



HAL
open science

**(Special Issue STEP-NC) Use of agents and neural networks
for acquisition and preparation of distributed information to
support NC planning**

Dieter Fichtner, Andreas Nestler, T. N. Dang, Andre Schulze, Ulf Carlsen, Steffen
Schreiber, Seok Won Lee

► **To cite this version:**

Dieter Fichtner, Andreas Nestler, T. N. Dang, Andre Schulze, Ulf Carlsen, et al.. (Special Issue STEP-NC) Use of agents and neural networks for acquisition and preparation of distributed information to support NC planning. *International Journal of Computer Integrated Manufacturing*, 2006, 19 (06), pp.581-592. <10.1080/09511920600651956>. <hal-00513372>

HAL Id: hal-00513372

<https://hal.science/hal-00513372v1>

Submitted on 1 Sep 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization



(Special Issue STEP-NC) Use of agents and neural networks for acquisition and preparation of distributed information to support NC planning

Journal:	<i>International Journal of Computer Integrated Manufacturing</i>
Manuscript ID:	TCIM-2006-IJCIM-0032
Manuscript Type:	Special Issue Paper
Date Submitted by the Author:	23-Feb-2006
Complete List of Authors:	Fichtner, Dieter; Dresden University of Technology, Institute of Production Engineering Nestler, Andreas; Dresden University of Technology, Institute of Production Engineering Dang, T.; Dresden University of Technology, Institute of Production Engineering Schulze, Andre; Dresden University of Technology, Institute of Production Engineering Carlsen, Ulf; Dresden University of Technology, Institute of Production Engineering Schreiber, Steffen; Dresden University of Technology, Institute of Production Engineering Lee, Seok Won; Dresden University of Technology, Institute of Production Engineering
Keywords:	NEURAL NETWORK APPLICATIONS, MANUFACTURING INFORMATION SYSTEMS, MANUFACTURING ENGINEERING, DATA MINING, DISTRIBUTED MULTI-AGENT SYSTEMS, EVOLUTIONARY ALGORITHMS
Keywords (user):	Technological Database, Distributed NC-information



For Peer Review Only

IESM-05 Special Issue**Use of agents and neural networks for acquisition and preparation of distributed NC information to support NC planning**

D. FICHTNER, A. NESTLER*, T. N. DANG, A. SCHULZE, U. CARLSEN, S. SCHREIBER, S. W. LEE

Technical University of Dresden, Institute of Production Engineering, Chair of Production Automatisations, Cutting and Erosion Manufacturing Engineering, D-01062 Dresden, Germany

Correspondence *nestler@mciron.mw.tu-dresden.de

Abstract

SPEP-NC information offers new possibilities for object-oriented data management. A strong need of high information content and the more increasing object-oriented CAM software requires a new information level for preparation and execution tasks in part manufacturing. This paper describes applications of different intelligent methods especially to support distributed technological data management. The problem of current technological databases will be solved with the help of agent technology and machine learning for distributed information on shop-floor and planning departments. The authors introduce a combination of agent-based organisation and self-learning of feature-based technological information. The prototype includes three components for coordination, data acquisition and preprocessing of technological information to assist human engineers in planning and manufacturing.

Keywords:

CAP, CNC, Technological Database, Distributed NC-information, Agent Technology, Agent-based Assistance, Artificial Neural Networks, Self-learning Procedure

AMS Subject Classification:**1 Introduction**

The idea of this research is to find out methods for acquisition and preparation of distributed technological information for NC planning and calculation methods in part manufacturing. STEP-NC information is relevant to the research because of the object-oriented and high level information characteristics based on feature technology. Features stand for information and integration objects in the computer aided manufacturing process and entire process chains (VDI 2218, 2003). The rapid development of advanced information and communication technologies requires extended feature-based information management (Fichtner and Nestler 2001).

The content of the paper is subdivided into the main points data model interpretation, intelligent methods for extended data management, agent-based assistance and flanking CAM functions based on STEP-NC information.

Today the technological information of enterprises is more and more distributed and data users are subjected to communicate more intensively. Continuous data acquisition in connection with searching in decentralised CNC databases is necessary for cooperative NC planning and NC manufacturing (figure 1). The information processing become labour-, time- and cost-intensive. One way out is to organise concurrently the provision of technological data in proactive processes.

'[Insert figure 1 about here]'

STEP-NC information is a prerequisite for high level information management. Therefore new information technologies enable a new horizon of user support. In establishing our objectives an automated information supply for calculation methods of CAM functions will be developed. In many instances, planners and skilled workers are missing necessary information support for cutting segments. An agent-based assistance system for distributed manufacturing data should use intelligent methods in connection with machining objects.

2 Intelligent methods for technological information

Up to now the quality of technological information - here especially cutting values - is a task of high interest for NC planning and NC manufacturing. The often practised way to only consider the combination of cutting values with calculated tool path is not sufficient for a continuous improvement of cutting values. Technological functions of CAD/CAM systems need a link to stored cutting data, e.g. data from a tool manufacturers catalogue or data for further describing attributes of calculation methods. Traditional calculation methods for cutting values are based on mathematical models with a static database. A main problem is to search, collect and prepare the correct data for current manufacturing tasks.

Agent technology is to be investigated for data management in networks. Software agents are fitted with abilities to support decentralised cooperative work for humans (Ferber 2001). In agent supported collaborative NC planning the teams of planners can coordinate and communicate with the teams of manufacturers through software agents. Software agents of multi agent systems (MAS) have an excellent capability for intelligent manufacturing (Shen and Norrie 1999), as well as the aforementioned tasks. Characteristics 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 (Fichtner et al. 2002).

A support for complex technological problems can also be given by intelligent computation which includes artificial intelligence methods, like artificial neural network, genetic algorithms, fuzzy logic or case based reasoning. One of the users motivations for developing and implementing intelligent computation in manufacturing relates to improved decision making, feature processing, flexibility, productivity, control strategies, accuracy and optimisation, enhanced real-time operation, automation and integration (Teti and Kimura 1997, Meziane et al. 2000). Artificial neural networks (ANN) are able to learn from technological examples and apply this knowledge to other, unknown situations due to a good generalisation ability (Fichtner et al. 1998). The Prediction of cutting values in connection with a technological database and with support by ANN has been described in (Fichtner et al. 1999).

In the following high level information of manufacturing features based on STEP-NC and intelligent methods for data management will be combined. A framework of an agent-based assistance for collecting and preparing process data will be presented.

3 Framework of the presented system

The architecture of the developed prototype agentAP is based on a model architecture for agent-based applications. The architecture consists of three major types of agents following the principle of separation of

concerns. Each type of agent is equivalent to a certain layer in multi-tier architectures. The system as a whole is distributed across different machines including machining tool controllers, desktop and server systems. Each machine executes an agent container wherein agents are executed. The overall system is shown in figure 2.

'[Insert figure 2 about here]'

Interface agents are responsible for the user interaction and located at the top of this three tier architecture. Task agents are long running instances which are capable of doing work and form the middle layer of the system. Information agents collect and distribute data and operate at the lowest level. For a given task of the application domain there will be usually but not necessarily one interface, task and information agent respectively.

Interface agents are utilised by users on their personal computer to interact with the agent system. In our particular implementation these agents are linked to assistants. Assistants 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 active task and information agents.

Task agents are responsible for a well defined piece of work within the application domain that is to be carried out in the agent system. All task agents reside within a single agent container that is executed on a server machine without user interaction. Usually they continuously collect data from information agents. The collection of data is modelled according to the well known observer pattern (Gamma, E., Helm, R., Johnson, R., Vlissides, J., 1994). Task agents are informed by the agent infrastructure when a new agent is born and subscribe to their event source in order to be notified of events generated by the new agent. Such events can be the process data that consists of a STEP-NC machining workingstep and the gathered cutting values.

This data is then processed and stored in a local small footprint database. Task agents can choose not to store particular data if they already know about some similar information. The stored data can be queried by other agents using an agent communication language.

Information agents act as data sources within the system. Such agents can represent a physical device like a machine tool control. These agents 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 uninterpreted data but in an object-oriented fashion.

