APEM journal Advances in Production Engineering & Management 3 (2008) 1, 3-16
ISSN 1854-6250 Original scientific paper

THE APPLICATION OF A MULTI-AGENT SYSTEM FOR PROCESS MANAGEMENT AND NC DATA TRANSFER AIDED BY A DNC SYSTEM

A. Nestler A. & Dang T.N.
Dresden University of Technology,
Institute of Manufacturing Technology,
Chair of Production Automatisation,
Cutting and Erosion Manufacturing Engineering
Mommsenstr. 13, D-01062 Dresden, Germany
E-mail: nestler@mciron.mw.tu-dresden.de

Abstract:
In the network-based DNC systems and especially heterogeneous DNC systems, data get more distributed due to the ability to store not only in the DNC Server, but also in the CNC controller or CNC Terminal. This presents a challenge to the DNC software in the implementation of its basic functions. The agent-based DNC software with assistants (ADNC) is suggested to solve the above problem. Its tasks are to collect distributed NC information, and transfer data safely as well as performing other extended functions, i.e. NC program generation, machine data collection, production data collection, control machines in system, etc. In this paper we presented our concept to develop the ADNC, and its first prototype. This prototype was developed in a process using a framework of agent-based assistants, called Agent AP. It is applicable on distributed manufacturing data and had been implemented on the agent platform JADE. The module Program Assistant, one of these prototype modules, which is responsible for management, bidirectional transfer, and monitoring change of NC programs, is also discussed.

Key Words: NC program, DNC system, agent-based assistance, multi-agent system

1. INTRODUCTION

In recent years, rapid development of the electronic technology as well as the software technology has had great impact on the numerical controllers in today's CNC machines. These numerical controllers are more intelligent, flexible, having also more internal memory capacity and provide various connection standards. This leads to a change in the way the DNC systems are established, especially with the heterogeneous DNC system based on the previous available DNC systems. With such a DNC system, data get more distributed due to the ability to store not only in a central DNC Server, but also in the CNC controller or CNC terminal. The change of this archive means requires the DNC software, in which they should perform completely the basic functions of DNC systems, such as distributed NC data managements, transfer in both directions to CNC machines. Moreover, they should also satisfy the new requirements, i.e. ability to allow users to build and use data relationships between part programs and other distributed shop data, such as tooling information, part drawings and production documentation, machine data collection, etc.

In the last years, agent technology i.e. MAS has been seen as a promising approach to address the problems of engineering today's complex software systems, such as heterogeneous system environments and distributed structures. In this paper we present therefore a prototype for agent-based DNC software (ADNC) in order to solve effectively above requirements.

2. DISTRIBUTED NC INFORMATION IN DNC SYSTEMS

2.1 Overview of DNC systems

The original definition DNC referred to Direct Numerical Control, which meant that a few numerically controlled machine tools were connected to the centralized computer, which stored the NC programs and, on demand, managed and distributed NC programs to the various machine control units [1]. This solved various problems caused by punched tape in early numerical control systems, and also substantially reduced setup time. However, one drawback remains that if the computer went down all its machines were inoperative [2].

The operating range of today's DNC, referred to as Distributed Numerical Control or Distributive Numerical Control, is distributed onto several computers, which communicate through a Local Area Network [3], and also facilitates two-way communication and information collection. The two-way communication allows the machine to receive and send data. This can be useful in storing adapted and changed part programs on the NC machine. Moreover, beside all functions of Direct Numerical Control, a today's DNC possesses the functions of information collection, condition monitoring and system control. Thus, the term DNC has come to mean both direct and distributed numerical control.

Today's DNC systems can be divided into three main groups: the RS232-based DNC systems, the terminal-based DNC Systems and the network-based DNC systems [3][4]. In fact, the RS232-based DNC systems are not used as often as the other systems. This is due to their disadvantages, such as many long cables, low level of safety transfer, slow transmission speed on distances > 15m, and they are mostly legacy software and outdated technology. On the contrary, the terminal-based DNC Systems and the network-based DNC systems are popularly used because of the following main advantages: the safe data transfer over any distance with high speed supported by controls, other production data (order data, labour data, etc.) can be collected at the same time, and they allow editing NC programs. Especially in network-based DNC systems NC programs can be directly accessed [4].

2.2 The heterogeneous DNC system

The DNC systems are considered as an integral component of today's flexible manufacturing systems. It means that in a manufacturing system many variants of DNC systems can be used. This requires the construction of a heterogeneous DNC system based on available variants of DNC systems in the manufacturing system due to the following reasons:

- The long-term investment in many phases or the enlargement of production leads to possessing many variants of DNC systems (so called DNC subsystems).
- The construction of a heterogeneous DNC system based on available variants of DNC subsystems achieves a more rational configuration or structure.
- DNC subsystems should communicate through LAN networks by which NC-information could be exchanged not only among the DNC subsystems, but also with CAD/CAM systems, CAP systems, etc.

A heterogeneous DNC system based on available DNC subsystems is described in Figure 1.

2.3 Functions of DNC systems

The VDI-3424 recommends two kinds of functions in a DNC system, i.e. basic functions and extended functions, those may be identified, irrespective of the type of control to which it is connected [1].

