL

date: 1/16/2001

Access: public

Contents

1 Introduction

Personal Computers have experienced an enormous increase in performance over the last years, whereas the price decreased in the same time. As a consequence, the software development community more and more moves classical workstation-based applications over to PC platforms. In this context similar actions can be recognised in the field of robotics simulation systems, which typically required high graphical performance only achievable with powerful workstations.

Lead by these new movements a novel offline-programming system for arcwelding applications, PROARC, has been developed under the ESPRIT program COPERNICUS.

The PROARC system is especially designed for the use in Small & Medium Size Enterprises and the application for small batch sizes. It is implemented on the world-leading PC-based CAD system, AutoCAD, by using its new object-oriented interface ARX. However beside this platform related attributes of the system, PROARC provides a number of efficient functionalities for the seam programming.

PROARC is capable of simulating, generating and reusing entire or partial welding sequences for robot program generation. The typical cell layout consists of a robot and a positioner. Torch cleaning and changing stations are also supported and the corresponding motions are automatically included in the welding sequence. The workpieces can have single or multiple seams and can be shifted individually. Technology macros have also been defined for fillet welds and V-type welds. Finally, PROARC is capable of importing different CAD formats as well as communicating with various robot controllers via an MMS interface.

This document now describes the elaborated requirements as well as the resulting structure and novel functionalities of the system.

Chapter 2 gives an overview over existing PC-based systems and classical connections to robot controllers and points out the strength of the PROARC's approach. The main objectives based on the evaluation of existing systems and a recherché concerning welding robots as well as welded workpiece types are summarised in chapter 3.

Chapter 4 proposes the basic structure of the entire system to allow the user associating latter described concepts to the corresponding software modules.

In chapter 5 the software structure and its modularity is presented. It will be shown, that for different domains capsulated modules have been developed.

The resulting functionalities are described in chapter 6 and the applicability of these functions is demonstrated in chapter 7.

Finally, chapter 8 discusses the realised system and its capabilities.

2 State of the Art

Recently more and more software systems not only in the fields of robotics move to the PC platform and apply PC operation systems such as Windows, Windows NT, or Windows95. Furthermore, a large robotics company even integrated a display in its teach-pendant running under Windows95.

Consequently, offline-programming systems for robotics applications as well use the enhanced capabilities of PCs.

From this group of PC-based systems three programming systems shall be mentioned, which provide the most powerful functionalities for offline programming.

The Windows based WORKSPACE system from Robot Simulations is basically used for the simulation of robotics applications. Without focusing on specific applications and providing tailored functionalities, this software especially concentrates on the calibration of the real and the modelled workcell. A special tool ROBOTRAK allows together with the use of a measurement system the update of the modelled cell and its components with a high accuracy. This is especially required for applications, in which no online sensing is or can be used. However, in the case of arc welding the arcsensor during and the tactile sensor before the welding process guarantee an appropriate welding of the seams.

The Windows based COSIMIR system from IRF (Institut für Roboterforschung, Dortmund) supports robot programming in the robot's native language including its interaction with other cell components. The system thus focuses on the accurate simulation of an entire workcell. Processes' cycle-times can be determined and new cell configuration can easily be tested for efficiency. However, this system as well lacks a support of specific welding functionalities. It is basically an offline teaching tool with advances functionalities for robotics cells' programming.

Finally, the ACT WELD from Aleph Technologies is the only PC-based system, which especially supports welding functionalities. It is a powerful system providing not only automatic generation of the welding paths and the corresponding technology data, but also a collision free path planner for the robot's movements. The system is executable on PCs (DOS) as well as on Workstations (UNIX) and besides the ACT WELD module further modules e.g. for Autonomous Vehicles, etc. are available. However, as none of the previous systems, ACT WELD neither uses common commercially available software such as the graphics modeller ACIS or the development platform AutoCAD. This is a major drawback, since software development in general proceeds very fast and by using non-common modules the reimplementation and maintenance of basic functionalities consumes a considerable amount of development time.

Thus, the PROARC system is able to concentrate on the application specific features. Some basic capabilities of the system are similar to those of ACT WELD. However, the major difference is the flexibility of the seam's programming in PROARC. No system is known, which allows a flexible restructuring of the welding sequence and the unconstrained back (modelling) and forth (simulation) stepping within a programming session. Thus, the PROARC module certainly is a useful tool for the efficient programming of welding applications.

Beside the programming of welding tasks, another common problem is the communication between the offline programming system and the robot itself. The following main set of functions are useful during the remote session:

- program download/upload
- remote start/stop, program delete
- determination of robot status (idle, loading, executing, etc.)
- read/write different variables
- create/delete variables
- performing event handling
- multi-access by multiple hosts

Most robot controllers provide only optional DNC connections via serial line and typically allow only a limited number of operations via that channel (e.g. download/upload). This serial connection has two main disadvantages:

1. It obviously means a point to point connection; so it is not possible to access or monitor the robot from multiple computers.
2. The speed of the communication is low.

Users wish to quickly connect robots to their network in a ‘client-server based’ manner. They prefer low-cost and vendor independent solutions, which are consequently standard based networks instead of proprietary ones of a single vendor. These connections mean very different remote interfaces of the robots.

As an alternative, the introduction of a standard based network solution such as MMS via Ethernet offers good capabilities and with the implementation of special Network Interface Units to the robots even the multi-access and the common interface problems can be solved.

3 Project’s Objectives

The target group are clearly S&M companies, which consequently cannot afford as high investment costs as large companies do. The costs of the required computer type, additional hardware and software has to be kept low in order to encourage the companies to invest in new technologies. Personal computers and standard software are important preconditions for this goal. Furthermore, the use of well known software and software tools for the realisation of a new system enables the user to work with the new system effectively after a very short time of practising. The user will accept the system faster and the productivity will raise with the user’s motivation.

Out of this motivation the AutoCAD system has been chosen as the development platform for the PROARC system. First it meets the precondition of a frequently used software in S&M companies and second it provides the required functionalities to develop a user friendly programming environment.

As a conclusion drawn from the analysis of existing offline-programming systems concerning the system’s capabilities following general requirements could be specified.

Modelling

- support of standard CAD interfaces (IGES, DXF);
- 3D solid modelling of the entire workcell, including robot, positioner, torchstation, and cleaning device;
- support of collision checking during simulation;

Programming

- support of Fillet and Butt welds;
- support of multiple layers;
- support of straight and circular weldlines as well as combination of those;
- automatic generation of welding paths including the appropriate welding data;
- unconstrained programming of welding sequences;
- effective access to welding entities;

Program generation and communication to the robot

- support of a common interface between the programming and program generation part of the system to provide the use of different robot languages;

- besides the DNC connection support of an MMS interface between the programming system and the robot in order to provide more control over the welding process for the user;

The specified functionalities roughly describe the scope and the capabilities of the system. However, the principle motivation for the development of a novel offline-programming system was the provision of an effective programming tool based on an affordable hard- and software platform.