Tasks that 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 Java Agent Development Framework (JADE) developed as open source code by the Telecom Italia Lab (TILAB). Key features are an object-oriented agent communication language with support for ontologies, directory services, reusable application logic through so called behaviours and a rich set of GUI tools to monitor and debug running agent applications (Caire, 2002). JADE is compatible to standards as laid down by the FIPA organisation.

An ontology is a means to describe things that exist in the reality. It defines actions and concepts that can be used in agent conversations. Ontologies are supported within JADE as objects that can be transmitted using an agent communication language. To define the classes that account for the ontology's concepts and actions a GUI tool can be used.

4 Interpretation of STEP-NC by open controls and extraction of process data

The main data sources in a machining process are control information from an open interface, low level NC code information and high level STEP-NC information (figure 3). The main interest is focused on object-oriented STEP-NC data. By using STEP-NC it is now possible to get information about machining objects, including geometry, technology and product data, directly on the shop floor. This information could be provided to an agent system, which is collecting data about the actual machining task.

[Insert figure 3 about here]

Data sources for STEP-NC part programs could be written in physical files (ISO 10303-21), XML files (ISO 10303-28) or databases. Independent of the written form of the STEP-NC data source, it is expedient to access the data via a standardised interface like Standard Data Access Interface (SDAI) (ISO 10303-22).

In order to use high level information, a so called STEP-NC interpreter provides process data for milling. The STEP-NC interpreter is the connector between the STEP-NC data and the numerical control. Therefore the interpreter has to fulfil a series of requirements. In addition to interpret STEP-NC data the interpreter should illustrate the working tasks. Beyond it is necessary to compare the available tools on the machine with the used tools in the STEP-NC part program and to change technologies or tools according to machine's possibilities and limitations if necessary.

The developed STEP-NC interpreter is based on a postprocessor prototype. Thereby it is possible to use existing functions of the CNC, e.g. cycles, to get a running prototype as fast as possible. By progressively upgrading the prototype we can obtain an ISO 14649 compliant prototype without any G and M code in result. For the structure of the prototype it is expedient to use a two parted construction. Thereby the interpreter consists of a graphical user interface (GUI) and a command line-based interpreter. Via the GUI the user is able to visualise the working features of the part program. It has the ability to do simple modifications on the STEP-NC data as well. So the user is supported and guided by visualisation. On the GUI the STEP-NC data is represented by a tree structure and by a 3D CAD model. The tree structure represents the object-oriented layout of the STEP-NC data model intuitively. For the geometric modelling OpenCascade (Open CASCADE 2000-2004, Rybak 2002) is used, because it supports a standardised data exchange by using the interfaces IGES (IGES 5.3) and STEP (AP202 and AP214)

List dialogs are suitable for giving a fast review on workingsteps, features, operations and tools. Therefore we integrated such dialogs in our application to increase convenience for users working with large part programs. The used tools of a part program should be verified with available tools on the machine. So we integrated appropriate algorithms for comparing and replacing tools in the part program to perform the machining task although the necessary tools of the part program are not available. Since the user has the possibility to manually replace tools, it is necessary that the part program is verified after changes are made.

Changed data of the part program can be saved, so all changes will also be available to the whole enterprise. By making use of agent-based systems it is possible to collect information about changed data and to visualise these changes.

The real interpretation of STEP-NC data is done by the command line-based interpreter application. At present, the STEP-NC workingsteps are mapped to cycles and G code of the CNC machine. Such a postprocessor solution is neutral to the greatest possible extent and can be used for controls of different manufacturers with little modifications. If the interpreter gets more functionality, it should individually adapted to the machine and the corresponding control. In that case the interpreter assumes functions, which were used to be performed by CAM systems, for instance the generation of tool paths.

In the agent system a MachineInfoAgent uses a subsystem to acquire data from a classic open numerical control. In our scenario the CNC called andronic2000 made by the company andron is used which itself is not

able to work with STEP-NC part programs. This subsystem consists of two parts that deliver data which has to be put into relation. On the one hand there is process data which is delivered by the numerical control and on the other hand there is the detailed object-oriented description of the machining process in the form of a STEP-NC document.

The STEP-NC interpreter software is used to generate classic NC programs which have additional comments embedded into the NC code that was generated through a translation of STEP-NC workingsteps. These comments include the machining workingstep entity instance as defined in the STEP-NC part program (figure 4).

'[Insert figure 4 about here]'

The numerical control offers a Windows developing library e.g. application programming interface (API) which helps to gain detailed information regarding current operations and configuration data. Continuous monitoring of machine movements and program execution is supported by call-back methods which deliver current data. The agent system is implemented in the Java language so that a compatibility layer had to be introduced. The Java Native Interface standard provides the means to use non Java code such as this library. Call-back methods of the Windows library are mapped to an object-oriented event mechanism which is more common in the Java language.

The subsystem asks the library to be notified of the current line number in the NC program. This information is used to determine which STEP-NC workingstep is currently executed by the machine tool. The detailed description of the workingstep is read from the original STEP-NC document which was used to generate the executed NC program.

Another similar approach was used in the past which included additionally usually unassigned M functions (miscellaneous function) in the NC program. These M addresses have to be recognised by the NC interpreter part of the numerical control which made a modification of this software necessary. No side effect is caused by the introduction of these addresses they are rather used to mark a certain point in the NC program. The execution of the NC program by the numerical control can be monitored through the call-back mechanism. When the numerical control reports the execution of such an introduced M address it is possible to conclude the current machining workingstep.

During the machining operation process data is retrieved from the numerical control. Currently only the maximum and minimum values of the cutting speed are analysed. Furthermore it would be possible to calculate cutting forces by analysing the torque measured by the numerical control and the tool geometry.

The gained data is mapped to the STEP-NC workingstep and published to interested task agents by the MachiningInfoAgent. The flow of information is shown in figure 5.

'[Insert figure 5 about here]'

For reading STEP-NC documents no free parser was available implemented in the Java language. Therefore a prototypical parser was created which is capable of reading a subset of STEP-NC entities. This parser reads clear text encoded STEP documents according to ISO 10303 – 21 and creates a graph of objects representing the document as an internal object-oriented STEP exchange structure. The object-oriented representation is based on a JADE ontology so that the document or parts thereof can be transmitted in agent messages easily. Objects of arbitrary ontologies can be created using an abstract form of ontologies. This abstract representation can be converted into a concrete ontology representation by the JADE framework. Therefore it is very convenient to create a object-oriented view of a STEP-NC document.

For the supported subset of entities a transformation of the graph is carried out generating a STEP-NC representation. The unsupported entities are left in their untransformed state as unrecognised STEP entities. The transformation process uses a proprietary description of the supported STEP-NC entities. In a complete

implementation the EXPRESS document describing STEP-NC would be used instead and no proprietary description.

An EXPRESS parser for Java should be implemented which can be used for the automatic Java code generation of the respective ontology.

The analysed process data together with the STEP-NC machining workingstep is sent to the MachiningTaskAgent which extracts characteristic data from the manufacturing feature and stores it in its knowledge base.

5 Preparation of collected data

In order to solve the preparation of collected data intelligent software methods are helpful. In our research these are artificial neural networks in connection with software agents. As a goal object-oriented cutting values are to be provided in an automatic way. Preprocessing should be realised by appropriate data management in a self-learning module designed to be applied for technological data stocks (figure 6).

Therefore, a two-step generation is assumed. In order to gain practical results with the method of artificial neural network human interaction is irreplaceable, firstly. The procedure for establishing a first ANN application can be subdivided into typical activity steps. In all steps a lot of different data stocks and parameters have to be managed. A developed project management tool can support handling and usage of all data stocks and net variants for the later application of a neural network.

'[Insert figure 6 about here]'

In the following will be checked, how a mostly automatic determination of cutting values can be organised. The proposed self-learning poses a high demand with respect to the different learning procedures (Zell 1998; Demuth and Beale 2000). The created software self-learning module can be used in two ways: in an interactive way controlled by the user (figure 7) and in a mostly automatic way. The highest increment will be controlled and supervised by software agents.

