Intelligent, Open Architecture Controller using Knowledge Server

János Nacsa, George L. Kovács, Géza Haidegger
{nacsa|gkovacs|haidegger}@sztaki.hu

SZTAKI (Computer and Automation Research Institute)
www.sztaki.hu, Budapest, Hungary

ABSTRACT

In an ideal scenario of intelligent machine tools [22] the human mechanist was almost replaced by the controller. During the last decade many efforts have been made to get closer to this ideal scenario, but the way of information processing within the CNC did not change too much. The paper summarizes the requirements of an intelligent CNC evaluating the different research efforts done in this field using different artificial intelligence (AI) methods. The need for open CNC architecture was emerging at many places around the world. The second part of the paper introduces and shortly compares these efforts.

In the third part a low cost concept for intelligent and open systems named Knowledge Server for Controllers (KSC) is introduced. It allows more devices to solve their intelligent processing needs using the same server that is capable to process intelligent data. In the final part the KSC concept is used in an open CNC environment to build up some elements of an intelligent CNC. The preliminary results of the implementation are also introduced.

KEYWORDS

Intelligent CNC, Open Controller, Knowledge Server

1. INTRODUCTION

There are many definitions of the intelligent machine tool in the literature. In the late 80's Wright and Bourne [22] said that „*We must therefore acknowledge that the degree of intelligence can be gauged by the complexity of the input and/or the difficulty of ad hoc in-process problems that get solved during a successful operation. Our unattached, fully matured intelligent machine tool will be able to manufacture accurate aerospace components and get a good part right the first time*". They told that an intelligent machine tool had the CAD data, the materials and the set-up plans as inputs and could produce correctly machined parts with quality control data as outputs. It is clear that AI techniques are necessary to apply if one wants intelligent NC machine, but - of course - the usage of them is not adequate in intelligent behavior. Table 1 summaries the features of an intelligent CNC as Wright and Bourne collected them and shows two further things: the positive changes done in the recent years and the still existing gaps where - according to the scientific community - AI offers solutions with its information processing methods.

Analyzing the Table 1., it is clear that many features do not require direct AI methods. We can state that the main reasons of the advancement were:

- The development of the hardware elements (more sensitive sensors, more precise actuators, quicker and stronger computers etc.) even in higher requirements.
- The development of the software and the methodology mainly in the preparation phases of the manufacturing (in design, planning, scheduling, resource management etc.) and in the user interface issues (more comfortable and informative 'windows-like' screens and menus).

Features (forecasted in 1988)	Big advance by 2001	AI methods still needed
Reduce the number of scrap parts following initial setup.		✓
Increase the accuracy with which parts are made.	✓✓	✓
Increase the predictability of machine tool operations.	✓	✓
Reduce the manned operations in the machine tool environment.		✓✓
Reduce the skill level required for machine setup and operations.		✓✓
Reduce total costs for part fabrication.	✓	✓
Reduce machine downtime.	✓	
Increase machine throughput.	✓✓	✓
Increase the range of materials that can be both setup and machined.	✓	
Increase the range of possible geometries for the part		✓
Reduce tooling through better operation planning	✓	
Reduce number of operations required for setup	✓	
Reduce setup time by designing parts for ease of setup	✓✓	
Reduce the time between part design and fabrication	✓✓	
Increase the quantity of information between the machine control and part design operations	✓✓	
Increase the quantity of information between the machine operations and the machine control		✓
Increase the quantity of information between the human and the machine control	✓✓	✓

Table 1. Commercial needs for the intelligent machine tools

Even if there is a big advance in the technology of the CNCs, the knowledge processing and other AI methods have not appeared within the controllers, so in some points there is no real development. Special heuristic rules, problem-solving strategies, learning capabilities and knowledge communication features are still missing from the recent controllers available on the market. It is also true for many new, open or PC-based CNCs, where DSP add-on-boards provide the necessary computation power and speed.