4 System Overview

Based on the logical structure, the system can be divided into three distinct modules, which are the *Modelling Module*, the *Welding Planning Module*, and the *Program Generation/Communication Module*. Figure 1 shows the different modules and the robot cell.

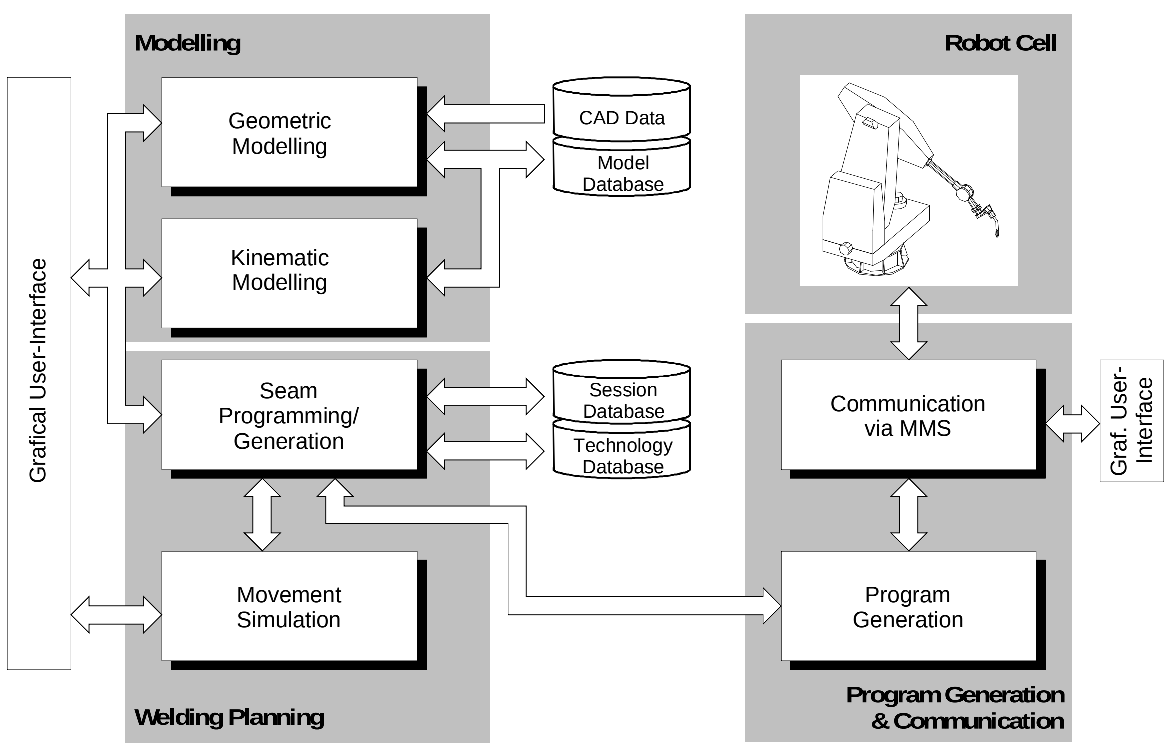


Figure 1: Structure of the PROARC System

The *Modelling Module* of the system allows the modelling of the entire workcell. The workcell consists of following items: robot, positioner, fixtures, and torches. In addition to this, the system provides models of cleaners and torchstations for auxiliary processes during a welding session (e.g. torch's cleaning/changing). Libraries of the most common types of these items will be included with the system along with the possibility to create user defined components[1].

In the *Welding Planning Module* the user is able to program seams on the 3D model of the workpiece interactively. In contrary to existing off-line programming systems for welding applications, the system not only provides the processing of single seams, but allows the user to access each layer of a seam individually. Furthermore, the user is able to define its own sequencing of the layers and thus uses his/her welding experience in this phase of programming.

In all states of the program it is possible to simulate the movements of the robot and the optional positioner. Thus, the user is able to jump back and forth between the programming and the visualisation phase. This gives a maximum of the welding process' controllability to the user. The programmed welding session will finally be translated into the robot language and downloaded to the robot. This is supported by the last, the *Program Generation/Communication Module*.

The interface to the Program Generation Module is defined in the, in STEP standardised „EXPRESS“ language. Since no predefined or standardised interfaces for the exchange of welding data exists, new Entities (exchange datatypes) were introduced.

In order to possess a maximum of control over the robot while working at the offline-programming system, an MMS interface has been implemented. In a first development step mainly the basic communication functionalities are supported. However, beside this MMS connection a common DNC connection has also been established. This additionally enables the basic communication to most robot controller types.

5 System Modules

In the following paragraphs the system's physical structure will be explained in more detail. Because of their close relationship the Modelling and Welding Planning Modules[2] are discussed together.

---

[1] All components of the system are modelled geometrically as solids.

[2] For clarity, it must be said, that these modules again have been split up in *submodules*. For the sake of simplicity, however, these *submodules* are continuously being called *modules*.

5.1 Modelling and Welding Planning

A modular approach has been enforced during system design and implementation to support incremental and iterative design, implementation and testing. Also, recent trends in software engineering show, that the software industry moves towards component based architectures such as COM (Common Object Model) from Microsoft [BRO] and CORBA (Common Object Request Broker Architecture) from Object Management Group [BEN]. Unfortunately, the PROARC project was not able to comply to these recommendations in its current scope. However, the modules could be adapted to COM with limited effort. This is also recommended, because COM based architectures are in the strategic plans of Autodesk (the producer of AutoCAD) as well as Microsoft (the producer of Windows, Windows 95 and Windows NT).