'[Insert figure 7 about here]'

These two ways offer a maximum of flexibility for both, the expert of neural networks and the user of the neural network application, e.g. the NC planner. A 'first-stage' useable neural network is planned to be created in a mainly interactive process by an expert. In most cases, best practice can only be guaranteed by this way. For practical use, it is planned to continuously update the user network mostly automatically with minimal human interaction. On the one hand, requirements of data delivery and management have to be considered, on the other hand, the way how to manage the network construction is to be paid attention to as well. R&D is aimed at a self-learning process capable of solving these demands within a defined technological scope.

The self-learning module integrates some basic functions for analysis, preparation of data and for creation, learning and estimation of ANN:

1. Functions for the analysis of data. For the quality of a neural network it is important to use data of good quality only. In order to assure this there are the following functions needed:
 - maximum-minimum determination
 - correlation analysis
 - frequency distribution.
2. Functions for the preparation of data. In order to use the data it must be presented in the way it is needed for the neural network software. The following functions are needed:
 - linear scaling in a specified interval (according to the chosen learning function)
 - dividing the data into learn, test and validation data

- changing the data into the specified format needed by the system.
3. Functions for the creation and learning of neural networks. In order to generate neural networks there are several steps necessary. These steps are written into batch scripts. These scripts contain all necessary information (e.g. learn function, parameters, files to use...). The use of these batch scripts provides a high level of flexibility.
 4. Functions for the estimation of neural networks. After finishing the learning process the best neural network is used to estimate its results. Therefore the following functions are implemented:
 - calculate and estimate percentual and absolute error of the neural network
 - frequency distribution of any encountered errors.

The following example shows the results of the learning process for manufacturing through the machining object pocket milling. For demonstration object-oriented data for an example was taken from the tool manufacturer catalogue FRAISA (Fraisa 1999).

The self-learning module was used to generate ANN mostly automatically using batch processes. Three different learning algorithms had been used to get the best result. These algorithms were Backpropagation, Levenberg and Bayes as shown in table 1. In this example the learn algorithm Levenberg created the best neural network. The sum squared error (SSE) is the lowest for this manufacturing feature.

'[Insert table 1 about here]'

The best trained network can transformed into a C program which means an enormous compression of data, compared to the training database. The user should be able to work with the net without any detailed knowledge about neural networks. The example shown in figure 8 demonstrates a user interface to determine the cutting values with the best ANN from the batch process. In order to compare different network topologies the result of the batch process and the result of the construction using the method of genetic algorithms are displayed in the user interface.

'[Insert figure 8 about here]'

In another example the self-learning module was used to generate ANN for different manufacturing features. These were the types through pocket milling, blind pocket milling and slot milling. The data for this example was also taken from the tool manufacturer catalogue FRAISA. The results are shown in table 2.

'[Insert table 2 about here]'

The batch process generates a specialised neural network for a special manufacturing feature, that means a new quality of technological information: specialised feature-based cutting values.

Further development is focused on how to deal with agent-based extension (figure 9). In this extension communication partners are neuro agents for preprocessing the collected data. Learning agents use self-learning modules with different methods. During the learning process all these algorithms were controlled by specialised learn agents (e.g. Backprop agent). After finishing the learning process the results as shown in table 1 were transferred to the neuro agent. This agent estimated the different neural networks and choose the best one.

'[Insert figure 9 about here]'

6 Implementation of a prototype system

The intention of self-organisation of feature-based information out of the NC manufacturing process for NC planning is demonstrated by a scenario. In the following example, the feature-based STEP-NC information represents different local knowledge bases at different NC machine tools.

A concept and first prototype for an agent-based information management used for machining objects of the milling process is based on technological information from the data sources STEP-NC, STEP-NC interpreter and control information from CNC. In our example, the feature-based technological information represents different local knowledge bases at different NC milling machine tools. The aim is a concurrent and continuous updating of a technological database and software assistants for machine tools and machining processes.

The prototype system agentAP focuses on the tasks to provide the NC programming personal with operating and process data that is collected on the shop floor. There are two task and information agents respectively handling the given jobs. The software introduced in chapter 4 is based on earlier works and has not yet been integrated into the new framework. The overall system is illustrated in figure 10.

[Insert figure 10 about here]

A MachineInfoAgent communicates with an open numerical control in order to be informed of current machining operations. This agent concentrates on retrieving operating data and forwards all other data to a MachiningInfoAgent. A STEP-NC based subsystem that was presented in detail in chapter 4 is used to monitor the machining tool. This system continuously notifies the agent about ongoing machining operations such as the completion of STEP-NC machining workingsteps or the completion of the execution of the NC program. The former information is considered as process and the latter as operating data.

Both information agents are executed once on every machine tool. The corresponding task agents called MachineTaskAgent and MachiningTaskAgent are executed only once on a server system. They collect the data generated by the information agents using an observer software design pattern.

The NeuroTaskAgent and related agents are not yet integrated into the new framework. These agents will communicate with the MachiningTaskAgent to get cutting values needed for the creation of ANN.

For gaining access to an object-relational technological database system developed in-house a respective service agent was implemented. It is used by the MachineInfoAgent to retrieve information describing the configuration of the machine tool. For instance the dimensions of the working space are stored in this database system. An agent for the authorisation and authentication of users and other identities uses this agent to store its data.

An assistance system called TAssist was implemented to make the presented agent system useful for the NC planning personal. Arbitrary assistants can be integrated into the assistance framework through a plug-in mechanism. The framework is responsible for presenting assistants that are currently running or which can be started by the user.

Assistants are tightly coupled to interface agents through which they can interact with the agent system. This is accomplished through the utilisation of the JADE in-process interface which provides the means to integrate JADE agents into other applications.

For displaying operating data of machines located on the shop floor on the NC planning personal's desk a MachineAssistant (figure 11) was implemented. This assistant locates all active MachineInfoAgents and shows their current state, the loaded NC program and the dimensions of their working space. Furthermore the operating data collected by the respective task agent can be queried. Its data set consists of e.g. NC program name and machine name as well as the duration of the complete program.

The collected process data is presented by a MachiningAssistant (figure 12). NC planning personal can use this knowledge base to lookup process data having similar values to get better initial cutting values. This approach has the advantage that only real life data gathered from production level NC programs is presented. The implemented user interface components for the preparation of collected data shown in chapter 4 have to be transformed into assistants which conform to the standard of this framework.

For the maintenance of the assistance system an assistant was implemented which can create, modify and delete users. Other assistants for management tasks are possible.

'[Insert figure 11a and figure 11b about here]'

'[Insert figure 12 a and figure 12b about here]'

7 Flanking CAM functions based on STEP-NC

In order to realise STEP compliant manufacturing philosophy based on ISO 14649 there are two development aspects. At first, in work preparation aspect: ISO 14649 must be seamless coupled with legacy CAM systems to exploit the diverse functions of existing CAM systems which are based on the feature technology.

At second, in the CNC controller aspect: To make STEP compliant intelligent CNC controller (SCICON), fundamental functionalities as follows are to be implemented:

- detection of disturbances during operations
- self-decision on process plans
- calculation of tool leading points
- control of machine axes (interpolation, acceleration and deceleration) and
- reporting of manufacturing conditions for further tasks

SCICON should be able to understand STEP based context including geometrical and technical information to manufacture with, to plan manufacturing processes based on available resources, to execute process plans and to report the results of processes and, if any, unexpected situations.

If the above mentioned functionalities are completely available on the CNC controller, machine tools can utilise their resources optimally coping with the manufacturing status.

For example, conventional machining centres have tool magazines equipped with redundant sister tools providing for the unforeseen machining situations (e.g. tool collision or tool breakage). If the sister tools are not equipped on a magazine after a tool breakage, it is considered as a erroneous NC program or extra manpower have to be charged for a process plan with a new tool at the work preparation level. SCICON, on the other hand, can detect abnormal manufacturing status, decide a new process plan and machine according to axis movement commands, that is, work more intelligently.

8 Outlook and summery

Considering the growing capabilities of information and communication technologies, enhanced information management for manufacturing features becomes even more important. Manufacturing features stand for uniform and continuous information and are regarded as suitable to represent the data relevant for NC planning and manufacturing processes. The authors discussed available STEP-NC information under the aspects of object-oriented NC planning and manufacturing.

