APPROACH FOR A RULE BASED SYSTEM FOR CAPTURING AND USAGE OF KNOWLEDGE IN THE MANUFACTURING INDUSTRY

Supporting manufacturing machines tasks with knowledge based services

Jivka Ovtcharova, Alexander Mahl, Robert Krikler
Institute for Applied Computer Science in Mechanical Engineering RPK, Universiy of Karlsruhe (TH), Kaiserstr. 12, D-76131 Karlsruhe, Germany, Email:ovtcharova@rpk.mach.uni-karlsruhe.de

Abstract: The support of manufacturing machines with IT solutions becomes state-of-the art in the production process. For this reason the European Chinese research project KoBaS (Knowledge Based Customized Services for Traditional Manufacturing Sectors Provided by a Network of High Tech SMEs) aims to create customized services for manufacturing machines providing a set of tools including advanced task and process planning machine configuration, maintenance, training and management support. This paper describes a software application for capturing and re-using rule based knowledge concerning manufacturing machine services listed before. The system is designed to support different kinds of manufacturing machines and manufacturing machine specific domains and furthermore to support an open and easy to implement data exchange and communication process.

Key words: Rule Based System, Knowledge Engineering, System Integration, Intelligent Manufacturing Machines

1. INTRODUCTION

Today the manufacturing companies have to cope with the increasing pressure from global marketplace. To compete successfully many industrial manufactured products need to be changed as described in [1], especially those that have common requirements for increased product customization

and improved competitiveness in terms of reduced cost, shorter delivery times and improved quality [2]. To fulfill the requirements coming from the market, the manufacturing industry needs a breakthrough regarding the current practices in the use of manufacturing machines.

In this context the European Chinese research project KoBaS (Knowledge Based Customized Services for Traditional Manufacturing Sectors Provided by a Network of High Tech SMEs) aims to support the manufacturing industry with a set of innovative software tools. These novel tools facilitate on one hand, an advanced task and process planning and, on the other hand, machine configuration, maintenance, training and management [2] for the manufacturing machine.
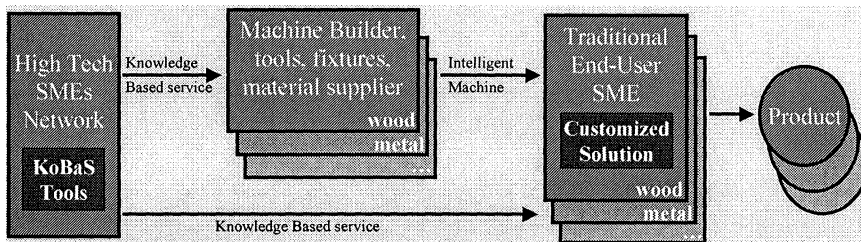

Figure 1. Schema of the KoBaS solution [2]

Since two manufacturing machines may be significant different in essential parameters like spindle RPM[1], maximal pressure, cutting speed etc. - even within one domain (wood, metal, plastic, stone etc.) - there is the need to provide a customized solution for each machine. The network of High Tech SMEs (to be established out of the KoBaS consortium) will install a customised solution of the software tools which take into account the requirements of the machine. In order to succeed in this task it is necessary to *re-use the knowledge* gained from former installations as well as to provide a *flexible, adaptable solution* which matches to the needs of the specific machine (and company). Thus the approach is to represent the knowledge using a rule based system which controls the flexible (customizable) part of the application logic of the software tools. This approach prevents to change/adapt the implementation of the software tools for each manufacturing machine.

Accordingly this paper focuses on the rule based system which builds the flexible knowledge core of the customized solution. The decisions of the rule based system on runtime have to take into account data from external IT-

---

[1] RPM: Revolution Per Minute

systems (e.g. PDM[2] or PPC[3] system). For example in order to provide a proper task planning the current status of the machine needs to be taken into account. Thus the rule based system must be flexible enough to be embedded into an existing heterogeneous system environment. Beside the description of the rule based system the paper will focus on an ontology based method for efficient realization of the setup phase of the customized solution. Within the setup phase the rule based system needs to adapt to the requirements coming from the involved IT-systems (e.g. data models).