Further requirements of intelligent CNCs can be found out and defined (Table 2.) based on other papers [12, 13, 20] and different discussions. The second column indicates whether the different AI techniques (mainly rule based systems, neural nets and fuzzy logic) would provide methods and solutions. One can find positive answers to all these issues in the recent literature:

Further features	AI would help
Model based on-line path generation	(✓)
Automatic tool selection	✓
Technological based settings of the operational parameters	(✓)
Automatic compensation of machine limits	✓
Automatic back-step strategies	✓
Detection and compensation of geometrical deflection	✓✓
On-line selection of control algorithms	(✓)
Intelligent co-operation with other devices to solve problems together	✓
Detection and correction of tool wear and breakage	✓✓
Automatic handling of rejected workpieces	✓✓
Detection and management of emerging situations of the machine tool	✓✓
Complex self-diagnostics	✓✓

Table 2. Future requirements of an intelligent CNC

The list may be continued with the learning capabilities and others. In the users' point of view these features rough in a controller, that "recognizes the problem" and "efficiently and reasonable solves them" with minimal disturbance of the environment of the controller.

2. RESEARCH RESULTS IN INTELLIGENT CNCs

On the one hand in the recent literature one can find many different topics related to intelligent CNCs. Unfortunately they often do not mean intelligent behavior but the application of intelligent methods. Sometimes it is the case that authors call their devices "intelligent" if one module of the system contains any AI based method. On the other hand the key controller vendors leave everything to the users or machine tool builders offering PC/Windows based CNCs. With these systems any software modules (e.g. even AI based ones) can be coupled into the controller but they do not offer real solutions or methodologies, but only software possibilities. Both facts are far from the user wishes stated in the previous chapter. The following list summarizes the most important active research topics in this field. A real intelligent CNC would contain most of these issues.

- Fuzzy logic based concurrent control of some operating parameters (e.g. cutting speed, depth of cut, feed rate) independently from the given tool and the workpiece.
- Neural nets and fuzzy rules in the CNC's control algorithms.
- Optimal path planning, real-time correction of the trajectory.
- Compensation of temperature (and other) deformations.
- Life time management of the tools and other parts of the machine tool including self-diagnostics.
- Tool breakage detection (maybe forecasting) and tool wear monitoring (maybe compensation) with AI methods.
- The utilization of CNC management (setup, orders, etc.) via intelligent agents.

Monostori [12] classifies the possible intelligent parts of a CNC into three groups, namely: (1) tool monitoring, (2) operation/machine tool modeling and (3) adaptive control. A general problem in all the three groups is, that the AI based solutions are typically limited and valid only in a very narrow field. If one changes some parameters of the operation or the environment, the earlier successful methods may become poor.

There are some research efforts where more than one intelligent modules are built into the controller. So Cheng at al. [4] developed a PC based controller where some DSP cards serve intelligent functions (e.g. adaptive control of cutting force using fuzzy logic; knowledge based self-diagnostic and error recovery/management; multisensor based neuro-fuzzy tool monitoring).

A special type of adaptivity partly helps on this hard and well-known problem. If it is possible to replace the different modules of the controller time by time, than one can guarantee, that a given AI module can run within its limitation, and over it another module (e.g. a much simpler one) covers the same functionality. It can be realized (among others) if the controller is open to allow this replacement.

3. DIFFERENT OPEN ARCHITECTURE CONTROLLERS

Manufacturing has constantly been a technological domain, in which the industry was driven to apply the current high-tech in the computer and control area. It is no surprise, that the Open Systems concept has also diffused into the manufacturing area, and factory managers are often referring to the open manufacturing systems. The terms and definitions are far less exact than the terms applied in the operating systems environment, but by now, the change of the global manufacturing paradigms (e.g. see Kovács [10]) are directing our focus on the key user aspects of openness.

The booming market for industrial automation led to a vendor dominating situation: dozens, or rather hundreds of controller manufacturers (vendors) are developing, implementing and installing different solutions for automation tasks. But several problems are arising because of this situation:

- there are a large number of incompatible products,
- the controllers can not cope with the frequently needed updates,
- the service, maintenance and repair costs are scoring high,
- professional personnel to work with controllers is decreasing in number.