Cooperative agent-based data acquisition from decentralised databases and neural self-learning modules to retrieve planning data were presented. The main point of assistance is the collecting and preprocessing of precedent technological data. Manufacturing features will be the information carriers in the future.

The self-learning of cutting values supports the user during the process of creating neural networks. At this time the shown self-learning module is usable only for some basic learning algorithms. In some cases the performance of the automatism is not quite as good as the performance of the interactive way. In the future better adjustment of the parameters to the technological problem will be developed.

References

- Teti, R. and Kimura, S.R.T., 1997, Intelligent Computing Methods for Manufacturing Systems. *Annals of CIRP*, Vol. 46/2 pp. 629-652
- Meziane, F. and a.o., 2000, Intelligent systems in manufacturing; current developments and future demands. *Integrated Manufacturing Systems*, 11/4 pp. 218-238
- Fichtner, D., Nestler, A. and Schulz, G., 1999, Wissensakquisition für Schnittwerte beim Fräsen unter Nutzung von neuronalen Netzen. *Fortschr.-Berichte VDI*, Reihe 20, Nr. 304, (Düsseldorf: VDI-Verlag)
- Fichtner, D., Nestler, A. and Schulz, G., 1998, An Approach to Technological Database - Possibilities for the Determination of Cutting Values with Neural Networks. In Migliore, H. et al. (eds.), *Flexible Automation and Intelligent Manufacturing, Proceedings of the Eight International FAIM'98 Conference*, Portland, Oregon, USA, July 1-3 (New York: Bergell house inc.), pp. 577-586
- VDI 2218, 2003, Feature-Technologie, *VDI-Richtlinien*, (Berlin: Beuth)
- ISO 14649, 2004, Industrial Automation Systems and Integration Data Model for Computerized Numerical Controllers, *International Organization for Standardization*, (Genf: ISO)
- Caire, G., 2002, Jade Tutorial – Jade Programming for Beginners, TILAB
- Fichtner, D. and Nestler, A., 2001, Manufacturing Features for NC-Programming with extended Information Management. In Deneux, D.; Soenen, R. (eds.), *Feature Modelling and Advanced Design-For-The-Live-Cycle Systems. Proceedings of the International IFIP Conference FEATS' 2001*. Valenciennes, France, 12-14 Juni 2001. University Valenciennes (LAMIH - UMR CNRS n° 8530)
- Fichtner, D., Nestler, A. and Arnold, F., 2002, Agent-based Acquisition and Preparation of Distributed Technological Information for NC Planning and Calculation Methods in Part Manufacturing. In Sullivan, W.G; Ahmad, M. a.o.(Eds.) *Proceedings of the 12th International Conference on Flexible Automation & Intelligent Manufacturing (FAIM'02)*, Dresden, Germany, Juli 15th-17th, 2002. München: Oldenbourg Verlag, pp.93-103, ISBN 3-486-27036-2
- Ferber, J., 2001, *Multiagentsysteme* (Munich: Addison-Wesley).
- Shen, W. and Norrie, D.H., 1999, Agent-Based Systems for Intelligent Manufacturing: A State-of-the-Art Survey. *Knowledge and Information Systems, International Journal*, 1(2), pp.129-156
- Zell, A. et al., 1998, *SNNS – Stuttgart Neural Network Simulator, User Manual, Version 4.2*
- Demuth, H. and Beale, M., 1999, *Neural Network Toolbox User's Guide*. (Natick, MA, USA: The Mathworks Inc.)
- FRAISA Schnittdaten, 1999 (Bellach: FRAISA SA)
- Open CASCADE, 2004, Technical overview 4/2002, www.oencascade.org
- Rybak, H., 2002, Open Source for CAD/CAM. *Digital Engineering* 1/2002 EADS Matra Datavision, www.matra-datavision.de

Gamma, E., Helm, R., Johnson, R., Vlissides, J., 1994, Design Patterns - Elements of Reusable Object-Oriented Software (Addison Wesley)

Figure Caption

figure 1: Life cycle of technological data

figure 2: Abstract architecture of AgentAP

figure 3: Data sources for machining objects

figure 4: Example of a NC document with embedded comments

figure 5: Extraction of STEP-NC based process data

figure 6: Preprocessing of feature-based cutting values with a self-learning module

figure 7: User interface of the self-learning module in the mode of user interaction

figure 8: User interface of an example calculated with ANN

figure 9: Design of self-learning activities for cutting values determination

figure 10: Agent system agentAP


figure 11: Machine assistant with optional information about machine data and operating data

figure 12: Machining assistant with collected information about machined features


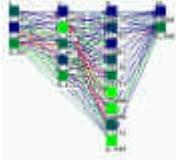

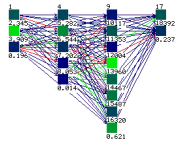

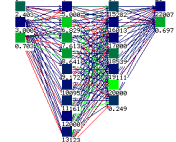
Table Caption

table 1: Results for example through pocket milling

table 2: Results for the example through pocket milling, blind pocket milling and slot milling

 (Source: Fraisa)	Learn algorithm	Error SSE	Network
	Backpropagation	0,00470	3-5-9-2
	Levenberg	0,00320	3-6-10-2
	Bayes	0,01400	3-9-2-2

For Peer Review Only

Feature	Symbol (Source: Fraisa)	Error SSE	Network
Through pocket milling		0,00465	
Blind pocket milling		0,018783	
Slot milling		0,010409	

(Source: SolidWorks, EXAPTsolid)

**Information management -
life cycle of technological data**



Execution: NC machine tool
(Source: Hecker)