2.   RULE BASED SYSTEM WITH OPEN ARCHITECTURE TO SUPPORT MANUFACTURING MACHINE TASKS

The rule based system presented in this paper uses an object oriented approach for defining rules. Existing object oriented rule based systems like JESS[4][3], JRules[4], QuickRules[5], drools[6], JEOPS[5][7] or Rules4J[8] provide mechanisms to manage application logic of a software application. The integrated approach presented in this paper allows executing and defining rules on data objects (e.g. PDM data) as well as application specific objects. Thus the knowledge represented by rules could be used and applied on data and application objects.

Hence it is necessary to deal with several data models simultaneously, whereas the data models may change during runtime. Consequently the rule base operates on meta data model level. The paper will present the meta data model approach as well as the methodology of object oriented rule specification and execution within a heterogeneous software environment dealing with different data models in parallel.

2.1   Object oriented approach for rule representation

As stated before the paradigm of object oriented rule representation is implemented within several object oriented rule base engines. The prototype of the rule based system provides a generic rule description which may use any object oriented rule engine in its kernel. A rule base is defined through
- a data model,
- a set of rules and

---

[2] PDM-system: Product Data Management system
[3] PPC-system: Product Planning Control system
[4] JESS: Java Expert System Shell
[5] JEOPS: Java Embedded Object Production System

    •   instance data elements (e.g. objects, relations).

The set of rules as well as the instance data needs to be accordingly to the specified data model. Thus in case an object type will be deleted all instances (objects) will be deleted as well. Based on a request send to the rule base none, one or more rule(s) of the set of rules may be applied (this is called a rule "fires" [3]). The result will be send back.

    Each rule consists of three sections: *declaration* of objects used within the rule, *condition* of the rule and *conclusion* (output of the rule). The condition is a logic expression (containing one or more logic operations (and, or etc.)) which decides whether the rule is applicable or not. The conclusion consists of one or more actions whereas an action is to create/modify an object, create/remove a relationship.

    This paper introduces an exemplary a rule which may be used by the part program creation module (PCM) and access instance data of other IT-systems. The task of the part program creation module is to generate a CNC[6] program for a part to be manufactured on the machine. The knowledge how to generate the CNC code is modelled using the rule base. This will be done by specifying parameters which will later be used by the CNC code generator algorithm of the part program creation module. In case the rule "fires" there is a request with a part which fulfils the specified condition. The request refers to a part of a product structure administrated in the PDM system. The expected response of a request will be the information stating the possibility to manufacture the part on the machine and describing the CNC-program code which will be used by the part program creation module.

Declaration

RequestPart	r;	

Part	part;	(PDM system)
UsedMaterialRel	R1;	(PDM system)
Material	mat;	(PDM system)
Machine	turning;	(PPC system)

CNCProgram	prog;	(PPC module)
Manufacturable	p;	

Condition

```
If ((r.number == part.number) &&
    (part.Name startsWith("spindle-4")) &&
    (mat.Number=="al54758") &&
    (part R1 mat) &&
    (part.lengthOfBoundaryBox < m.MaxLength))
```

Conclusion

```
// create new object instances
p = new Manufacturable(part,m));
prog = new CNCProgram("T-056",...);
```

---

[6] CNC: computerized numerical control

Figure 2. Object oriented rule specification

Figure 2 presents an example for a Java based object oriented rule representation. The declaration section defines the objects which occur in the condition or conclusion section. The condition section is an (complex) if statement: The first "and-clause" searches for the requested part number in the rule base. If no part object with the specified requested number was found the rule will not fire. Otherwise it will checked, if the name of the part starts with "spindle-4", the specified material for the part has got the Id=$"$a154758$"$ and the Boundary box of the part is smaller than the maximum length of the working space specified in the machine.

In case the rule fires the (requested) part could be manufactured by this machine. This will be specified by the creation of a new instance of a "Manufacturable"-object. Furthermore the CNC-program will be specified using some internal parameters (e.g. CNC-module-Id=$"$Turning-056$"$ etc.). The rule will be applied in case the request send to the rule base contains a request part object which refers to a part object that fulfils the specified conditions.