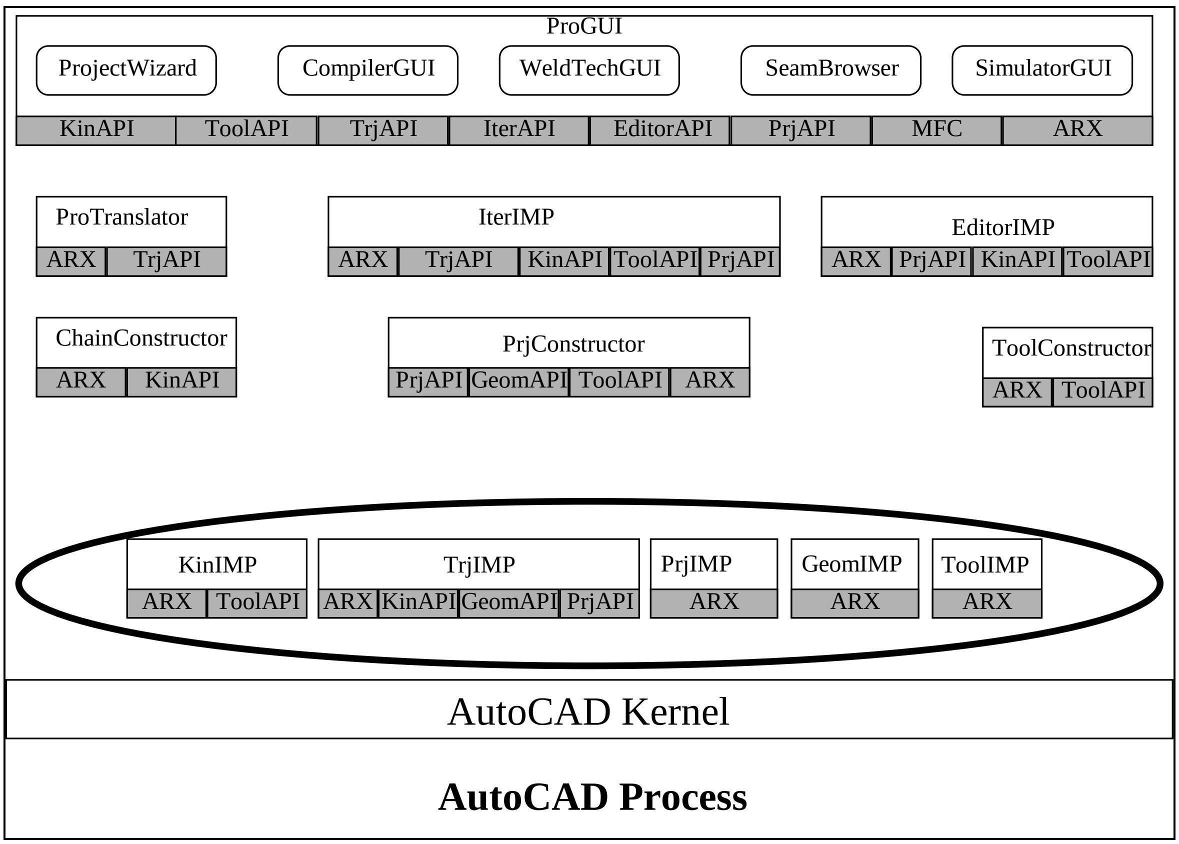


Figure 2: Modules for modelling and welding planning

The modules shown above in figure 2 correspond to a major set of objects in the domain model as well as function points of the Modelling and Welding Planning modules. In particular, modules encircled on figure 2 provide the models of the problem domain objects while the modules shown above them correspond to major functions of the system. It must be mentioned that a specific “control modelling” especially for the integration of different robot’s and positioner’s controller models is implemented within a specific module KinIMP.

In the following the modules’ structure and their tasks will shortly be described.

A module is a single dynamic link library (DLL), which can be loaded into the AutoCAD system at runtime. The PROARC system consists out of a number of such DLLs. Each DLL again is able to exist on its own or needs further DLLs, which export certain services.

In the PROARC system exist three ‘autonomous’ modules, located very close to the AutoCAD kernel.

PrjIMP

This module plays a fundamental role as it accommodates a central repository for the components of the workcell such as the robot, positioner, torchstation, etc. Components can be entered and checked out from this library through a well defined interface.

ToolIMP

The ToolIMP module implements the torchstation, cleaner, and torch models and their corresponding behaviour. Currently it supports *turning* torchstations and any torches and cleaners.

GeomIMP

The GeomIMP module provides geometrical representations for seam related entities. Seam related entities are seams, seamlayers, movements between seams or poses on the seam. The GeomIMP module is the interface between this abstraction of welding domain entities and the CAD system.

The next encircled module is the KinIMP module. In contrary to the previous three modules it uses services provided by an other module, the ToolIMP.

KinIMP

KinIMP contains the implementation of the robot and positioner. Because of the object-oriented interface (ARX), the commonalties among the robot and positioner model could efficiently be exploited. Currently the module supports a KUKA361, a KUKA160 robot, and any positioner.

TrjIMP

This module is responsible for the proper processing of seam related entities. Those entities are, on a higher level, seams, seam layers, inter seam movements, searches, and movements to the torchstation or cleaner and, on a lower level, poses, macros, and simple PTP-/CP-paths. The exported functionalities of the TrjIMP module are principally modifying, shifting, inserting, and deleting the entities mentioned above (with certain restrictions).

This finishes the description of the predominantly *domain specific* modules. The next group of modules mainly serve to provide certain system functionalities.

The three constructor-type modules enable to user to create cell components such as robots, tools, etc. These modules are interfaces to the above mentioned KinIMP, PrjIMP, and ToolIMP.

ChainConstructor

ChainConstructor is the module, that provides the necessary commands to the user enabling him/her to build the robot and positioner models. This robot modelling requires the modelling of the solids as well as the determination of the robot’s joints and links.

PrjConstructor

The PrjConstructor module serves as a basic user interface adapter for the services contained in PrjIMP. It implements commands driving the PrjIMP as well as additional commands such as mounting/unmounting torches.

ToolConstructor

ToolConstructor provides the commands to build the models supported by the ToolIMP. This, e.g. is the setting of the torches’ frames or associating a certain identification string to the torch.

After this „modelling“ related modules three modules - except for the graphical user interface module - remain providing miscellaneous functionalities.

EditorIMP

EditorIMP provides a possibility to directly modify automatically generated poses on the screen. Poses are displayed by a Cartesian co-ordinate system and can be altered interactively by using the mouse.

IterIMP

This module serves as an integrator as well as a simulator. The integrator takes the output from a seamprogramming session (seams, seamlayers, moves, etc.) and checks, e.g. if moves to the torchstation have to be included or if all seams are in a horizontal position (the system is restricted to those types of seams). After a successful integration, the simulator interpolates the robot movements and displays the results on the screen. It also provides mechanisms indicating during the interpolation occurred errors.

ProTranslator

Finally, the ProTranslator module performs the generation of the STEP interface file. Files can be generated for arbitrary sequences of welded paths.

As a last module the user interface module ProGUI remains.

ProGUI

The ProGUI module encapsulates the graphical user interfaces to the above mentioned modules.

Summing up it may be said, that special emphasis has been given to integration activities. Due to the chosen, modular approach, the integration can be done by simply loading the concurrently developed modules at once into the AutoCAD process. This kind of binary interoperability is achieved by well designed interfaces, that are bound at run time.

5.2 Program Generation and Communication

The modules of Program Generation, figure 3, in contrary to the in AutoCAD used modules above, are statically linked together and build a single executable (EXE) file. In the following the modules' structure and their tasks will shortly be described.

PGen

This part handles the user interface of the Program Generation. The frame of this module and the three related ones were generated using the Application Studio tool of MFC (Microsoft Foundation Classes). About 10 different Dialogue Boxes and one Menu were defined within the tool. The default Dialogues are used for file opening and saving and some others allow to set different values e.g. to (re)edit the task file data, or change search options. As an extra tool with the simple 'view' technology of MFC it is possible to take look at any Step or robot program file.