- The basic functions are the management of NC part programs and the distribution of the NC data at the appropriate time to the machine control units.
- Extended functions for DNC software are integrated according to the particular application and interface of the hardware into system. These extended functions may include NC program generation, storage and management of tool data, correction values, machine data collection, production data collection, etc.

In addition, today's DNC systems offer extended leading-functions, such as management of pallet, tool and work piece for several machines; flow of material control through the system and special functions according to request.

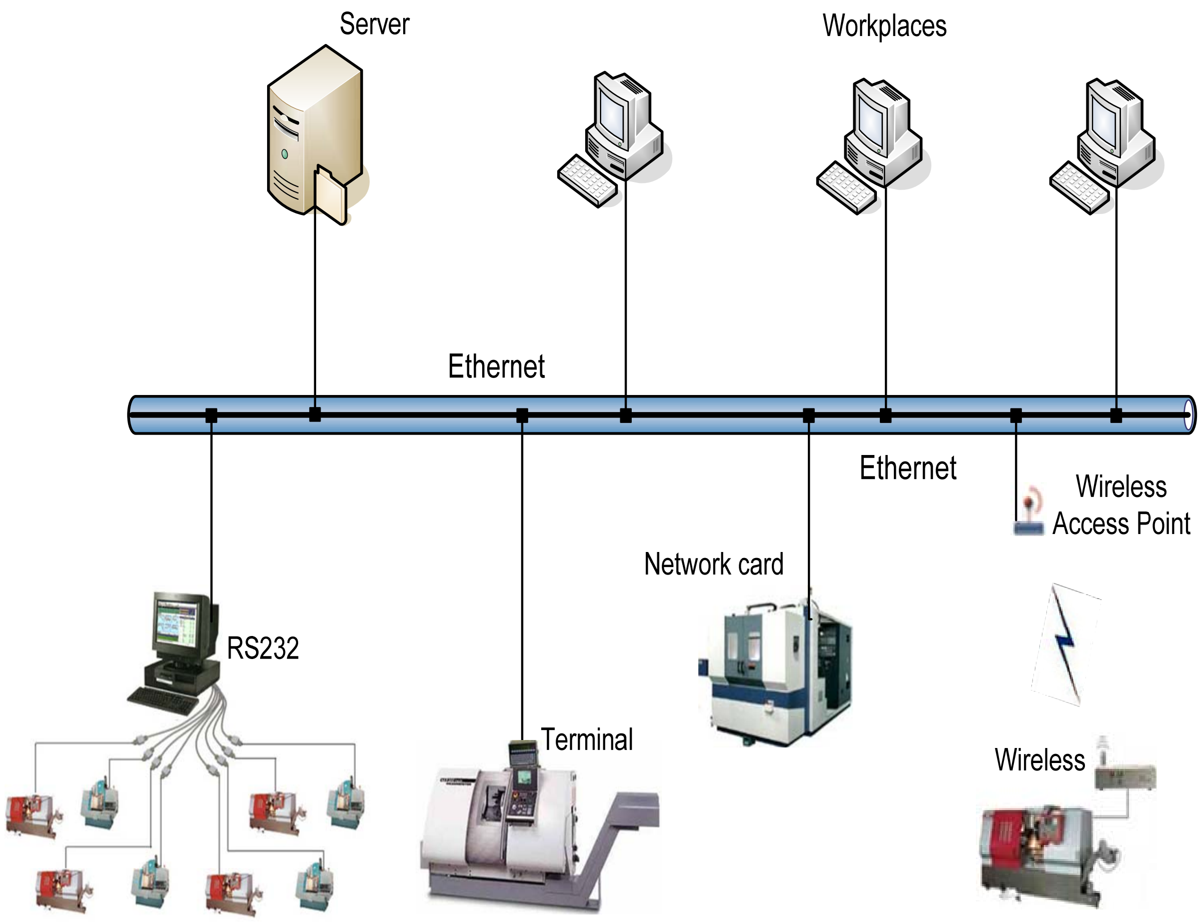

Figure 1: A heterogeneous DNC system.

2.4 DNC-Software

The basic and extended functions of a DNC system, as described in section 2.3, are realized by DNC software. Conceptually, a DNC software system allows centralized process control management for one or more NC machines or robots. The machines are typically connected to common routing points through a hardwired or wireless network. Control for each machine (or group of machines) is handled through network nodes via a series of dedicated software applications [5]. Today, the DNC software is responsible for the management and transfer of part programs, because additional modules can handle functions, such as machine cycle monitoring, paging and e-mail alerts, in-process gauging integration, paperless manufacturing, or customized modules made specifically for your application [6,37].

2.5 Deficiencies and new requirements of DNC systems

The features of a heterogeneous DNC system (see Fig. 1), a special form of the network-based DNC systems, are listed as follows:

- NC information can be stored distributional in the DNC server, CNC controller and terminal, especially in the workplace.
- Partners should communicate through LAN network, extranet or internet.
- Communication in the system can be accomplished by using hard-wired or wireless connection.

Based on features of a heterogeneous DNC system, as well as the ability of today's DNC software [4, 7-8], main deficiencies of DNC systems are determined as follows:

- There is a lack of the mechanism, or ability of the distributed NC-information management.
- Data losses and other errors during the transfer process of the NC data are not recognizable.
- Searching programs are complicated, because combinations of criteria cannot be realized.