2.2   Approach for generic data model description

To enable the rule based system to be embedded into an existing heterogeneous IT-system environment and adapted to the needs of a manufacturing machine, the rule based system needs to handle any data model. Hence the operations on instance data must be parameterized by the data model of the customized solution but act on meta model elements. A meta model describes the elements which are used to build a model. In case of an (object oriented) data model elements can be: object type, relationship type and attribute type as well as instance elements like object, attribute and relationship. The API[7] of the rule based system provides functionality for creating and manipulating these elements. Thus it is possible to modify the data model on runtime which will also effects the objects (internal facts) of the rule base. In case of deleting an object type all instances consequently will be deleted as well.

This approach bases on the results of the former European Chinese research project DRAGON [9] which was successfully finalized by RPK. The application case within DRAGON was limited to PDM data and didn't take in account (rule based) reasoning mechanisms nor object oriented rule specification. The approach in this paper enhances the DRAGON approach additionally by allowing the simultaneous work with several data models.

---

[7] Application Programming Interface

2.3   Exchange of instance data

Each rule based system contains one or more rule bases whereas each rule base consists of a data model, set of rules and instance data. The set of rules and instance data base on the data model. Thus each change of the data model might affect the set of rules and instance data. Modifications of the data model result in changes of the instance data (see 2.2). In order to avoid inconsistent rules a data model element must not be modified or deleted while it is used within the specification of the rule. This action is refused by the rule based system unless the rule is removed or modified first.
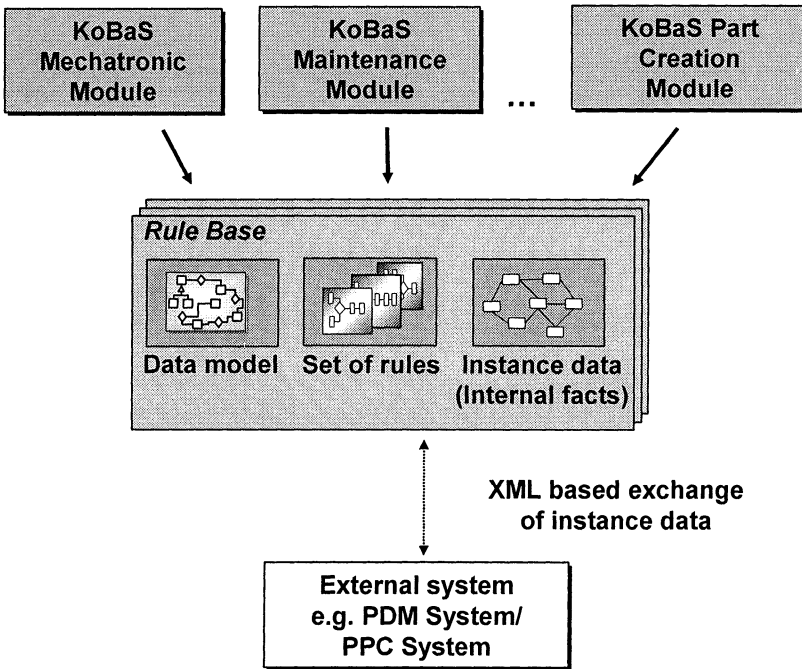

Figure 3. Exchange and access of instance data from heterogeneous data sources

The rule based system presented in this paper allows application to execute rules on instance data of application specific data objects and external systems like PDM or PPC systems (see *Figure 3*, furthermore see example presented in 2.1). Hence it is possible to use and modify product data within the execution of a rule.

The exchange of instance data is XML based whereas the correspondent XML document bases on a general description of instance data using the meta data model elements. E.g. each object has to specify its object type etc.

The instance data will be checked on runtime if the description is conform with the current data model of the rule base.