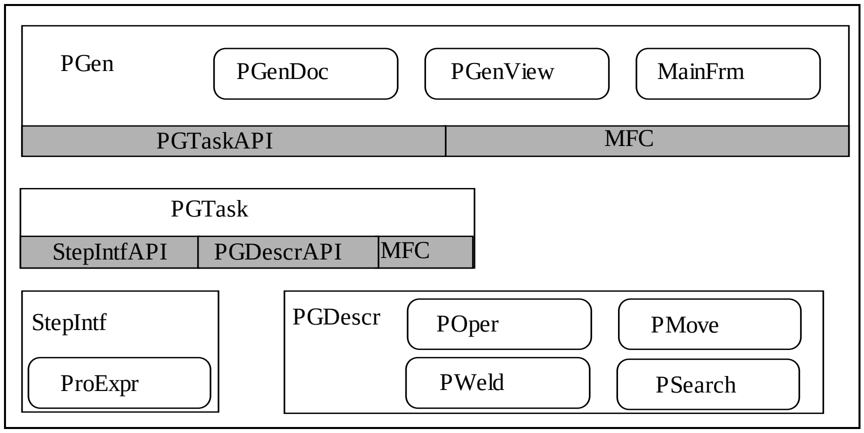


Figure 3: Modules for program generation

PGTask

The PGTask module manages all the operations belonging to the STEP, the robot program, and the task files and contains three general classes belonging to each file category. So the robot program will be built up from a set of these operations in the order given by the STEP file. It possesses general file management operations as well as specific ones dedicated to the given classes (e.g. load non data parts of the STEP file, update the task file with the new search options). Additionally, the module contains a class, which describes the features of the actual robot (e.g. existence of the sensors or I/O channels belonging to the different welding features).

StepIntf

This module translates the entity based STEP description into an internal representation of the program with the checking of different errors. It has a special structure which is independent from specific STEP specification, and thus could be used as a pre-processor of any STEP based file. The PROARC specific definition is given in the ‘proexpr’ part of the module.

PGDescr

The operations classes of these modules are responsible for the program generation. They are defined in a certain way to build a class structure that is independent from the used robot type or robot language. However, the contents of those is initialised especially for a specific environment. The classes of the module are static and are working with dynamically generated data. They are used to define how the different elements of the robot program should be generated. They were built up to give a robot-type independent structure, where only the contents of the classes need to be rewritten for other robot programs. Within this module, the *Pmove* is responsible for the elementary movements, the *Pweld* generates the welding specific parts, the *Psearch* creates search subroutines within the robot code, and *POper* has different routines belonging to each operations defined in the STEP specification. There are additional parts within the PGDescr generating the frame of the robot program (header and footer things).

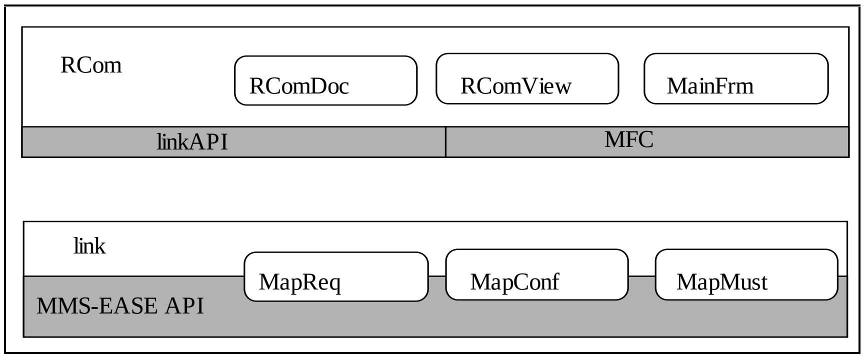


Figure 4: Modules for the communication

RCom

The (sub)modules of the Communication Module, figure 4, are also linked together to have a single executable (EXE) file. The software structure is simple: the user interface and file management parts (Rcom) are connected via a general linkAPI to the MMS-EASE parts of the program. The general structure of Rcom is very similar to the Pgen of the Program Generation, of course the functionality is different.

Link

The linkAPI was developed in an earlier project in CAI and provides a general interface for remote control of different robots. In the link part, the *MapReq* collects the network request calls, the *MapConf* the confirmations and the *MapMust* contains the further necessary elements required by the MMS-EASE API (e.g. initialisation, error handling).

6 System Functionalities

In contrary to the software related view of the proceeding chapter, *System Functionalities* describe the capabilities of the system from the users point of view.

6.1 Modelling Module

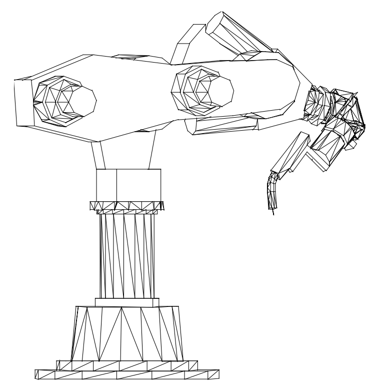
To perform offline robot programming, the users must be able to build a model of their manufacturing workcell. The PROARC system enables the user to use the following workcell components:

- robot
- torch
- positioner
- cleaner
- torchstation
- obstacles

Once building a welding environment the users can save and load it, thus ensuring that the cell once laid out can be reused from time to time inserting different workpieces as required.

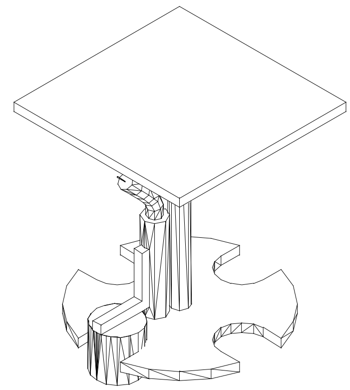
AutoCAD has been extended with a number of commands in a consistent manner to enable users to model their own custom components. In the following sections this creation process is discussed briefly to give a reader an insight of what can be and cannot be done with PROARC's workcell component modelling.

Building custom component in PROARC is always done in two steps. First the user must create a solid model of the component using AutoCAD means. Second, the user must use PROARC-defined commands to add attributes to the solid model and convert it into a PROARC custom component.



The most complex of the models is the robot model that is still quite simple. PROARC uses a kinematics model of the robot and positioner. However, in case of robots the inverse kinematics algorithm has to be provided for each kind of robot. Currently, it is possible to model KUKA 361 and KUKA 160 robots only, for which the PROARC system includes the inverse kinematics. The special attributes to be added here are the joint axis, joint limits, base frame and tool frame of the robot and positioner. PROARC supports a command 'kincreate' to let the user define these properties. The system also provides commands to test the created kinematics chains in terms of forward kinematics and inverse kinematics.