Therefore, new requirements of DNC systems are offered based on the previously available functions, and the new technological conditions to eliminate the above mentioned deficiencies. These requirements are suggested to be:

- Management of different versions of NC programs
- Monitoring of NC data's change
- Searching NC programs for a special machine/control combination
- Overview of all NC programs with specification of the machine/control combination
- The DNC system should avoid data loss during the transfer process.

These require a new information management method, which can provide user and/or NC planner with necessary information. Agent technology is one of suggested potential methods to solve the above problem, in which software agents are fitted with abilities to support decentralised cooperative work for humans [9]. In agent supported collaborative NC planning the planner's team could coordinate and communicate with the manufacturer's team through software agents. Software agents of multi agent systems (MAS) have an excellent capability for intelligent manufacturing [10], as well as above mention tasks. Features, such as mobility, goal directedness, autonomy, communicativeness, collaborative and others, will be used for actions of acquisition and preparation of distributed technological information for NC planning and calculation [11].

Based on the above discussion, we used Agent Technology for establishing a distributed information collection for NC information management. This application will be discussed in the next sections.

3. AGENT TECHNOLOGY AND FRAMEWORKS

In our research, agent technology is selected to solve the remaining problems, which today's DNC software deals with. Before presenting the details of problem solving by this technology, we discuss the overview of agent technology, and their controversial problems, as well as used frameworks to develop our prototype.

3.1 Agents overview

The term "agent" is one of the difficult terms to define. Thus, over the past years, a lot of researchers in the area of agents and agent-based systems have offered a variety of definitions, which emphasize different aspects of agency. *Bradshaw* identifies two approaches to the definition of an agent as: (1) Agent as an ascription, (2) Agent as a description [12]. The second approach is considered appropriate for defining software agents in the context of the current report and from it has a many definitions of an agent. However, the definition of *Wooldridge* is one of the most frequently used by researchers, as follows [13]:

> *"An agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives."*

Although many definitions of agent have been proposed, an agent is generally characterised by the following [14][15][16]:

- An agent acts on behalf of its users.
- An agent is situated in an environment, and is able to perceive this environment.
- An agent has a set of objectives, and takes actions in order to accomplish these objectives.
- An agent is autonomous.

Autonomy is the most important property of an agent, without which the notion of agency would not exist. Autonomous agents can take decisions without the intervention of humans, or other systems based on the individual state and goals of the agent [17]. Beside the properties of autonomy, an agent should also be characterised by one or more of the properties, such as reactivity, pro-activeness, cooperation (social ability) [14,18]. The properties, i.e. autonomy, reactivity, pro-activeness, and cooperation are not optional, but instead they define the term agent. Wooldridge and Jennings have defined an intelligent agent as one that has full these properties [18]. Other properties, sometimes mentioned in the context of intelligent agents, include: learning/adaptation, mobility, temporal continuity and personality [12,19].

3.2 Multi-Agent Systems overview

Multi-Agent Systems (MAS) are becoming more and more popular in Computer Science and Artificial Intelligence, it endeavoured to attack more complex, realistic and large-scale problems, which are beyond the capabilities of an individual agent [20]. Therefore, various definitions from different disciplines have been proposed for the term multi-agent system (MAS). Wooldridge described that a MAS consists of a number of autonomous, intelligent agents, which can interact with one another in order to pursue their own goals or solve cooperatively common problems [13]. According to the FIPA 2003 [21] MAS is defined

> *"A Multi-Agent System is a system composed of a great number of autonomous entities, named agents, having a collective behaviour that allows to obtain the desired function/service."*

More recently, the term MAS has been given a more general meaning, and it is now used for all types of systems composed of multiple autonomous components showing the following characteristics [20]:

- Each agent has incomplete information or capabilities for solving the problem and, thus, has a limited viewpoint
- There is no global control system
- Data are decentralized
- Computation is asynchronous

Therefore MAS are ideally suited to represent problems that have multiple problem solving methods, multiple perspectives and/or multiple problem solving entities, in which MAS provide many potential advantages, such as efficiency, reliability, extensibility, robustness, maintainability, responsiveness, flexibility, reuse, adaptability, scalability, inter-operation of existing legacy systems [20][22]. In general, benefits of an agent-based information system include simplification of complex distributed computing, time savings, more and better information, better decisions, improved business processes etc. [23].

Agents in MAS must communicate with one another to exchange information and knowledge, by which they can solve cooperatively common problems. Therefore, the following characteristics i.e. coordination, negotiation and communication have become more crucial importance in MAS. The coordination is required to determine organisational structure amongst a group of agents and for task and resource allocation, while negotiation is required for the detection and resolution of conflicts [24]. The communication and cooperation with other agents is realised by specific languages, such as KQML [25], ACL [26]. It is still necessary for communicating agents to share a framework of knowledge in order to interpret message they exchange. This is not really a shared semantics, but a shared ontology [25]. Ontology makes this communication process become more meaningful when a common understanding of all the concepts exchanged by agents besides an ACL is used.

3.3 Ontology

Communication in multi-agent systems is possible only in the case that the communicating agents share their ontologies [27]. In other words, each agent must understand terms used in content of the message that it receives from other agents. Thus, these terms are defined by ontologies because they provide not only the means for describing intended semantics of the language used for expressing the content of messages, but also a machine-processional semantics of information sources, that can be communicated between different agents (software and humans) [28]. Ontologies are increasingly gaining their importance in interoperable systems to capture meanings and relationships of concepts, used in various domains. They are used to capture knowledge about some domain of interest, describe the concepts of the domain and also the relationships that hold between those concepts. Therefore, ontologies play a very important role in multi-agent systems, because of their collaborative nature of the system, agent-to-agent communication, knowledge management and interoperability reasons that exist between different database systems [23].