2.4   Setup phase of rule based system

The outcome of the whole KoBaS project will be a set of software modules which build the foundation for an intelligent manufacturing machine (see chapter 1). Within the setup phase these modules need to be adapted to the specific requirements of the manufacturing machine. Manufacturing machines of different domains fulfill different tasks and differ in design, operation mode and measurements. Thus the data model (and rules) for describing the machines will be different within machines of different domains. Within the research project the domains woodwork and metal are focused due to the presence of end-users while the approach is not limited to these domains. After project end it is expected to introduce the KoBaS solutions for various domains of manufacturing machines e.g. plastic, molding etc.
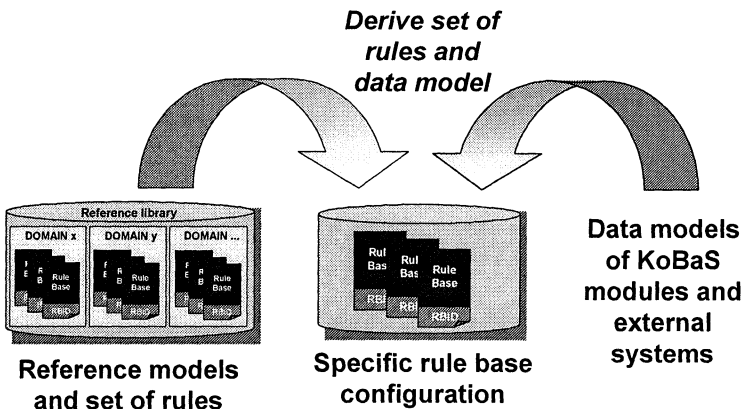

Figure 4. Exchange and access of instance data from heterogeneous data sources

In order to re-use the gained knowledge from former projects (installation of KoBaS software for other machines and enterprises) and speedup the setup phase a *reference library* for each domain will be created and incremental extended (e.g. after each KoBaS solution). This reference library contains for each domain one or more rule bases whereas (in general) one rule base is assigned to one KoBaS application.

Within the setup phase for each KoBaS application (e.g. part program creation module) the rule base (data model and set of rules) needs to be specified according to the manufacturing machine and enterprise specific

processes. Therefore several different data models needs to be taken into account (see *Figure 4*): the reference data models (from reference library), data models of KoBaS applications (software modules) and data model(s) of external system(s) (e.g. PDM-system(s)) which manage(s) the data within an enterprise. The rule based system provides a *graphical rule base editor* which supports the setup process by providing views of several rule bases (and hence data models and instance data) as well as functionality for specifying the new data model based on these models. The rule based system provides an XMI[8][10] interface which allows importing the different data model specifications. Thus the data models may be specified using any XMI supporting UML tool like Borland Together, Enterprise Architect, Omondo or Rational Rose.

The specification of a rule rest upon on data model elements like object types, relationship types or attributes (see 2.1). Thus the reuse of existing rules is only possible if these elements exist within the new data model or if it is described which elements they correspond to. In order to specify the data model dependencies the rule base editor allows specifying mapping rules between data models.

The re-use of data model elements does not imply the re-use of the rules. This is only valid if the model elements are used with the same *semantics*. Therefore we will present an *ontology based method* which allows defining the semantic interpretation of the data model elements. An ontology is a formal, explicit specification of a shared conceptualization [11] and hence defines the semantic within a particular domain. The meta model of an ontology contains *concepts* and *semantic relations* between these concepts. A more formal definition could be found in [12]: *"An ontology is a tuple $\Omega := (C, is\_a, R, \psi)$, whereas C is a set of concepts, „is_a" the inheritance relation on C, R a set of relationship names and $\psi : R \rightarrow \wp(C \times C)$ a function".*

While an object is assigned to exactly one object type, it could be linked to several concepts of the ontology. The link between object and concept describes the semantic interpretation of the object. Consequently the semantic (semantically interpretation) of an object type could be modeled by linking an object type with one or more concepts. It is possible to model synonyms translations of a concept. Thus the ontology based approach addresses the non-uniform naming of data model elements (synonyms, multi-lingual).

Figure 5 shows the necessary four steps within the setup phase. After the initial definition/update of the ontology the semantic interpretation of the *relevant* data model elements of the external system(s), KoBaS applications

---

[8] XMI: XML Metadata interchange (see: [10])

and reference library needs to be defined. The semantic interpretation of the reference data model will initially be defined and updated within each incremental update of the reference library. Therefore this step is limited to the data model of the external systems and KoBaS application. The last two steps are the derivation of the data model and initial set of rules. In a later stage of the implementation this could be done semi-automatically by using ontologies and data model matching algorithm of artificial intelligence.