The cleaner and torch models are fairly simple concerning their additional attributes: they require two frames, although the meaning of these co-ordinate frames are different in the two models. In case of the torch the two frames represent the base and the tip of the tool frame of the torch. In case of the cleaner one of the frames represent the so-called effector frame where the torch cleaning happens, while the second one models the so-called approach frame. This letter one needs some more explanation. When the robot approaches the cleaning station it does this in two phases. First it moves to the approach frame using configuration space interpolation (PTP movement) and this is followed by a path interpolated in Cartesian space (linear move) to the effector frame. The same process is executed in reverse order after the cleaning has happened.



The torchstation model is a bit more complex. Basically it is a collection of so-called slots. Each slot is represented by a co-ordinate frame. The system currently supports turning torchstations. They also feature an axis, an effector, and an approach frame. The effector and approach frames play the same role as in case of the cleaner. Thus the model works like this: The robot must move to the effector frame of the torchstation (after checking that an empty slot is being aligned with it) unmount its torch, mount on the torchstation and request a new one from the torchstation. Then the torchstation is turned around its axis to align the slot of the new torch with the effector frame. At this time the torch can be unmounted from the torchstation and mounted to the robot. Then the robot can move on. Again the PROARC system supports a command 'toolcreate' to build these models.

The obstacle models are quite simple as they are plain solids.

Since the PROARC system builds on AutoCAD all the capabilities of AutoCAD can be used during manipulation of PROARC models. Also, building on standards (AutoCAD represents the de-facto standard in PC-based CAD) brings all the ease of learning and interoperability that users at S&M companies demand.

6.2 Welding Planning Module

The welding planning module is designed in such a way, that it takes as much action from the user and leaves those tasks, which require the welding expert's knowledge. Therefore, the entire generation of the robot's movements on the seams is automatically generated.

Macro Definition for Seams

The orientation of the torch basically is defined by the seam type and the connections of different seams. Consequently, the system has been designed in a way, that it is able to generate orientations on a seam automatically. This is realised by splitting up the seam into distinct parts, so called macros, for which a specific start- and endorientation as well as a set of technology data is valid. Those standard macros are used throughout the entire seam generation process and provide the default setting for possible refinements.

Figure 5 demonstrates the principles of the macro generation process. Initially, two seams have been specified by picking the corresponding edges on the workpiece. The system takes those seams as an input, detects the connection and the angle between the seams and splits the seams up by choosing the appropriate macros. The arrows approximately show the torch's orientation on the seam.

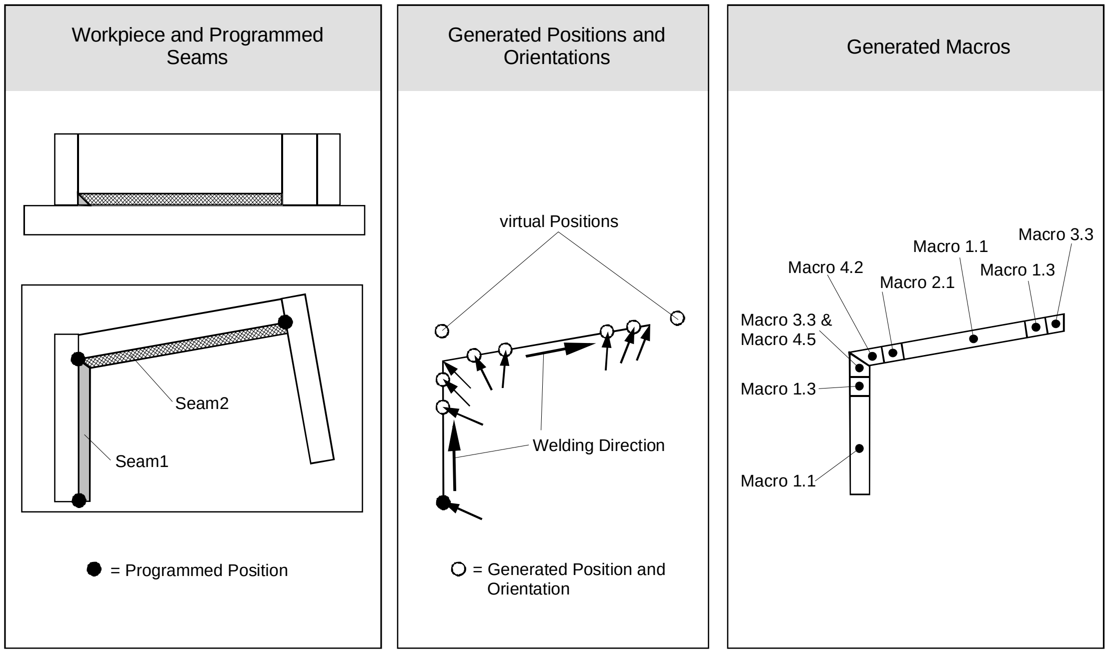


Figure 5: Macro generation for two connected linear seams

Search Sequences

The offline-programming session defines the entire sequence of movements for the welding of a workpiece. However, there will be deviations between the generated positions and the real positions e.g. due to calibration errors or inaccuracies of the real workpiece. Therefore a set of search movements has been specified, which determine the real position of the workpiece by using a tactile sensor. These movements are defined for different workpiece geometries.

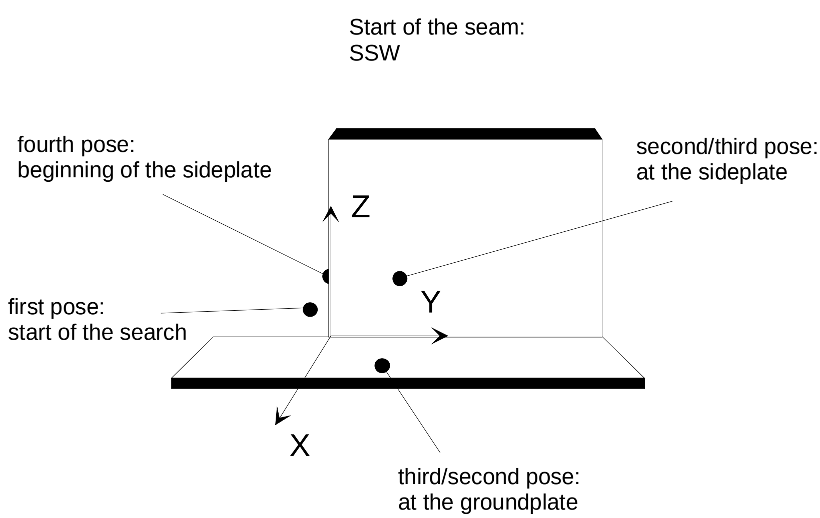


Figure 6: Search sequence for the search type „Search Side Wall“ (SSW)

Figure 6 shows the searchpoints generated for the search of a vertical plate. The first point is the approach point, on which the search starts. A vertical and a horizontal movement determine the error in the X- and Z-direction. Finally, a movement to the front of the plate specifies the remaining error in the Y-direction.