3.4 JADE

JADE (Java Agent Development Framework) is a software environment to build agent systems for the management of networked information resources in compliance with the FIPA specifications for interoperable multi-agent systems [36]. It is implemented fully in Java language, available on all the Java versions and now distributed under open source software (LGPL Version 2) license. JADE provides a middleware for the development and the execution of agent-based applications, which can seam-less work and interoperate both in wired and wireless environment. Its key features are an object-oriented agent communication language with support for ontologies, directory services and reusable application logic through so-called behaviours [29].

This framework can also facilitate the development of distributed multi-agent applications based on peer-to-peer communication architecture. It supports the development of multi-agent systems through the predefined programmable and extensible agent model, and a set of management and testing tools as well [30]. Moreover, JADE adopts not only the multi-thread solution, which is supported directly by Java, but is also able to schedule the tasks of cooperative behaviours [31].

With a rich suite of graphical tools, JADE supports both the management, monitoring and controlling of the status of agents, the creation and the execution of an agent on a remote host as well as control other agent platforms, included non-JADE platforms. JADE participated to the bake-off organized by FIPA where the interoperability with the other FIPA based middleware has been verified, and is currently under further development.

3.5 Framework of the Agent System AgentAP

The architecture of the developed prototype agentAP (AgentAP: Agentenbasierter Arbeits-Platz - that means agent-based workplace) is based on the model architecture for agent-based applications [32]. It is divided into three tiers following the principle of separation of concerns. Each agent type is equivalent to a certain layer in multi-tier architectures. The system as a whole (Fig. 2) is distributed across different machines including machining tool controllers, desktop and server systems. Each machine has an agent container, where in agents are executed.

Interface agents located at the top of these tiers are responsible for the user interaction. In our particular implementation they are linked to assistants, which provide a graphical user interface to be used to control the interface agent, which acts on behalf of the user. Although mainly targeted on user interaction interface agents can contain minor application logic in order to be able to conduct queries or locate and identify the active task and information agents.

Task agents, located at the middle layer of the system, are responsible for a well-defined piece of work within the application domain that is to be carried out in the agent system. Usually they continuously collect data from information agents. The collection of data is modelled according to the well-known observer pattern [33]. Task agents are informed by agent infrastructure, when a new agent is born and subscribed to their event source in order to be notified about events generated by the new agent. Such events (so called process data) are then processed and stored in a local small footprint database. Task agents could decide not to store particular data with the similar information. Other agents, using an agent communication language, could query the stored data.

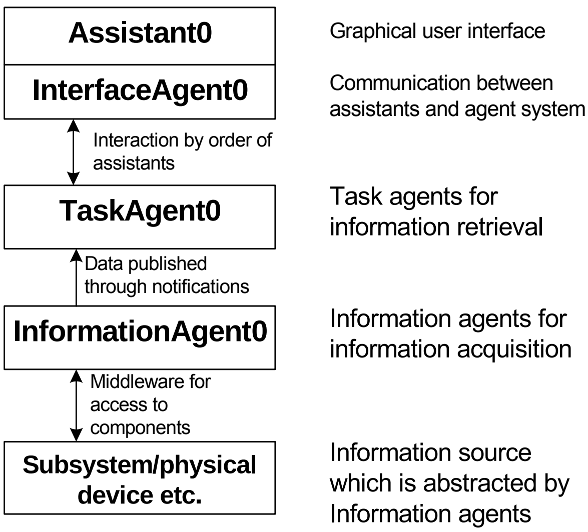

Figure 2: Abstract architecture of AgentAP.

Information agents operate at the lowest level of the multi-tier architecture, and are responsible for collecting and distributing data. They act as data sources within the system. Such agents can represent a physical device like a machine tool control. They primarily monitor their physical counterparts, and publish events that occur within these devices to interested parties of the agent system. Information agents do not process data for a particular purpose, but generate data on a general level. It is the responsibility of task agents to receive these events, and to draw further conclusions from them. However, the generated events are not sent as raw un-interpreted data, but in an object-oriented pattern.

For a given task of the application domain there will be usually, but not necessarily, one interface, task and information agent respectively. Tasks, which are not specific to the considered application domain are handled by service agents. Such agents can provide services to authenticate and authorise users, or give access to an object-relational database. These agents form the infrastructure to be used by other agents. The agent development framework itself contains a number of such service agents that mainly provide directory services.

The system implementation is based on the JADE, and fully written by Java language.

4. PROTOTYPE REALIZATION

4.1 Concept agent-based DNC software

The concept agent-based DNC software focuses on the tasks to perform basic functions as well as extended functions of DNC systems, in which NC information can be stored distributionally. From this focus, the ADNC is constructed based on platform JADE and framework AgentAP as following:

- It is developed depending on module idea, in which each module undertakes a few fixed functions. This means we can add other needed functions to the DNC software by adding new modules into software.
- Its conceiving consists of 2 main modules: Program Assistant and Machine Assistant.