STEP
Work plan
NC program



Planning data

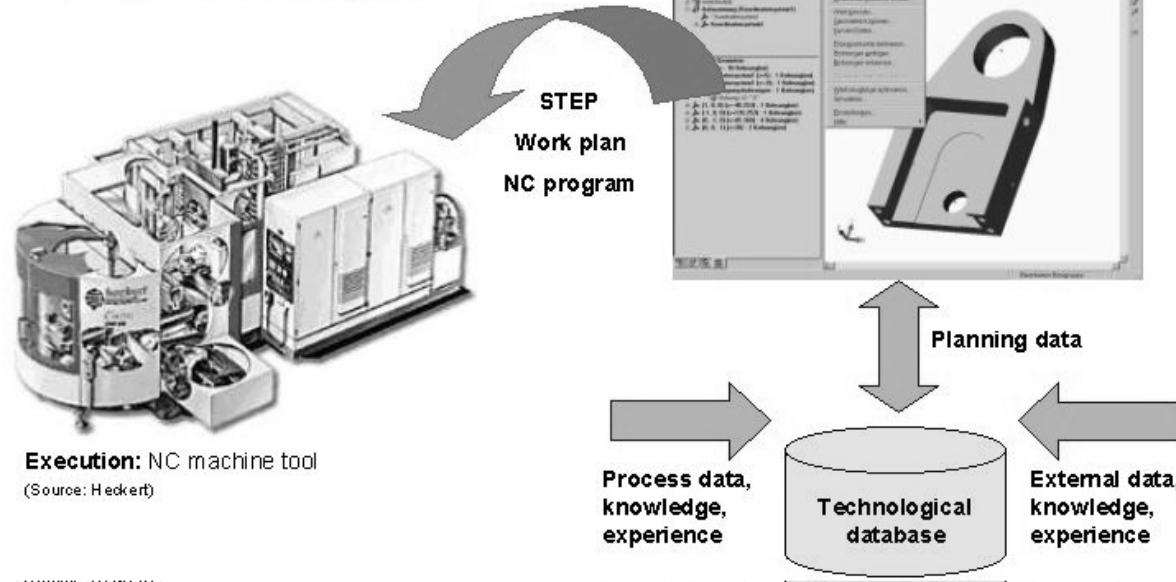


Review Only

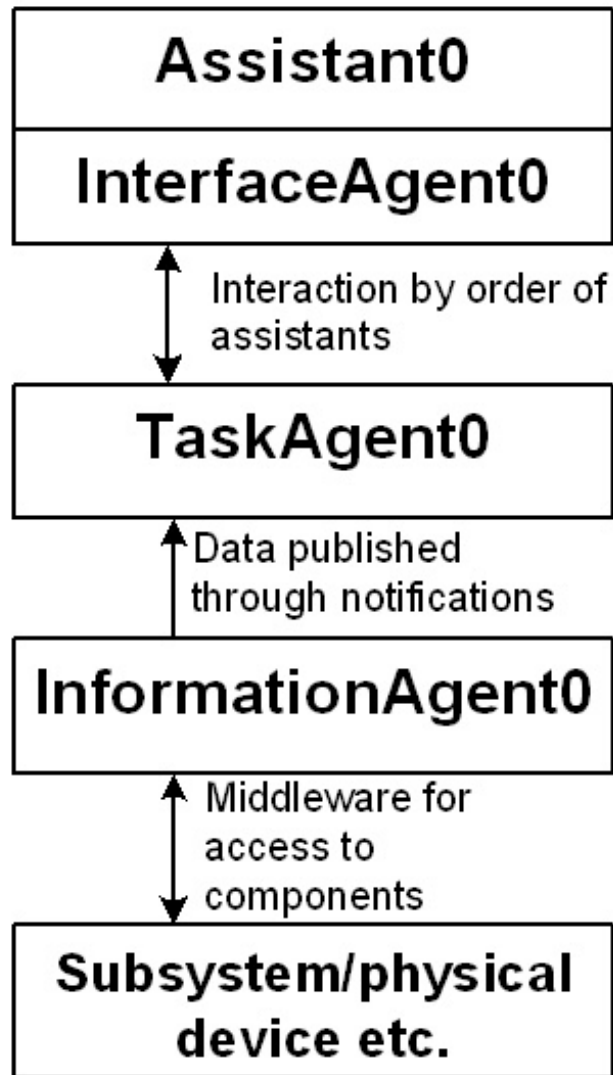
Preparation: Planning system CAD/CAM

(Source: SolidWorks, EXAPTsolid)

Information management - life cycle of technological data



Review Only



Graphical user interface

Communication between assistants and agent system

Task agents for information retrieval

Information agents for information acquisition

Information source which is abstracted by Information agents

Only

Control information**Information source for agents:**

Open interface, API

Dependent on machine control interface**NC code information**

```
%MPF1
...
N0010T514100 L70 R49=0
...
N0020( FRAESWERKZEUG
IDNR.:514100 EINSATZD.: 10.00)
:0025G0G90G54G60S446T0M03
...
N0070G1G94Z0.F74.
N0080Y-7.
N0090Y207.
....
N0670M30
```

Information source for agents:

NC program, code of Sequences

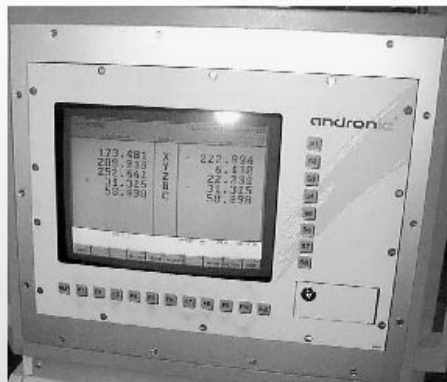
Low level information, vendor specific extensions**STEP - NC information**

```
FILE NAME
...
#10=WORKPLAN('MAIN WORKPLAN'
, (#11, #14, #15), $, #36);
#14=MACHINING_WORKINGSTEP('WS
ROUGH POCKET1'...)
...
#21=BOTTOM_AND_SIDE_ROUGH_MILLING
( ($, $, 'ROUGHPOCKET1', #28, 15.0, $,
#50, #33, #35, $, $, 2.5, 5.0, 1.0, 0.5)
#33=MILLING_TECHNOLOGY ($, .TCP., $,
2.0, $, 0.3, .F., .F., .F.);
#35=MILLING_MACHINE_FUNCTIONS (.T.
, $, $, .F., $, (), .T., $, $, ());
#49=MILLING_CUTTING_TOOL('MILL
4MM', #52, (), (50.0), 80.0, $, $);
...
```

Information source for agents:

Data model, structured part program

High level information, object oriented, independent (neutral)



```
%MPF1
...
N0010T514100 L70 R49=0
...
N0020 ( FRAESWERKZEUG
IDNR.:514100 EINSATZD.: 10.00)
:0025G0G90G54G60S446T0M03
...
N0070G1G94Z0.F74.
N0080Y-7.
N0090Y207.
....
N0670M30
```

```
FILE NAME
...
#10=WORKPLAN('MAIN WORKPLAN'
, (#11, #14, #15), $, #36);
#14=MACHINING_WORKINGSTEP('WS
ROUGH_POCKET1'...)
...
#21=BOTTOM_AND_SIDE_ROUGH_MILLING
( ($, $, 'ROUGHPOCKET1', #28, 15.0, $,
#50, #33, #35, $, $, 2.5, 5.0, 1.0, 0.5)
#33=MILLING_TECHNOLOGY ($, .TCP., $,
2.0, $, 0.3, .F., .F., .F.);
#35=MILLING_MACHINE_FUNCTIONS (.T.
, $, $, .F., $, (), .T., $, $, ());
#49=MILLING_CUTTING_TOOL('MILL
4MM', #52, (), (50.0), 80.0, $, $);
...
```

Information source for agents:

Open interface, API

Dependent on machine control interface

Information source for agents:

NC program, code of Sequences

Low level information, vendor specific extensions

Information source for agents:

Data model, structured part program

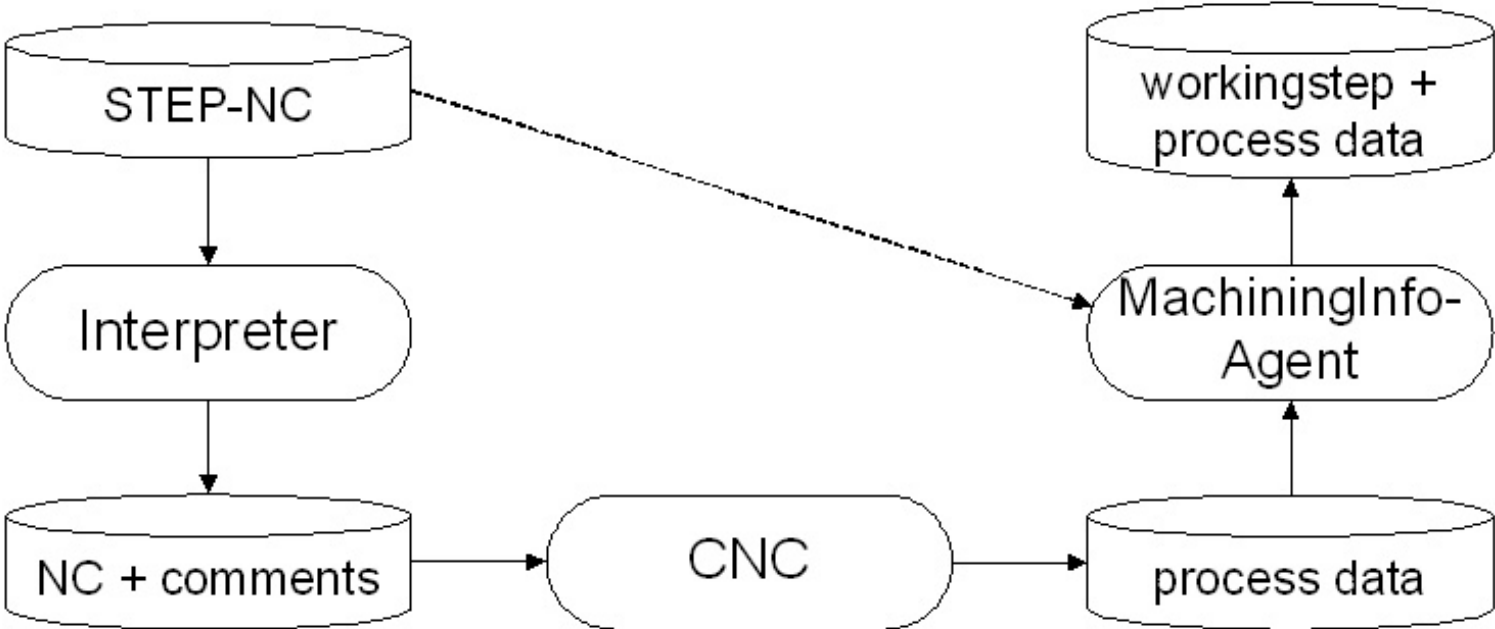
High level information, object oriented, independent (neutral)