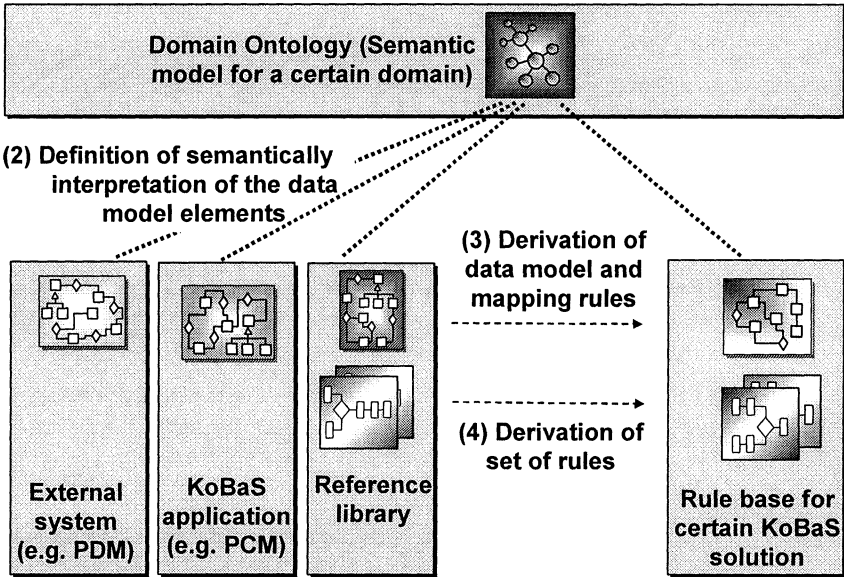**(1) Definition / update of the semantic model (ontology)**

Figure 5. Setup phase of a rule base for a certain solution (machine)

3.   CONCLUSION

In this paper the authors described a framework to create customized rule based system solutions in an object oriented way. The presented tool enhances the efficiency of the implementation, setup and management activities of rule based systems related to services for manufacturing machines. Moreover the advanced (ontology based) data integration utility of the system allows using distributed data and application objects during the rule definition (setup phase) and runtime (rule execution) phase of the

customized solution. The open API (Application Programming Interface) of every customized solution allows in one hand the runtime management of the available rule bases, e.g. creation of new rule sets or rules, in other hand the deployment of new rule bases. Thus the software applications interfaced to the customized rule bases solution uses up-to-date data objects and knowledge. In order to proof the approach it will be validated within real industrial cases (manufacturing machines) in the KoBaS project.

4.   REFERENCES

1. M. Rigamonti, T. Tolio; M. Tomasella, A Valente: Production requirements definition in a dynamic environment, CARV 2005, International Conference on Changeable, Agile, Reconfigurable and Virtual Production, Munich, Germany
2. KoBaS (Knowledge Based Customized Services for Traditional Manufacturing Sectors Provided by a Network of High Tech SMEs) project proposal, (2004)
3. E. Friedman-Hill: JESS in Action. Manning Publications Co., USA, 2003
4. N.N, JRules-Homepage: http://www.ilog.com, online at 12th January (2006)
5. QuickRules-Homepage: http://www.yasutech.com, online at 10th January (2006).
6. drools-Homepage: http://drools.org, online at 10th Januray (2006).
7. JEOPS-Homepage: http:// www.di.ufpe.br/~jeops/, online at 08th January (2006)
8. Rules4J-Homepage: www.rules4j.com, online at 08th January (2006)
9. A. Mahl, O. Hornberg, A. Ehrler : Concept and industrial use of platform for flexible cross enterprise collaboration, in: *product data journal* Vol. 2/11, (2004)
10. XMI-Homepage: http://www.omg.org/technology/documents/formal/xmi.htm
11. T. R. Gruber: A translation approach to portable ontology specifications. *Knowledge Acquisition.* Vol. 5. (1993).
12. Andreas Faatz, Ralf Steinmetz: *Ontology Enrichment with Texts from the WWW.* In Semantic Web Mining, WS02, Helsinki, Finland. (2002)