The concept of the agent types and the overall system involving assistants, task and information agent of the suggested ADNC is described in Figure 3.

The Program Assistant is built as an essential module of the ADNC in order to perform basic functions of a DNC system (see 2.3). Its suggested tasks are listed below:

- The management and monitoring of NC program's changes
- The bidirectional safe transfer of NC programs.

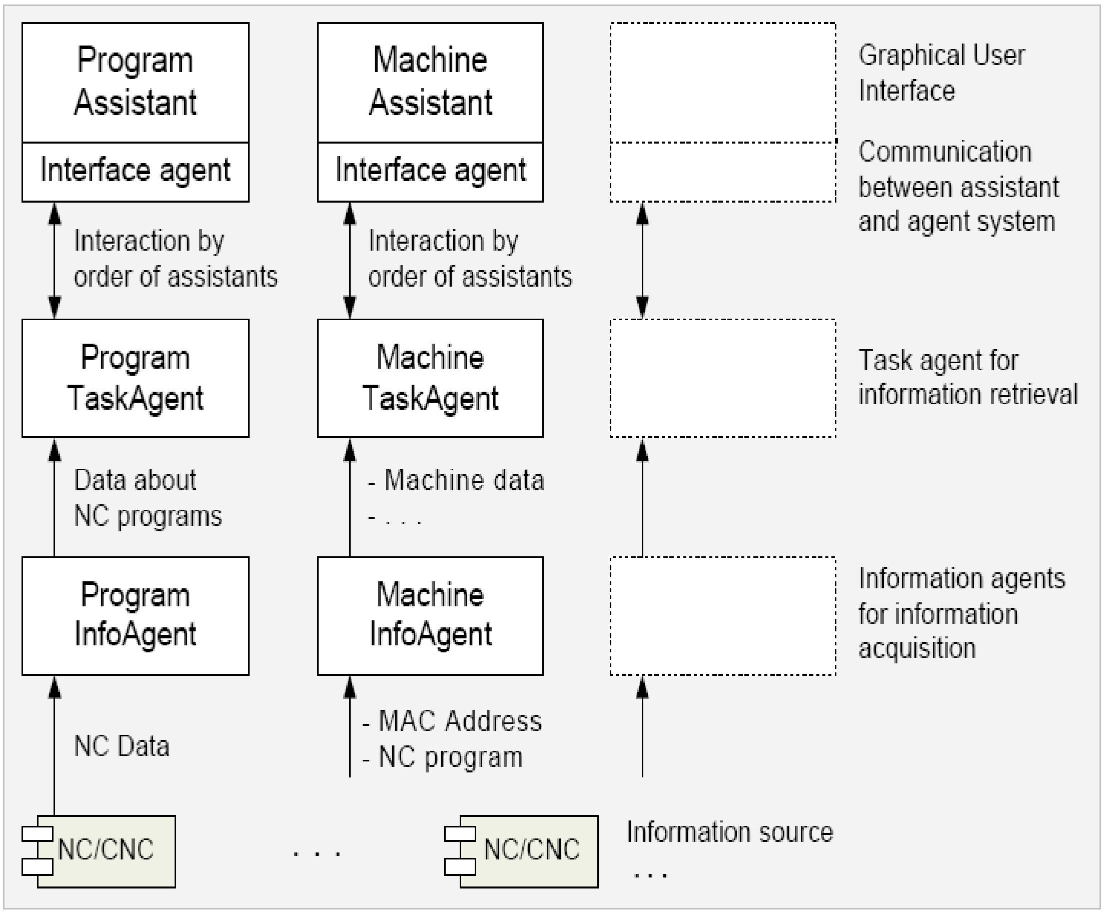

Figure 3: Concept of agent-based DNC.

The Program Assistant can be used not only for NC programs, but also for various NC data, such as STEP-NC programs, tool lists, documents, etc.

The Machine Assistant, that is an extended module of the ADNC, is established with the projected functionality about the MDE/BDE in DNC systems. These main tasks include:

- Collecting and offering the machine data of a NC/CNC machine, such as machine type, working space, work piece spectrum and other.
- Monitoring the machining process at the CNC machine, and then offering the collected information. That means, the machine assistant can offer the user online-information about the running machining step through the way of collecting, and the decoding of each NC sentence, which is performing at the CNC machine.

In addition, further Assistants can be added into software as new modules. Each module will be built to undertake other fixed functions.

From this concept we designed a prototype, in which the module Program Assistant was implemented as the first module. This prototype was developed based on a framework of an agent-based assistance system (so called TAssist) [34].

The framework TAssist is implemented to make the user's agent system become more useful. It includes arbitrary assistants, which can be integrated into the assistance framework through a plug-in mechanism. This framework is responsible for presenting assistants that are currently running or which can be started by the user.

4.2 Program Assistant

An overview about the tasks of the Program Assistant is introduced in section 4.1. In this section, the design method and the way of implementation of Program Assistant will be presented in more detail.

4.2.1 Management and monitoring of NC program's changes

The Program Assistant, who was supported by appropriate agents, such as information agents and task agents (see Fig. 3), could look for NC programs in CNC machines of a DNC system, and indicate all requested information on a GUI interface. It could also inform about the changes of NC programs, which took place outside of the reach of the NC planner. The Program Assistant, including, could present the information of found NC programs as follows:

- Filename, drive, path, file size, date of file creation, date of file modification.
- Version, identification of product, machine and comment on the NC program.
- Changes of NC programs as renaming, deletion, modification are also displayed.