Even if many classifications are possible, Fig. 1 shows a widely accepted one - introduced by OSACA - three different ways of openness of a controller architecture.

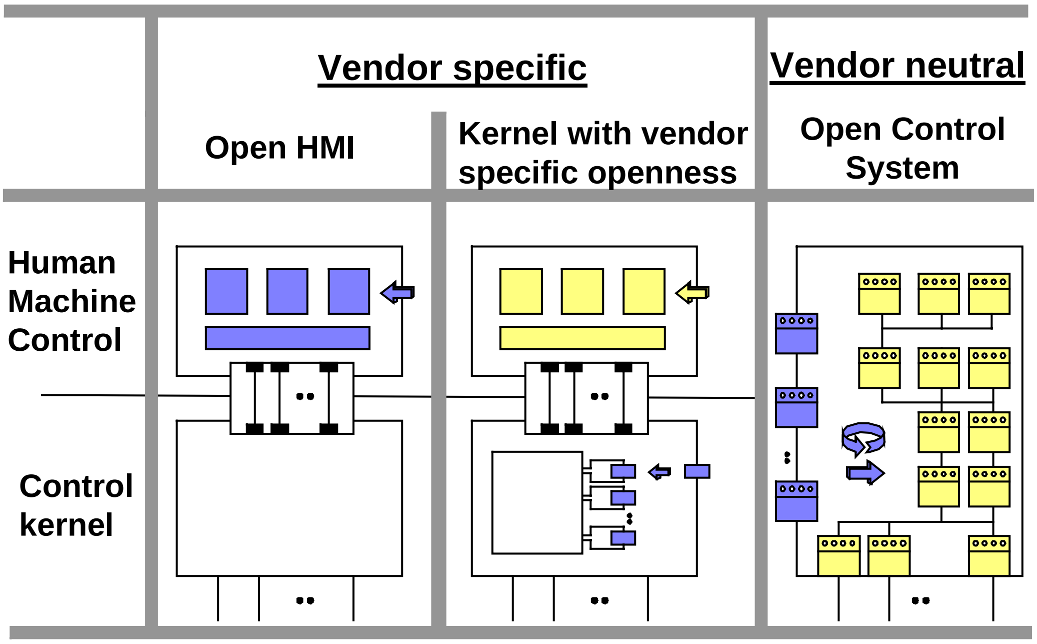


Figure 1: Different types of architectures of open controllers [1]

The first architecture means an open human machine interface (HMI), that is a must nowadays. Most of the vendors (e.g. GE Fanuc and Siemens) offer these features using a PC/Windows based environment coupled to the NC kernel which solves the motion control tasks in real-time.

In the second type of open architecture the vendor offers interface also to the NC kernel or -at least - to insert user specific filters (e.g. Sinumerik 840C can be set-up in this way with special agreement from Siemens). Many smaller vendors have PC based 'open' solutions where an add-on DSP card makes the motion control. Others run a real time operating system parallel to the MS Windows on the same single processor. In both cases users have an API to build their applications based on the vendor specific open platform.

The third type of open architecture is vendor neutral. In this case a joint consortium defines all the interfaces of the different controller modules including the real time parts (e.g. motion control, axis control, PLC functions). The three most important vendor neutral initiatives are shortly introduced in the following.

The need for a new and open CNC architecture was emerging at many places around the world. One of the most important work was done from 1992 within the frames of the European project named OSACA (Open System Architecture for Control Applications) [18, 1]. The main results of the project are: an analysis of the state-of-the-art and future requirements of NC controllers, a reference architecture, a general and platform independent API for inner CNC and outside communication and a configuration system supporting the possible machine tool vendors.

Similar efforts are going on in Japan within the IROFA Consortium [17]. In the OSEC they defined a 7 layer reference model (similar to the OSI network layers). They have paid attention to the programming interface between the different layers. They call it Message Coordination Field and defined different C functions to each layers. It is sensible that only a limited number of the functions are mandatory in an OSEC controller.