Additionally, searches are supported for horizontal plates, corners, and open geometries at the beginning as well as corners and open geometries at the ending of a seam.

Flexible changing of welding sequences

Welding sequences consist of movements on the seams and the layers as well as movements in space. Usually the programmer, once programmed a welding session, is not able to change the order of the sequence without distroying it. This had been changed in the PROARC system. Here the user is able to shift seamlayers to different positions in the welding sequence depending on his/her preferences and experience. The connectivity of the shifted seamlayers is checked automatically.

Figure 7 gives an example of the different types of movements and the shifting of the seamlayers via drag&drop. The system automatically restructures the movements at the locations, where the seamlayer has been taken from and where the seamlayer is shifted to.

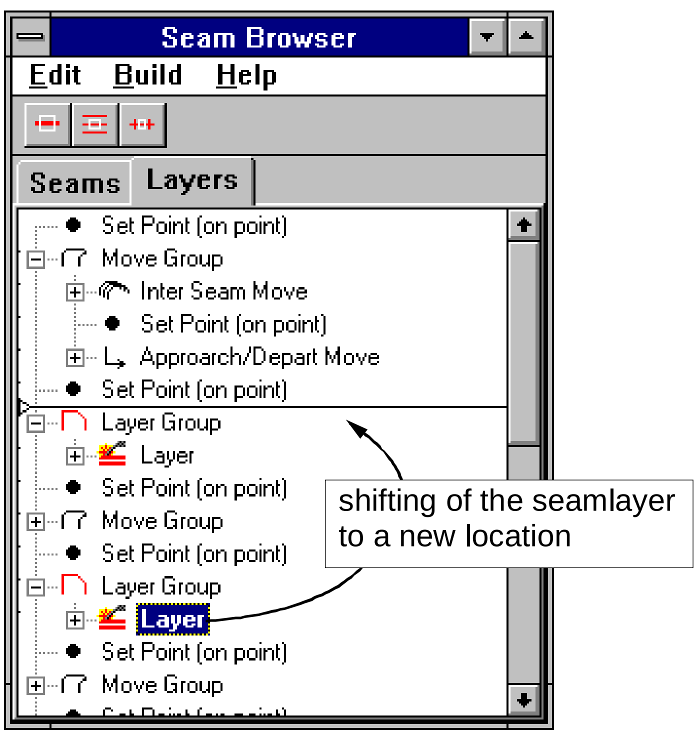


Figure 7: Movements on the seam within a welding session can freely be shifted

Dissolution of rigid sequential programming

Most systems follow a top down approach for the programming of welding sessions. PROARC enables the user to apply different modules (seam programming, simulation, program generation, etc.) at any time in a programming session to any part of a welding sequence. For example, it is possible to generate a robot program for multiple seams or for a single seam out of the entire set of movements. This enables individual checking and - if necessary - reprogramming of problematical program parts.

Figure 8 demonstrates the difference between the common top/down and the additionally in the PROARC system provided way of programming.

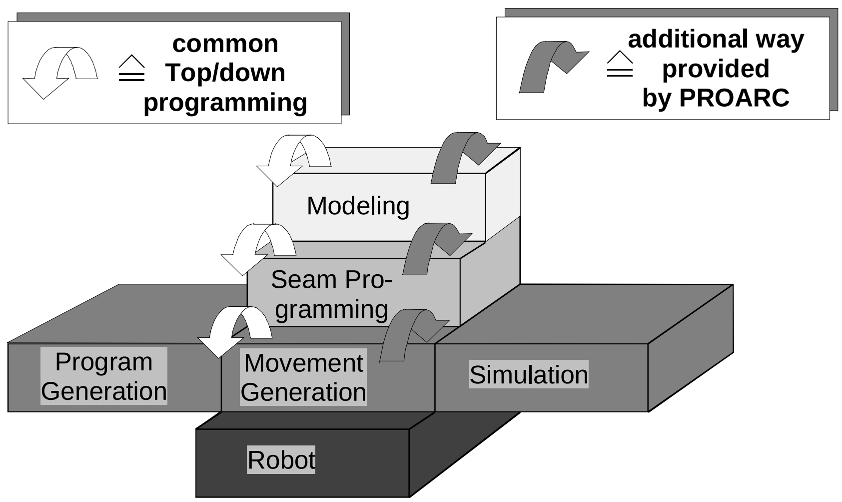


Figure 8: Dissolution of rigid sequential programming

Technology database

In order to enable the system to automatically associate the appropriate technology data to the seam's macros, a corresponding datasource has to be provided. This datasource should be filled with data gained in a number of tests performed by welding experts. The experts' experience and knowledge as well as the mapping of this information into a datafile mainly influences the quality of the welded product.

However, for the welding expert it should not be necessary to know how to operate an offline programming system. It is sufficient for him/her to enter the data into spreadsheets or word processing documents. This decouples the initial phase of data determination entirely from the programming phase and does not require the overhead of the complete programming system.

For the realisation of this approach, the Open Database Connectivity (ODBC) interface has been used, figure 9. This interface allows applications to access data in database management systems (DBMS) such as EXCEL, ORACLE, ACCESS and others in a uniform way. Thus, the PROARC system can access datafiles created in different DBMSs with the same user interface just by loading the corresponding DBMS's driver.

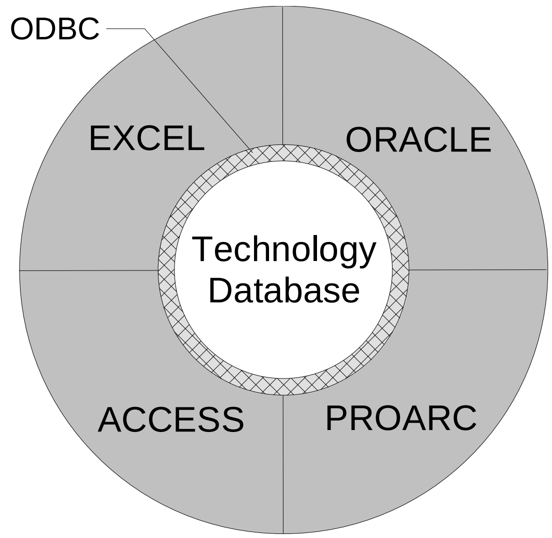


Figure 9: Access to technology data by different Database Management Systems and PROARC

Simulation and display of critical configurations of the robot

The simulation of the robot's movements is able to display the entire robot or only the torch if necessary. During the simulation a check for collision can be performed with selected solids of the robot (torch, n links).