In order to execute these jobs, the Program Assistant should offer information sources. They are the detailed information of NC program files and NC program headers, which were stored distributionally in the system.

Source of file information are information or properties of files, which are found at the archive location. These properties include drive, path, file size, date of creation and modification.

The NC program header, which was used as an information source for the Program Assistant, provides the information as: program name, version, project name, product identification, machine and comment [35].

It is a fact that today there are not any consistent standards for the NC program header. Each manufactory had, therefore, its own standard for construction of the NC program header. The Figure 4 represents a NC program header, which was established by our suggested standards. This NC program header was used as an information source in the Program Assistant.

; PROGRAM: KUGEL	(Program name)
; VERSION: 1.3	(Version)
; PROJECT: QUADER	(Project name)
; PRODUCT ID: P00123	(Product identification)
; MASCHINE: MAHO-800C	(Machine)
; COMMENT: Good	(Comment)

Figure 4: Part of the NC program header suggested.

4.2.2 Bidirectional transfer of NC programs

NC program files can be transferred either to and from the CNC control, or to and from the workplace. The necessary information, such as the source host and destination host, should be provided by the GUI of the assistant, to transfer selected files from the list of found NC program files (see section 4.2.1). As a result, NC program files can be transferred through a network from a workplace to a specific CNC control and vice-versa.

4.3 Implementation of Program Assistant

The overall system of the prototype Program Assistant is illustrated in Figure 3. It has a task agent and information agents, which can respectively handle the defined jobs as follows:

- Collecting detailed file information of NC programs from the CNC machines, NC programming systems, workplaces, etc.
- Monitoring version and changes of the NC data
- Data of communication between user and CNC machines
- Automatic logging of the data transferred in two-way manner, i.e. data received and sent by the machines.

An information agent ProgramInfoAgent supervises a specific folder on the CNC controller, where the NC data have been stored. A CNC controller used these data in order to conduct orders commissioned. Detailed file information, as well as the changes of NC data (e.g. renaming deletion or modification), will be collected and stored by the ProgramInfoAgent. This collected information is then forwarded continuously to a ProgramTaskAgent, that is a corresponding task agent of the ProgramInfoAgent, and responsible for collecting the data, which are generated by ProgramInfoAgents using an observer software design pattern.

The ProgramTaskAgent searches for all active ProgramInfoAgents. It can be informed about the start of new ProgramInfoAgents by JADE module. After that it registers itself with these ProgramInfoAgents to receive messages about the files' changes, such as the file creation, modification, renaming and deletion. By the way of using these messages, the ProgramTaskAgent can maintain the two following lists [35]:

- List 1: a list of all files on all machines, in which each entry of each version at each time contains the data namely i*dentifier, version number*, *MD5, file information, machine name, product identification data*, c*omment*, s*tate of the file*.
- List 2: a list of information about NC programs, those are available at the moment on the individual machines such as *machine name, identifier, MD5, version number*.

The ProgramTaskAgent administrates both of the above-mentioned lists. This means that it will add new entries, when new files are created, or delete entries when files were deleted.

The ProgramAssistant would perform the presentation of collected data on NC programs. NC planning or programming personal can be provided with information about the existence of NC programs, which were located in different CNC machines of the DNC system. Furthermore, NC programs can be selected through the ProgramAssistant and then transferred.

The figures below provide an illustration about the agent and assistance system (Fig. 5), and a list of files, which exist on specific folder of the CNC controller of the milling machine MAHO-800 (Fig. 6).

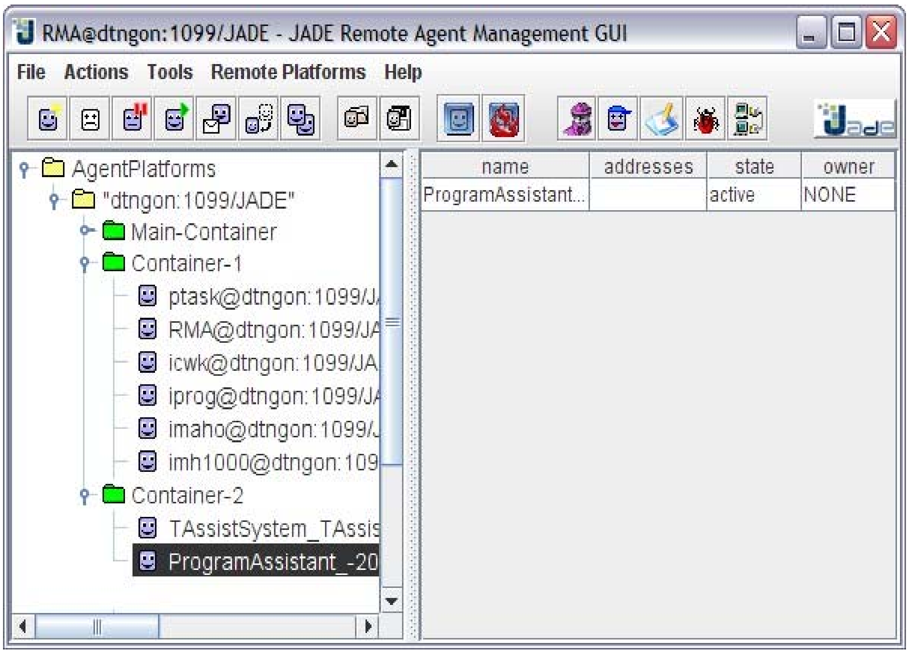

Figure 5: Agent and assistance system.