In the US in 1994 the three big American car manufacturers published a white paper [5] about their requirements of the future open and modular controllers. OMAC has not defined a fix reference architecture, but a set of modules [16] to build up with them different types of controllers . They are specified in IDL (Interface Definition Language), and many of them have subparts. The detailed specification of the modules is getting critical, e.g. the Axis module has more than 10 subparts with more then 400 methods

In all these projects the module structures and hierarchies (reference architecture) of the NC controllers and the APIs of any defined modules were published. Unfortunately even if the aims of these efforts are nearly the same the resulted controllers are incompatible in many senses [15]. But because of the open modules and the precisely defined APIs, any of these efforts could be a very good starting point of building intelligent CNCs. It is possible to implement any modules of the reference architecture using AI methods or to add further advanced modules to provide extra features for the controller.

4. KNOWLEDGE SERVER APPROACHES

The features of World Wide Web led Eriksson [8] to introduce knowledge server to easier solve the installation and version control problems, distributed and remote access issues of expert systems and to provide a web based interface of the knowledge base for the different users.

Some advanced knowledge based systems are based on this concept. So the Cyc system, the most important research on the common sense, is organised as a knowledge server [11]. Also the Istar knowledge server [2] provides on-line advises in many different topics (e.g. stock exchange, Internet security). There are also some applications of knowledge servers in manufacturing (e.g. in our institute Váncza at al. [21] uses it in a robotic inspection planning system). These systems basically keep the original concept of Eriksson.

In the HPKB (High Performance Knowledge Environment) [6] some hundred thousand rules are performed in an intelligent knowledge environment. In this project the different intelligent components are called knowledge servers. The components are communicating with each other via the OKBC (Open Knowledge Base Connectivity) protocol [3] specified at Stanford.

5. KNOWLEDGE SERVER FOR CONTROLLERS (KSC) CONCEPT

Knowledge Server for Controllers (KSC) is defined as a server providing capability of intelligent data processing for other systems. It allows the basic system to reach external intelligent processing resources, because it does not have any. The KSC contains a high performance reasoning tool, and different knowledge based modules. All the modules have their special rules and procedures. The client system calls these modules, passes them specific data if necessary, and the KSC module can collect data if the knowledge processing requires. All the data acquisition and user interaction is done by the client system. It is clear that in KSC the clients have much more tasks than a simple browser based user interface and in the applications listed in the previous chapter.

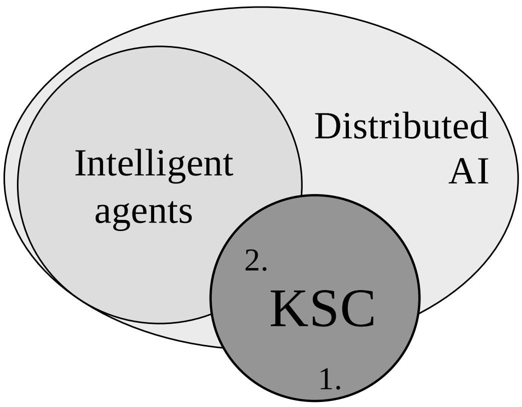


Figure 2. KSC in the distributed AI world

It should be stated that KSC does not deal with fuzzy and neural net based AI modules. The computing power and the necessary software costs and complexity of these methods are less than the rule- or model-based ones. (In the case of the neural nets it is true only if the net is not trained on-line.) Another important comment is that the KSC concept and the idea of the intelligent agents are different. Fig. 2 shows the KSC in the world of distributed AI, while the Table 3 compares an intelligent agent and a KSC module.

In Fig. 2 it can be seen that the KSC based systems basically belong to the world of distributed AI, but it is also possible that:

1. The client systems using the same KSC are local AI applications.
2. The KSC modules act as intelligent agents.

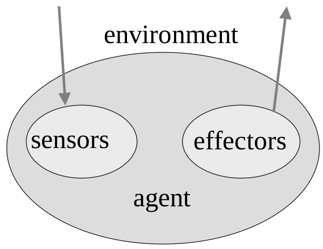
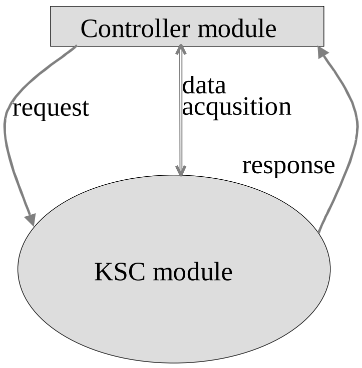
intelligent agent	**KSC module**
	 similarities with the inner structure of the simple reflex agent [19]
percepts	is called
autonomous	service
can launch	(in default case) can not launch
it has aim	it has task
temporal continuity	inactive, a request event starts it
optional features: cooperation, learning, adaptive, artificial personality etc..	optional features: agent like behaviour

Table 3. Features of an intelligent agent and a KSC module

The KSC allows the different modules to run independently, to co-operate as agents or to control each other. The third case means that one module is started by another one because either the second one uses the results of the first one or the inference of the first one led to the need of the second module.

Generally the resources of the KSC can use more clients (controllers or SCADA systems) simultaneously. It leads to a cost effective AI solution, because one costly AI environment can solve more problems parallel. So all the intelligent problems in a distributed environment are grouped in a single server. The overhead of the KSC (network connection, one more computer, some delay etc.) is much less than the advantages (AI environment licensing, less computing power in the clients/controllers etc.). It is also possible that one KSC service is used by more clients, e.g. when same type (or similar) machines/controllers are working in a workshop.

Using the KSC together with the component based software technology (e.g. CORBA) gives a very adaptive software frame to solve complex problems.

As an example a CNC with an embedded PLC controls a machine tool (see in the Fig. 3.). Assuming that the modules of both controllers (CNC and PLC) satisfy the reference architecture of a given (e.g. OMAC) open controller, they can be seen as boxes in the figure. If it is an intelligent CNC, than it is also possible that the inner algorithm of some modules need special, AI based calculations. So using the KSC concept, these modules are clients of the knowledge server. It means that these modules can use special AI methods during their work. These methods were implemented as independent services, and are running in the KSC.

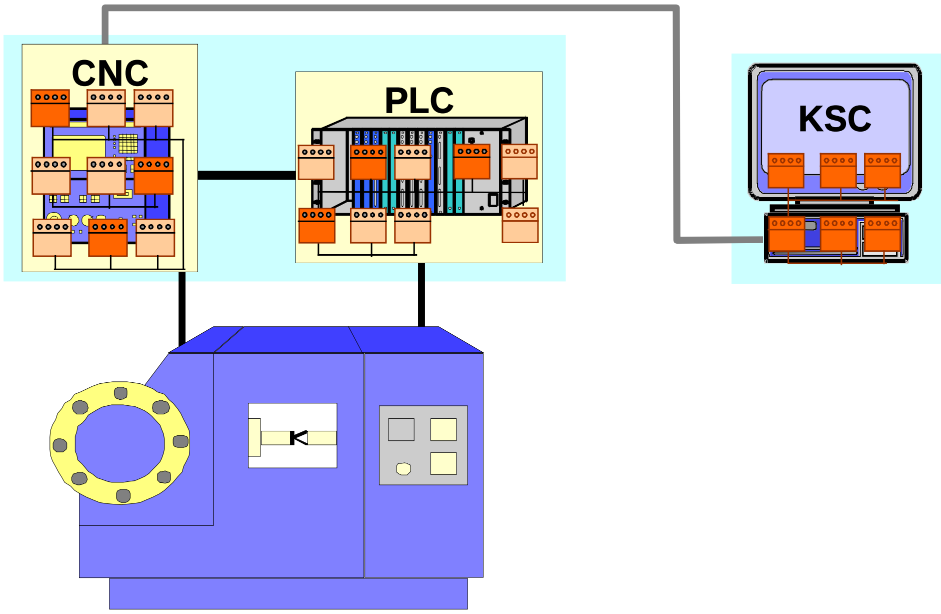


Figure 3. Intelligent CNC using KSC

6. PROTOTYPE CNC USING KSC

One of our colleagues, Sylvester Drozdik developed a Corba based open controller that controls a 3D milling machine (see. Fig.5) [7]. The controller is running on a pure PC with an Advantech PCL832 Motion Control Card. The first version of DAC was running on Windows95, while the Version 2 in QNX is under development. This controller was the basis in the development of the intelligent CNC prototype using KSC.