Review Only

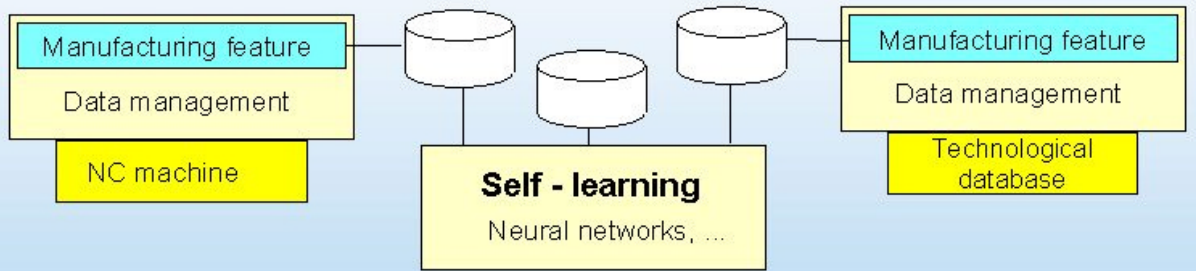
...

```
T3 M6
F1600 S840
G81 X0.0000 Y30.0000 Z30.0000 B10.0000
G79 X20.0000 Y60.0000 Z0.0000
;;#10 = Machining_Workingstep( 'WS DRILL HOLE', #51, #12, #19, $ )
T4 M6
F1600 S1020
G81 X0.0000 Y30.0000 Z30.0000 B10.0000
G79 X20.0000 Y60.0000 Z0.0000
;;#11 = Machining_Workingstep( 'WS ROUGH POCKET', #51, #13, #20, $ )
T12 M6
F1600 S1250
G87 X78.0000 Y48.0000 Z29.5000 R10.0000 B30.0000 K2.5000 I25.0000 J1.0000 D90.0000
G79 X70.0000 Y70.0000 Z0.0000
;;#12 = Machining_Workingstep( 'WS FINISH POCKET', #51, #13, #21, $ )
T12 M6
F1600 S1250
G87 X80.0000 Y50.0000 Z30.0000 R10.0000 B30.0000 K2.0000 I50.0000 J1.0000 D90.0000
G79 X70.0000 Y70.0000 Z0.0000
M30
```

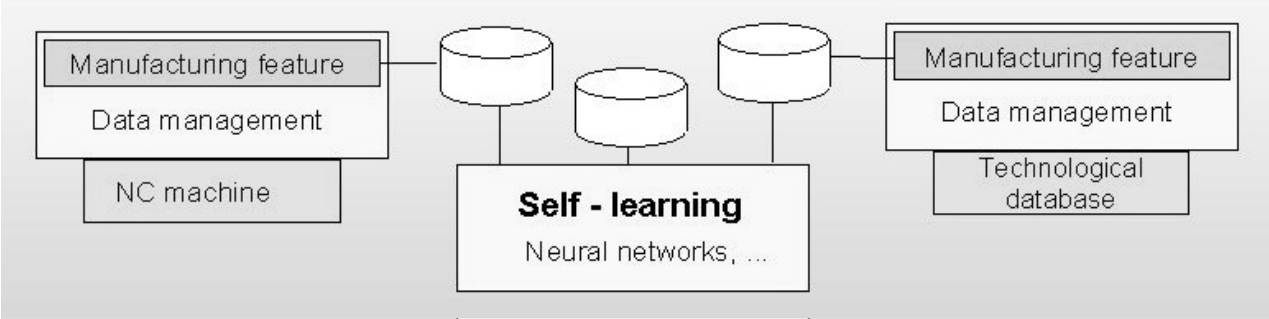
only



Review Only



For Peer Review Only



For Peer Review Only

Learn a neural network

Neural network to learn

Folder
(all neural networks in this folder will be used for learning!)

E:\Temp\Planeta2\Planeta2-4-10-0-2-sc.net

Data files to use

Learn
E:\Temp\Planeta2\LERNDATEN.pat

Validate
E:\Temp\Planeta2\VALIDIERUNGSDATEN.pat

Test
E:\Temp\Planeta2\TESTDATEN.pat

Parameter for learning process

Learn algorithm: Rprop
Init function: Randomize_Wei

Number of cycles for each learn parameter: 7
Number of cycles maximum: 15000

Number of validations: 5
Neural network OK, if SSE less than: 0.4898

Shuffle

Learn a neural network

Neural network to learn

Folder
(all neural networks in this folder will be used for learning!)

E:\Temp\Planeta2\Planeta2-4-10-0-2-sc.net

Data files to use

Learn
E:\Temp\Planeta2\LERNDATEN.pat

Validate
E:\Temp\Planeta2\VALIDIERUNGSDATEN.pat

Test
E:\Temp\Planeta2\TESTDATEN.pat

Parameter for learning process

Learn algorithm: Rprop
Init function: Randomize_Wei

Number of cycles for each learn parameter: 7
Number of cycles maximum: 15000

Number of validations: 5
Neural network OK if SSE less than: 0.4898

Shuffle

Neural Net for Manufacturing Feature: Through Pocket Milling (Fraise Catalogue)

Calculation of feed per tooth and rotation number

Material parameters

Material group

- Fraisa M1 (HRC < 24)
- Fraisa M2 (HRC 24 - 32)
- Fraisa M3 (HRC 32 - 41)

Cutting tool parameters

Cutting tool coated

- yes
- no

Cutting tool diameter

15 mm

Calculation results

	Method batch process	Method genetic algorithm
Revolutions per minute n:	1138.72 1/min	1056.73 1/min
Feed per tooth fz:	0.0411258 mm	0.0414775 mm

Calculate

Exit

Neural Net for Manufacturing Feature: Through Pocket Milling (Fraisa Catalogue)

Calculation of feed per tooth and rotation number

Material parameters

Material group

- Fraisa M1 (HRC < 24)
- Fraisa M2 (HRC 24 - 32)
- Fraisa M3 (HRC 32 - 41)