Critical Positions of the Robot are displayed on the effected part of the seam. It is often difficult to determine the reason why a robot is not able to reach certain positions and orientations. Singularities, movements out of the workspace or out of the valid axes' values are possible problems. Knowledge of the problem type enables the user to immediately take appropriate action.

Summarised, it can be said, that the listed functionalities are those functions, which discern the PROARC system from other commercially available systems. Basic features of the system such as libraries of different robots, workcells, and workpieces, or simple calibration functionalities are not explicitly mentioned in more detail. They are common to most simulation systems and are basic preconditions for the efficient offline-programming session.

6.3 Program Generation and Communication

The two modules help to transform the results of the offline programming session into real movements. The Program Generation Module generates an executable robot program from the PROARC's (STEP based) internal format. The Communication Module helps the user to work with the welding robot directly from the offline PC (download, start etc. robot programs).

Both modules are independently executable programs. However, it can freely be switched from one to the other.

Presently the modules are running under MS Windows 3.1. Additionally, the Program Generation Module is executable under Windows95 and WindowsNT. The Windows95 version of the Communication Module is under testing. However, it is certain, that it will be executable under Windows95 as soon as the internal used MMS Application Program Interface (API) (the SISCO's MMS-EASE Windows95 version) will be commercially available (3Q96). From that time on all the modules of PROARC system can be run on the latest, low-cost operating system.

Program Generation Module

The main function of the program generation module is the compilation of the STEP based robot movement description including the welding features to the actual arc-welding robot language with a format, which can be directly download into the robot controller. This means, that the module has strong dependencies to the actual robot environment and has to be modified, if applied to a robot controller. However, the internal structure of the module supports this programming effort.

```
# PROARC program generator task file
# automatic generated
####################################
# STEP
STEP file=C:\PROARC\PROGGEN\CHANGE.STP
# ROBOT PROGRAM
Generated robot program=C:\PROARC\PROGGEN\MP61.IRD
Uploaded robot program=C:\PROARC\PROGGEN\UPLOAD1.RP
Inner program name=HP61
# OPTIONS
Welding off=0
Search off=0
Skip seam group off=1
Search finement=9
# TASK DESCRIPTION
Description=auto generated: April 04, 1996
### End of file ###
```

Figure 10: A sample task file

All the parameters of the program generation are stored in a so called *task* file, figure 10. The user is able to generate it automatically and all the modifications are written directly to the file. Thus, there is no need to save it during the work. The task file contains not only the input and the output of the program generation (STEP file and robot program), but also special switches (welding on, search on e.g.).

Different types of robot programs can be generated from the same source STEP file. The 'welding off' switch allows to check the exact path of the robot during the real movement without the welding itself. Three types of searches are defined. Programs can be generated skipping searches ('search off'), with checking only the correct positions ('skip seam group on') and using the grouping of the search and welding moves the robot can change its path according to the result of the searches ('skip seam group off'). It is possible to set also how fine the search has to be performed.

If any errors happens during the robot program generation the user receives a note, and the generations stops. Three types of errors can occur:

- Errors in file management (e.g. Unable to open the file)
- Error during the processing of the STEP file (e.g. STEP 'entity' not found), in that case the user is informed which STEP line has problems.
- The data of the STEP file contains any bad information (e.g. The given speed value is out of the robot range or This interpolation is not allowed here).

The user can take a look to the files within the program, but the direct editing of them (either the STEP or the robot program files) is not allowed. It was decided because of safety reasons, but it can be changed easily if it is requested.

Communication Module

The Communication Module in the present version of the PROARC system is using the MMS over Ethernet industrial network to exchange data with the robot controller, which includes a PC-based Network Interface Unit (NIU) to connect to the network.

The connection can be established with an initiation message and closed with a conclude message. Both communication partners, the PROARC station (Communication module) and the robot (more precise its NIU) can skip the connection with an abort message.

Some stati actions were built in to allow the user to get continuously information from the robot. It was realised, because from the networking point of view the robot is not active. So all the information is coming from it via message polling. It is important, because without it, the Communication Module will not know, if the robot program ends, if manual torch cleaning, or changing is necessary or not.

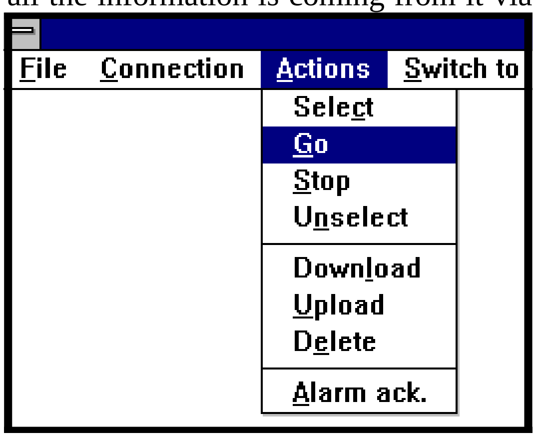


Figure 11: The **Actions** *menu in the Communication Module*

The menu list in figure 11 shows the actions the user can activate to work with the robot. The first group of actions supports the direct remote control operations (a set of the MMS Program Invocation services), the second group manages the robot programs within the controller (MMS Domain Management services). Additionally, there is a possibility (via an MMS Variable called R_ALARM) to acknowledge the alarm messages of the robot controller (e.g. Transmission Error or No workpiece found).

Some actions ask internal robot program names (e.g. Select, Delete), while others are using the Windows standard File Open Dialogues (Download, Upload).

If the user wants to work here also with the task files created in the Program Generation module, he/she directly gets the download file from the task file data. If an earlier uploaded version of the same robot program exists, a warning message forces the user to choose which one to download. Similar warnings will be shown, if the user wants to upload the same robot program a second time. At one time only one open task file is allowed.

7 Application in Real Environments

The PROARC system has been applied in two different environments with two different robot types.

During the final workshop in Budapest a KUKA IR 361/8.0 with an RCM3 Siemens controller was used. The workcell consisted of the robot, a welding table, and a tactile sensor. A workpiece with linear and circular geometries had been prepared.

As a first step, to adept the modelled robot to the real robot, the zero positions and the direction of the moving axes had to be determined. Since the RCM3 controller is only capable to process Cartesian co-ordinates instead of joint angles as an external position and orientation definition, this specific definition has to be adapted as well. Finally, the torch and the correct TCP and TOV values had to be measured and set in the PROARC system and the controller. A simple calibration by measuring three distinct points on the workpiece were necessary to move the modelled workpiece to the correct location in the modelled workcell.

After this preparing actions, sample programming sessions were performed. Those sessions included:

- movement generation for connected linear and circular seams
- generation of search sequences by using the tactile sensor
- generation of multiple seamlayers

It had been shown, that the generation of executable programs could be performed very quickly. In rare cases the simulated movements differed from the real movements due to the only kinematical modelling of the robot. In those cases, however, adaptations could directly be performed in the offline system and within the range of minutes the new program was generated.