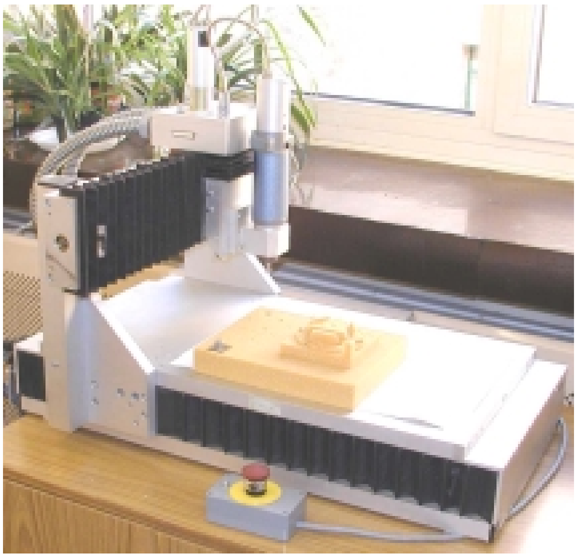


Figure 5. The 3D milling machine controlled by the prototype CNC

In the OMAC API specification [16] there is as an example a one axis equipment with homing and jogging functions. The architecture of this example supports position and velocity control. This example was selected to be the first prototype of the KSC concept in the CNC world.

So in this early prototype the axis module was implemented based on the OMAC module specification [16]. As a basic configuration the OMAC *Axis* module uses two *ControlLaw* modules to have position and velocity control over the real axis that is reached via the *IOPoints* module. The *Axis* is manipulated from the *AxisHMI*.

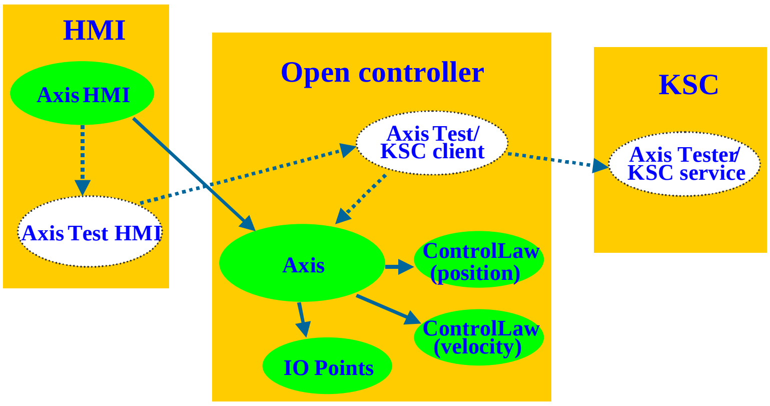


Figure 4. Intelligent tester of a machine tool axis

An advance axis tester is put on the top of this. The *AxisTest* module handles all the tests, and it gets the necessary position and velocity values from a knowledge based general tester running as an application on the KSC. The KB tester determines some goal positions and motion speeds, that the *AxisTest* module executes with the axis using jog commands. The results (execution time, tuning in errors etc.) are sent to the KSC that analyses and qualifies the axis.

In the prototype the modules are built in CORBA, the controller and the HMI is programmed in Java, while the KSC is based on G2 environment [9].

7. CONCLUSIONS

The different research results and the open problems of intelligent CNCs were shortly introduced. It was stated that many wishes from the 80's are still unsolved and many of the existing open issues claim AI solutions.

The knowledge server concept of Eriksson was introduced in the field of controllers with some important modifications comparing with the original idea. The features of KSC were discussed and an early prototype (axis tester) was introduced. Further works are going on to develop a complete intelligent CNC for a 3D milling machine using KSC.

It should be mentioned that a KSC was successfully implemented in an advisory system of an electrical substation of a nuclear power plant [14]. In this application the KSC was connected to a SCADA system and it supported 5 different intelligent decision support system functions.