Cutting tool parameters

Cutting tool coated

- yes
- no

Cutting tool diameter

mm

Calculation results

Method batch process

Method genetic algorithm

Revolutions per minute n: 1/min

1/min

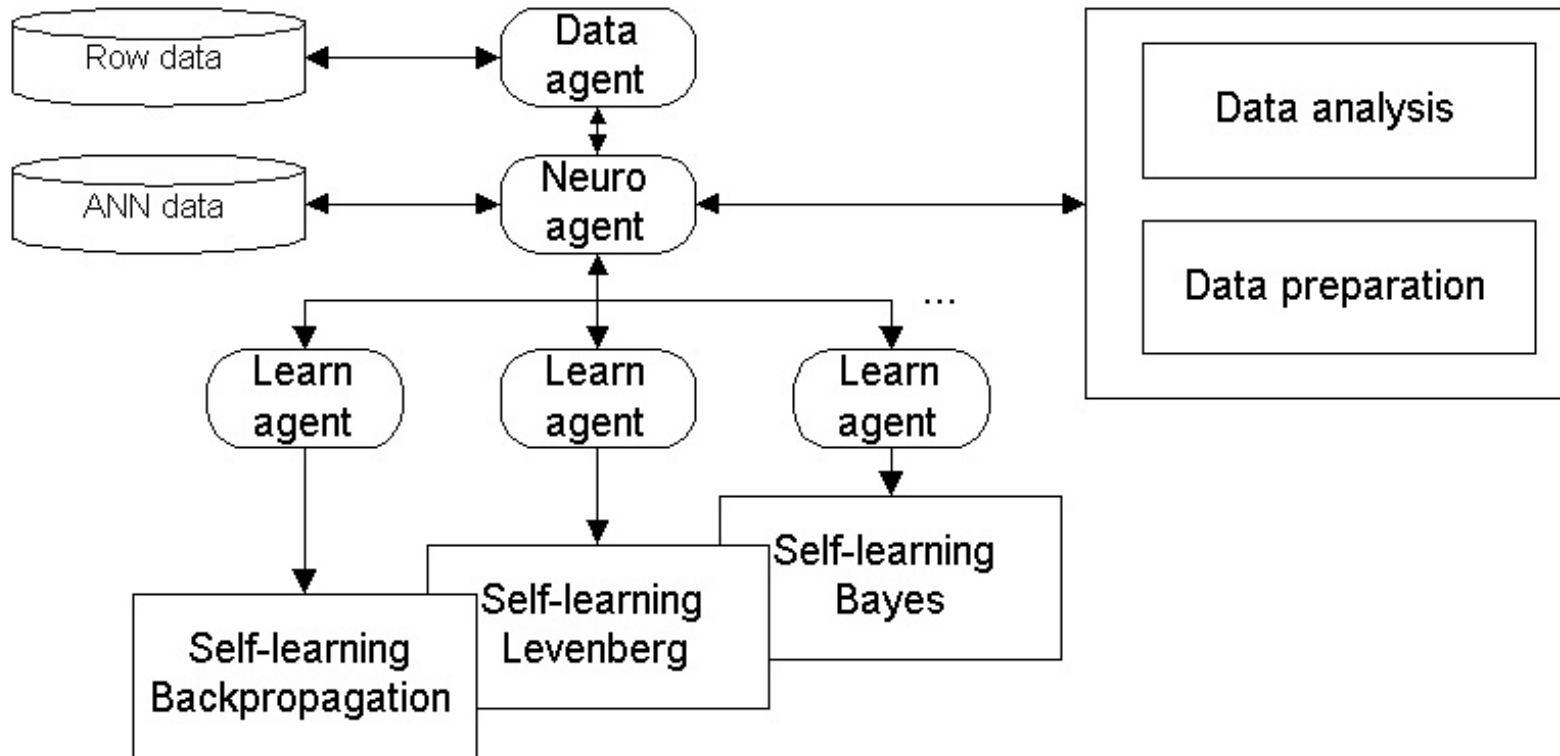
Feed per tooth fz: mm

mm

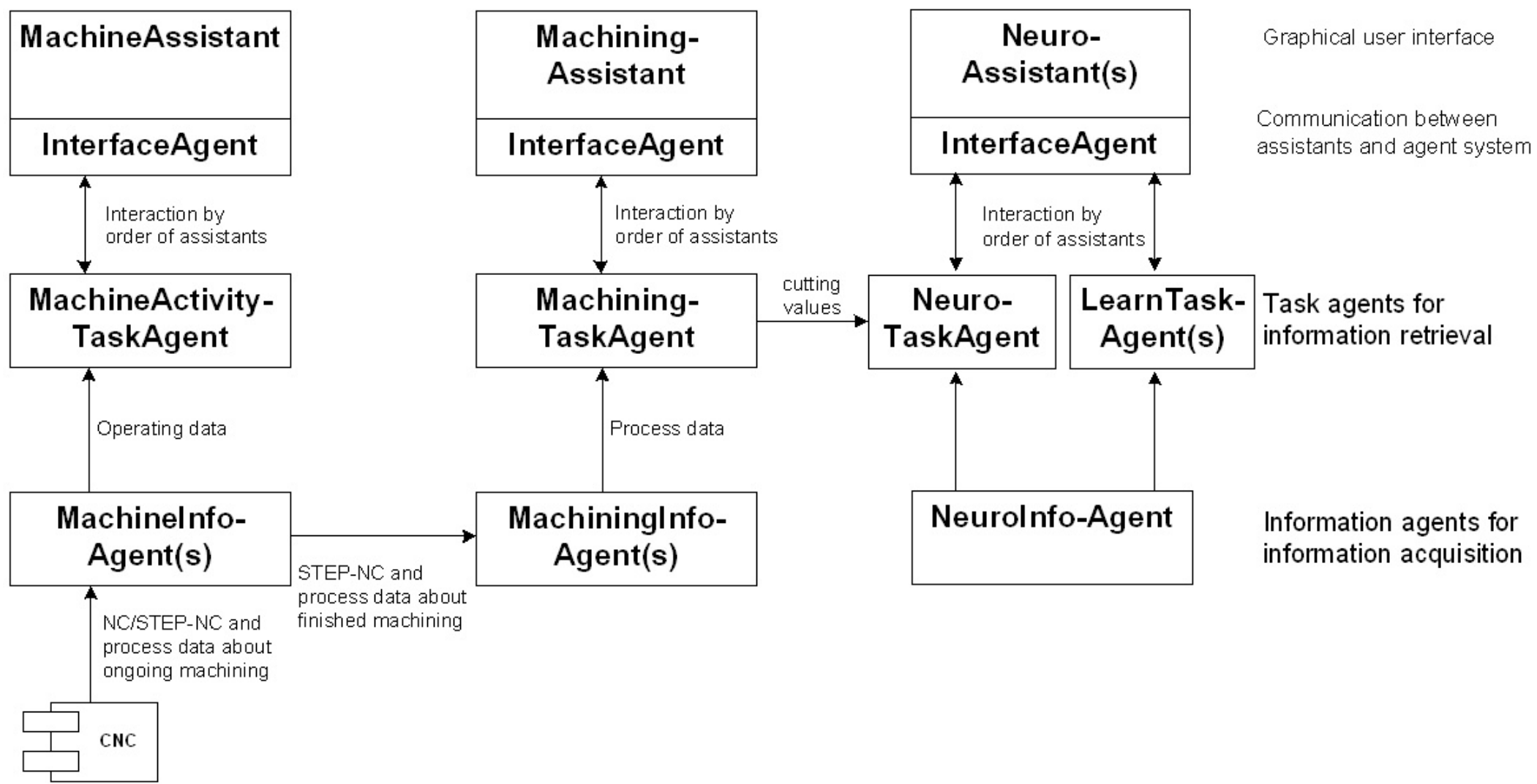
Calculate

Exit

Review Only



Review Only





Review Only



Review Only

MachiningAssistant MachineAssistant X

Machinedata NC Program execution

Add search criteria

Property started Operator between Add

Criteria 14 . 1 . 2005 14 : 29 : 10 14 . 1 . 2005 14 : 29 : 10

Machine starts with maho Rem

and started between 09/20/04 14:28:19 today 14:28:19 Rem

Conduct search

Aggregation [none] Limit 10 Search

Machine	NC-Program	started	completed	duration
maho800	Beispiel2 ...	10/07/04 20:19:47	10/07/04 20:22:23	2,60min
maho800	Beispiel2 ...	10/07/04 20:22:54	10/07/04 20:23:02	0,13min
maho800	Beispiel2 ...	10/07/04 20:24:26	10/07/04 20:24:59	0,54min
maho800	Beispiel2 ...	10/07/04 20:25:31	10/07/04 20:25:46	0,24min

MachiningAssistant MachineAssistant X

Machinedata NC Program execution

Add search criteria

Property started Operator between Add

Criteria 14 . 1 . 2005 14 : 29 : 10 14 . 1 . 2005 14 : 29 : 10

Machine starts with maho Rem

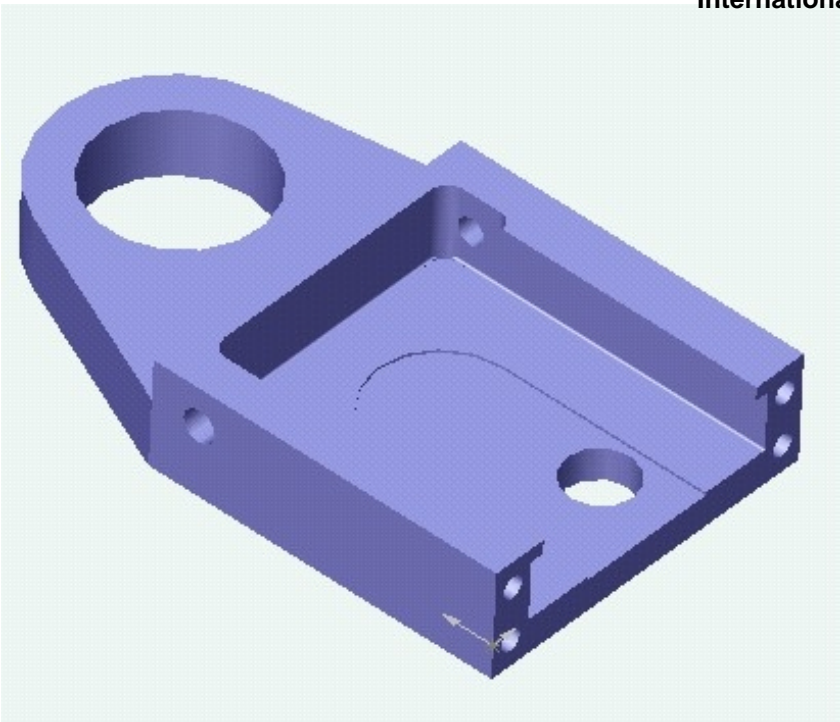
and started between 09/20/04 14:28:19 today 14:28:19 Rem