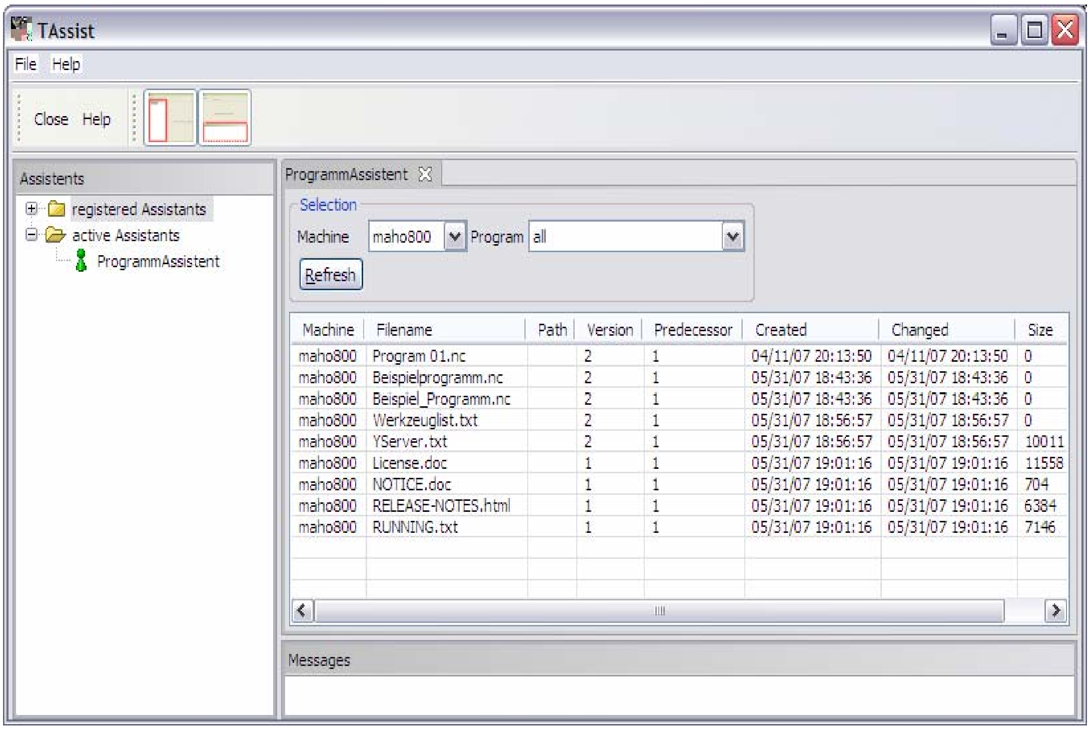

Figure 6: List of files on the Machine MAHO-800.

5. CONCLUSION

NC data in today's DNC systems get more distributed due to the ability to store, not only on the server, but also at the other archive locations. With this distributed store environment, DNC software should be built on a new architecture base, which allows performing smooth functions of DNC systems. The concept of cooperative agent-based data acquisition, from decentralised databases in order to retrieve distributed NC information the so-called ADNC, was presented. It includes two modules: (1) Program Assistant and (2) Machine Assistant, which could perform tasks, such as the management and the distribution of the NC programs, machine data collection, production data collection, etc. Besides that ADNC also has an open structure that allows adding new modules; each module would undertake a few fixed functions. In this paper, a module Program Assistant, the first module of the ADNC, was developed based on a framework of an agent-based assistance system called TAssist, and implemented on the agent platform JADE. It was shown by means of the pilot software system, presented here, that it is possible in principle to improve the distributed NC information management in a DNC system. The Program Assistant could not only provide NC planning or programming persons with the ability of management, but also allow monitoring the changes of distributed NC information, which took place outside of the reach of them. Current and future works will focus on completing the development of transfer data security and implementing module Machine Assistant.

REFERENCES

[1] Weck, M. (1984). *Handbook of Machine Tools, Vol*. 3: Automation and Controls, John Wiley & Son Ltd., Chichester

[2] Robb, D.; Estates, H. (1998). Look What's Happened To DNC, *Metalworking Technology Guide* (available at: http://www.mmsonline.com/articles/mtg9809.html)

[3] Kief, H. B. (2005). *NC/CNC Handbuch*, Carl Hanser, München, Wien

[4] Qunix AG. Netzwerk-DNC, from http://www.quinx.com/de/netzwerkdnc/index.html, accessed on 10.04.2007

[5] CIMCO Integration I/S. CIMCO DNC-Max 5, from http://www.cimco-software.com/product_description.php3?prod_id=12, accessed on 10-04-2007

[6] Lorincz, J. (2007). Wired and Wireless Systems, *Manufacturing Engineering Magazine*, Vol. 138, No. 1

[7] DLoG (UK) Limited. DNC - Direct Numerical Control, from http://www.dlog.co.uk/dnc.html, accessed on 10-04-2007

[8] COSCOM Computer GmbH. DNC Overview, from http://www.coscom.de/en/produkte/dnc/, accessed on 10-04-2007