The second environment in which the capabilities of the PROARC system could be demonstrated was a robotics cell at the WZL in Aachen. In this cell a KUKA IR 160 was applied again with a RCM3 Siemens controller. The adaptation between the PROARC system and the real environment was performed within two days and the results of the generated programs were highly satisfying. A more complex workpiece shown in figure 12 had been chosen for demonstration. Six seams were programmed on six distributed locations on the workpiece. Except for the last seam, where the robot moved in a critical configuration of the 4th and 6th axis, the program could be executed directly on the controller. With refinement steps the entire process to generate the final executable program lasted approximately four hours.

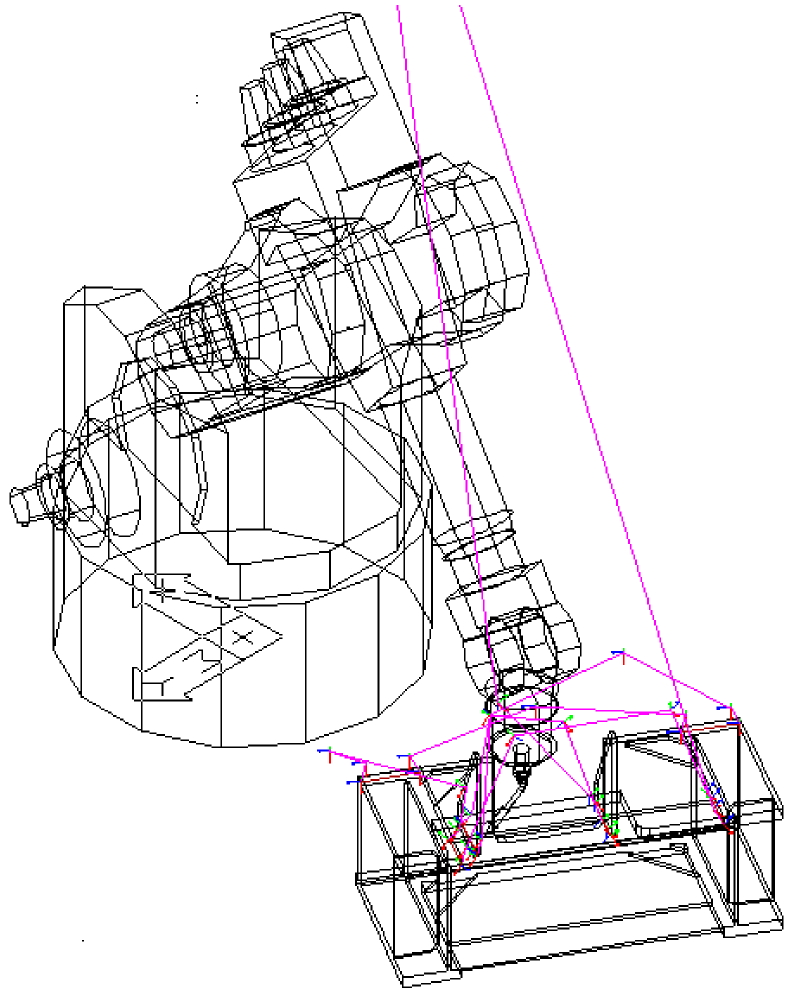


Figure 12: Workcell with the KUKA IR 160 at the WZL, Aachen

These two examples for the realistic applicability of the PROARC system demonstrate both, the efficiency and the flexibility of the system. It could be proven, that without major effort the initial adaptation of the system to the real workcell and the generation of executable robot programs could be performed very quickly.

8 Discussion of Results

The most important achievement of the Workcell Component Modelling and Welding Entity Management Modules is the extension of AutoCAD functionalities in a consistent manner. Here the emphasis should be on the 'consistent manner' since this makes a shortened learning curve and interoperability with other software possible. This kind of tight integration could be achieved using state-of-the-art development environment (ARX) that also makes the system worthwhile for further investment of development efforts. Interoperability manifests itself in two different ways. First, naturally all other third party applications designed for AutoCAD can be used in conjunction with PROARC (There are more than 4000). Second, since AutoCAD solid modelling is based on the ACIS technology (that is a de-facto standard in solid modelling), the exchange of models with other ACIS based applications can be done with ease. (This can be important when the S&M company is a part supplier of a bigger corporation and the models should be exchanged electronically.)

Furthermore, AutoCAD has been found an ideal platform for cost effective CAD-based manufacturing application such as PROARC because of its wide acceptance at the targeted companies.

On this powerful development environment a number of functionalities have been implemented, which considerably facilitate the programming of a welding task. The chosen approach is simple: similar to shopfloor oriented programming systems the user should only need to perform those actions, which necessarily require his/her knowledge or experience.

This is realised by several achievements:

- The user simply "clicks" on the edge of the workpiece, which has to be welded.
- Only this seamdata has to be given to the system, which is necessary to specify a seam.
- Movements on the seam including optimal orientation of the torch are generated automatically.
- Technology data for distinct seamtypes, workpiece material, torch orientations can by determined by a welding expert decoupled from the programming system.
- Problematical movements of the robot are immediately displayed and specified.
- Direct access to the different welding entities is possible by the use of a structured view to those entities.

These features relieve the programmer from the most often time consuming specification of robot configurations on each single path location.

Basic to these features is the underlying datastructure of the welding entities. This structure has been designed in a way, that single movements or movement sequences can easily be modified, shifted, or deleted. In contrary to existing systems does this consequently result in a more flexible way of programming. Furthermore, allows this structure an easier extension of the systems functionalities. Thus, specific needs of the user can efficiently be implemented in the system and tailored for individual applications.

The result of a welding session is stored in a STEP based interface format. Welding specific Entities have been defined in EXPRESS in order to cover a wide range of programming languages on a higher level of abstraction. Consequently, the programming of the welding task and its simulation is entirely decoupled from the applied robot language.

The Program Generation Module has been designed in a way, that it as well easily can be adapted to further robot programming languages. Optional settings for the program generation such as for skipping search movements or disabling the actual welding are useful for an initial test run of the offline generated program.

Communication via MMS or conventional DNC connection enable to directly influence the action on the robot controller. Beside basic remote functionalities such as up- and download of programs, advanced operations such as checking of the robot's status are supported. Finally, the use of a task file optimally allows the programmer to manage those files which are related to one single programming session (commonly known as project files).

Summarised, the PROARC system with its novel features, embedded into the well-known, PC-based AutoCAD system is an attractive alternative to existing offline-programming systems. Beside its relatively low costs the intuitive user interface and the underlying functionalities promise to increase user acceptance for this new tool and consequently increases productivity of the programming process.

9 References

[BEN] Ben-Natan, R.: *CORBA: a guide to common object request broker architecture*, McGraw-Hill, 1995

[BRO] Brockschmidt, K.: *Inside OLE2*, Microsoft Press, 1995