8. REFERENCES

1. Amezaga J., Barg J., Brühl J., Lutz P., Nacsa J., Pohl M., Sozzi M., Wälde K., Zulauf P.: OSACA Handbook, OSACA Association, Stuttgart, Germany, 1997
2. Basden A: The Istar Knowledge Server, (2000), http://www.basden.u-net.com/pgm/Istar/index.html
3. Chaudhri F. V., R. Fikes, P. Karp, J. Rice. (1998) OKBC: A Programmatic Foundation for Knowledge Base Interoperability. Proc. of AAAI-98,

4. Cheng, T., J. Zhang, C. Hu, B. Wu, S. Yang (2001) Intelligent Machine Tools in a Distributed Network Manufacturing Mode Environment, Int. J. Adv. Manuf. Technol. 17:221-232
5. Chrysler, Ford Motor and General Motors. Requirements of Open, Modular Architecture Controllers for Applications in the Automotive Industry, 1994, White Paper - Ver. 1.1.
6. Cohen Paul R., Robert Schrag, Eric Jones, Adam Pease, Albert Lin, Barbara Starr, David Easter, David Gunning and Murray Burke. 1998. The DARPA High Performance Knowledge Bases Project. In Artificial Intelligence Magazine. Vol. 19, No. 4, pp.25-49
7. Drozdik Sz: (1) DAC System Design, (2) DAC Virtual Device, (3) DAC NC kernel, Reports in Hungarian, SZTAKI, 1999
8. Eriksson H: Expert Systems as Knowledge Servers (1996) IEEE Expert, Vol. 11, No. 3
9. Gensym (2000) A Strategic Choice for Improving Business Operations: G2 Classic, http://www.gensym.com/products/g2.htm
10. Kovács, G. L. (1996) Changing Paradigms in Manufacturing Automation. Proc. of IEEE International Conference on Robotics and Automation, Minneapolis, USA, Vol. 4. pp. 3343-3348
11. Lenat, D: CYC: A Large-Scale Investment in Knowledge Infrastructure, Comm. of ACM, 38(11) pp. 33-38, 1995; + Cyc Knowledge Server: http://www.cyc.com/products.html
12. Monostori L. (2000) Intelligent Machines, Proc. of 2nd Conf. on Mechanical Engineering, Budapest, Hungary, May. 25-26, pp. 24-36
13. Nacsa J, G. Haidegger (1997) Built-in Intelligent Control Applications of Open CNCs, Proc. of the 2nd World Congress on Intelligent Manufacturing Processes and Systems, Springer, Budapest, Hungary, June 10-13., pp. 388-392
14. Nacsa J., Kovács G. L. (2001) Alert State Detection and Switching Order Generation at a 400/120 kV Substation using Knowledge Server, ISAP2001 Conference. (Intelligent Systems Application to Power Systems), Budapest, 18-21 June
15. Nacsa J. (2001) Comparison of three different open architecture controllers, IFAC MIM, Prague, 2-4 Aug. 2001
16. OMAC API Work Group (2001) OMAC API SET, Version 1.0, http://www.arcweb.com/omac/
17. OSE Consortium, (1996) OSEC: Open System Env. for Controller Architecture Draft, http://www.mli.co.jp/OSE/
18. Pritschow, Sperling (1994). Open System Controllers - A Challenge for the Future of the Machine Industry. CIRP Annals
19. Russel, S.J, Norvig P: Artificial Intelligence. A Modern Approach, Prentice Hall Inc. 1995
20. Teti R, S. Kumara (1997) Intelligent Computing Methods for Manufacturing Systems, Annals of CIRP, Vol 46/2, pp. 629-652
21. Váncza J., Horváth M., Stankóczi Z.: Robotic Inspection Plan Optimization by Case-Based Reasoning, Proc. of the 2nd World Congress on Intelligent Manufacturing Systems and Processes, June 1997, Budapest, Hungary, pp. 509-515
22. Wright P.K., D.A. Bourne (1988) Manufacturing Intelligence, Addison-Wesley