[9] Ferber, J. (2001*). Multiagentensysteme*, Addisson-Wesley, Munich

[10] Shen, W.; Norrie, D.H. (1999). Agent-Based Systems for Intelligent Manufacturing: A State-of-the-Art Survey. *Knowledge and Information Systems, International Journal*, 1(2), 129-156

[11] Fichtner, D.; Nestler, A.; Dang, T. N.; Schulze, A.; Carlsen, U.; Schreiber, S.; Lee, S.W. (2006). Use of agents and neural networks for acquisition and preparation of distributed NC information to support NC planning, *International Journal of Computer Integrated Manufacturing*, Vol. 19, 6, 581-592

[12] Bradshaw, J. M. (1997). An Introduction to Software Agents, Bradshaw, J. M. (Editor), Software Agents, The MIT Press, Cambridge, 3-46

[13] Wooldridge, M. J. (2002). *An Introduction To Multiagent Systems*, John Wiley & Son Ltd., Chichester

[14] Nwana, H. S. (1996). Software Agents: An Overview, Knowledge Engineering Review, Vol. 11, No 3, 205-224

[15] Wooldridge, M. (1999). Intelligent Agents, Weiss, G. (Editor), Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, The MIT Press, 27-77

[16] Jennings, N. R. (2000). On agent-based software engineering, *Artificial Intelligence*, Vol. 117, No. 2, 277-296

[17] Chira, C.; Chira, O.; Roche, T. (2005). Multi-agent Support for Distributed Engineering Design, Ali, M.; Esposito, F. (Editors), Innovations in Applied Artificial Intelligence, IEA/AIE 2005, Bari, Italy, Springer, Berlin Heidelberg, 155-164
[18] Wooldridge, M. J.; Jennings, N. R. (1995). Intelligent Agents: Theory and Practice, *Knowledge Engineering Review*, Vol. 10, No. 2, 115-152
[19] Franklin, S.; Graesser, A. (1996). Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents, Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages, 21-35
[20] Sycara, K. P. (1998). Multiagent Systems, *The AI Magazine*, Vol. 19, No.2, 79-92
[21] The Foundation of Intelligent Physical Agents (2003). *FIPA Methodology: Glossary of Terms (Rel. 1.0)*, Status: 2003/11/18
[22] Jennings, N. R.; Sycara, K.; Wooldridge, M. (1998). A Roadmap of Agent Research and Development, *Journal of Autonomous Agents and Multi-Agent Systems*, Vol. 1, Issue 1, 7-38
[23] Lavbič, D.; Šaša, A.; Krisper, M. (2006). Knowledge Management in Multi-agent Systems, The International Conference on Knowledge Management in Organisations (KMO'2006)
[24] Green, S.; Hurst, L.; Nangle, B.; Cunningham, P.; Somers, F.; Evans, R. (1997). *Software Agents: A review*, Trinity College Dublin & Broadcom Ireann Research Ltd., Ireland
[25] Finin, T.; Fritzson, R.; McKay, D.; McEntire, R. (1994). KQML as an Agent Communication Language, Proceedings of the 3rd International CIKM'94, 456-463
[26] The Foundation of Intelligent Physical Agents (2002). FIPA ACL *Message Structure Specification*, Status: 2002/12/06
[27] Gruber, T. R. (1993). A Translation Approach to Portable Ontology Specifications, *Knowledge Acquisition*, Vol. 5, Issue 2, 199-220
[28] Fensel, D. (2001). Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce, Springer, New York
[29] Caire, G. (2003). Jade Tutorial - *Jade Programming for Beginners*, TILAB, Italy
[30] Bellifemine, F.; Bergenti, F.; Caire, G.; Poggi, A. (2005). JADE - A Java agent development framework, Bordini, R.; Dastani, M.; Dix, J.; Seghrouchni, A. E. F. (Editors), Multi-Agent Programming - Languages, Platforms and Applications, Springer, New York, 125-147
[31] Bellifemine, F.; Poggi, A.; Rimassa, G. (2001). Developing multi-agent systems with a FIPA-compliant agent framework, *Software: Practice and Experience*, Vol. 31, Issue 2, 103-128
[32] Shehory, O.; Sycara, K.; Jha, S. (1997). Multi-agent Coordination through Coalition Formation, Singh, M. P.; Rao, A. S.; Wooldridge, M. (Editors), Lecture Notes in Artificial Intelligence, Vol. 1365, Springer, London, 143-154
[33] Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J. (1994). Design Patterns: Elements of Reusable Object-Oriented Software, Addison Wesley
[34] Schulze, A.; Fichtner, D.; Nestler, A., Dang, T. N. (2005). Agent-based Assistant for Distributed Manufacturing Data, Jedrzejewski, J. (Editor), Machine Engineering. Intelligent Machines and Factories, Machine Engineering Editorial Office, Wroclaw, Vol.5, No. 1-2, 2005, 27-35.
[35] Nestler, A., Dang, T. N. (2007). Agent Based Assistance for Distributed NC Information in DNC systems, 6th International Conference on Industrial Tools and Material Processing Technologies (ICIT&MPT 2007), 291-296
[36] JADE Homepage: http://jade.tilab.com/, accessed 2007/11/11
[37] Balic, J. (1999). Contribution to integrated manufacturing, DAAAM International publishing, Vienna