Conduct search

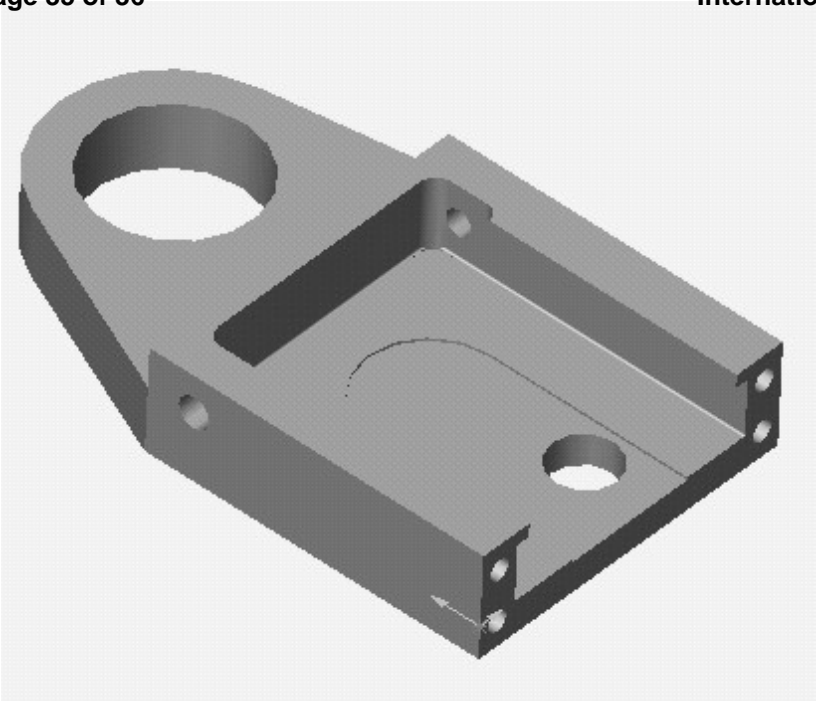
Aggregation [none] Limit 10 Search

Machine	NC-Program	started	completed	duration
maho800	Beispiel2 ...	10/07/04 20:19:47	10/07/04 20:22:23	2,60min
maho800	Beispiel2 ...	10/07/04 20:22:54	10/07/04 20:23:02	0,13min
maho800	Beispiel2 ...	10/07/04 20:24:26	10/07/04 20:24:59	0,54min
maho800	Beispiel2 ...	10/07/04 20:25:31	10/07/04 20:25:46	0,24min

Review Only



er Review Only



Peer Review Only

MachiningAssistant x MachineAssistant

Selection

Machine: all Machining Feature: all

Program: Beispiel1_01 Limit: 100

Refresh

Mac...	Program	Time	Dur...	vsMin	vsMax	Dia	Length	Type	Drill
cwk	Beispi...	09/29...	20,00	169.14334	183.21768	22,00	-30,00	THRO...	20,00
cwk	Beispi...	09/29...	51,00	144.26193	155.00618	22,00	-30,00	THRO...	20,00
cwk	Beispi...	09/29...	20,00	137.79025	144.76459	22,00	-30,00	THRO...	20,00
cwk	Beispi...	10/08...	51,00	133.26636	139.04689	22,00	-30,00	THRO...	20,00
cwk	Beispi...	10/08...	20,00	109.892...	117.3699	22,00	-30,00	THRO...	20,00
cwk	Beispi...	10/08...	51,00	107.81946	113.34866	22,00	-30,00	THRO...	20,00
cwk	Beispi...	10/08...	51,00	74.51858	90.729195	22,00	-30,00	THRO...	20,00
cwk	Beispi...	10/08...	20,00	72.25663	77.15752	22,00	-30,00	THRO...	20,00
cwk	Beispi...	10/08...	51,00	52.338932	66.099106	22,00	-30,00	THRO...	20,00
mh1000	Beispi...	09/29...	52,00	56.737164	64.02566	22,00	-30,00	THRO...	20,00
mh1000	Beispi...	09/29...	20,00	45.678757	58.559288	22,00	-30,00	THRO...	20,00
cwk	Beispi...	10/08...	20,00	42.223003	56.485836	22,00	-30,00	THRO...	20,00
cwk	Beispi...	10/08...	20,00	33.31345	48.93345	22,00	-30,00	THRO...	22,00
mh1000	Beispi...	09/29...	52,00	27.83451	34.746014	22,00	-30,00	THRO...	20,00
mh1000	Beispi...	09/29...	20,00	17.090263	28.588493	22,00	-30,00	THRO...	20,00
cwk	Beispi...	10/08...	51,00	5.906194	23.122122	22,00	-30,00	THRO...	20,00
mh1000	Beispi...	09/29...	20,00	7.187064	20.042362	22,00	-30,00	THRO...	22,00

MachiningAssistant x MachineAssistant

Selection

Machine: all Machining Feature: all

Program: Beispiel1_01 Limit: 100

Refresh

Mac...	Program	Time	Dur...	vsMin	vsMax	Dia	Length	Type	Drill
cwk	Beispi...	09/29...	20,00	169.14334	183.21768	22,00	-30,00	THRO...	20,00
cwk	Beispi...	09/29...	51,00	144.26193	155.00618	22,00	-30,00	THRO...	20,00
cwk	Beispi...	09/29...	20,00	137.79025	144.76459	22,00	-30,00	THRO...	20,00
cwk	Beispi...	10/08...	51,00	133.26636	139.04689	22,00	-30,00	THRO...	20,00
cwk	Beispi...	10/08...	20,00	109.892...	117.3699	22,00	-30,00	THRO...	20,00
cwk	Beispi...	10/08...	51,00	107.81946	113.34866	22,00	-30,00	THRO...	20,00
cwk	Beispi...	10/08...	51,00	74.51858	90.729195	22,00	-30,00	THRO...	20,00
cwk	Beispi...	10/08...	20,00	72.25663	77.15752	22,00	-30,00	THRO...	20,00
cwk	Beispi...	10/08...	51,00	52.338932	66.099106	22,00	-30,00	THRO...	20,00
mh1000	Beispi...	09/29...	52,00	56.737164	64.02566	22,00	-30,00	THRO...	20,00
mh1000	Beispi...	09/29...	20,00	45.678757	58.559288	22,00	-30,00	THRO...	20,00
cwk	Beispi...	10/08...	20,00	42.223003	56.485836	22,00	-30,00	THRO...	20,00
cwk	Beispi...	10/08...	20,00	33.31345	48.93345	22,00	-30,00	THRO...	22,00
mh1000	Beispi...	09/29...	52,00	27.83451	34.746014	22,00	-30,00	THRO...	20,00
mh1000	Beispi...	09/29...	20,00	17.090263	28.588493	22,00	-30,00	THRO...	20,00
cwk	Beispi...	10/08...	51,00	5.906194	23.122122	22,00	-30,00	THRO...	20,00
mh1000	Beispi...	09/29...	20,00	7.187064	20.043262	22,00	-30,00	THRO...	20,00

view Only