Development of a Software Application for Machine Tool Reconfiguration using a Knowledge-Based Engineering System Approach

Mauricio Hincapié
Dirección de Investigación, Institución Universitaria Salazar y Herrera
CR 70 No 52-49, Medellin, Colombia
Tel.: +574 448 87 02
E-mail: e.hincapie@iush.edu.co

David Guemes
Centro de Calidad y Manufactura, Instituto Tecnológico y de Estudios Superiores de Monterrey
Ave. Eugenio Garza Sada 2501 Sur, Col. Tecnológico C.P. 64849, Monterrey, N.L., México
Tel.: +52 81 8158 2000
E-mail: guemes@itesm.mx

Manuel Contero
Escuela Técnica Superior de Ingenieros Industriales, Universidad Politécnica de Valencia
Camino de Vera - Edificio 5F, 46022 Valencia, CA, España
Tel.: +34 963 87 71 70
E-mail: mcontero@upv.es

Miguel Ramírez
Departamento de Mecatrónica y Automatización, Instituto Tecnológico y de Estudios Superiores de Monterrey
Calle del Puente 222, Tlalpan, Ejidos de Huipulco, 14380 Ciudad de México, D.F., México
Tel.: +52 55 5483 1647
E-mail: miguel.ramirez@itesm.mx

Christian Diaz.
Dirección de Investigación, Institución Universitaria Salazar y Herrera
CR 70 No 52-49, Medellin, Colombia
Tel.: +574 448 87 02
E-mail: c.diaz@iush.edu.co

Abstract

The automation processes industry has become increasingly expensive, which is why some small and medium sized enterprises are incapable of buying machine tools with automatic systems. This means that their processes are manual in many cases, and as a result they often have to rework their developed products due to the lack of precision and efficiency in their production processes. Considering that current manufacturing systems with variable machining and turning centers are gradually replacing dedicated systems for medium lot size production, the production systems' basic element, the machine tool, must be capable of working at high speeds with precision, and it must be reconfigurable. These systems must also be compatible and convertible in order to create economic benefits for customers. This article describes a specific software architecture designed to record all the data, information and knowledge concerning manufacturing systems. The software allows for the creation of a new knowledge database and works with it in the reconfiguration of machine tools depending on the rules, requirements and parameters needed to effectively modify production processes or products.

Keywords: Reconfiguration Management System, Machine Tool Reconfiguration, Unified Modeling Language

1. Introduction

In emerging countries, there is a high percentage of small and medium sized enterprises [1]. They employ conventional machine tools because the price of buying and utilizing automated machine tools remains too high. This is why the manufacturing industry has not invested heavily in retrofitted machine tools [2].

Moreover, in the last few years these countries' manufacturing industries have increased their economic activity, showing increased market participation. In 2012 the global machine tool production market was estimated to be worth $94.3 billion, a 30% increase from the 2010 figure of $66.3 billion [3].

Over time, the automation processes industry has become increasingly expensive [15]. Some small and medium sized enterprises are unable to buy machine tools with automatic systems. This means that their processes are manual in many cases, and as a result they often have to rework their developed products due to the lack of precision and efficiency in their production processes [20].

Computer Numerical Control (CNC) machine tool systems, which feature high levels of flexibility and versatility to deal with dynamic changes in production volume and part variation, are highly sought after in emerging countries. Traditional dedicated production lines, such as flexible transfer lines, cannot efficiently adapt to the nature of changing parts and fluctuating lot sizes. Manufacturing systems with variable machining and turning centers are gradually replacing dedicated systems for medium lot size production. This requires the production systems' basic element, the machine tool, to exhibit high speed, precision, and to be reconfigurable, compatible, and convertible in order to create economic benefits for customers [4].

Recently, Reconfiguration Management Systems (RMS) have been proposed to provide the flexibility required by new market conditions. Studies conclude that reconfigurable manufacturing is one of the six grand challenges of the future of manufacturing [5] [6]. These trends highlight the importance of proposing strategies like those described in this article for making the manufacturing industries in emerging countries competitive.

This article offers small and medium sized enterprises (SMEs) a technological alternative that allows them to incorporate a CNC system into the traditional machine tools that have performed well under manual control. With the proposed software, a SME can become more competitive without absorbing the high prices of the CNC machinery acquisition. A specific software architecture was designed to record all the data, information and knowledge about the manufacturing system. The software allows for the creation of a new knowledge database and works with it for the reconfiguration of machine tools, depending on the rules, requirements and parameters needed to effectively alter the production processes or products. The development of the software architecture was supported by the use of Unified Modeling Language (UML).

This article is structured as follows: Section 2 reports on similar projects and opportunities for research, and introduces the universal numeric control project in which this work was developed; Sections 3 to 5 describe the solution proposed for the fast reconfiguration of machine tools using the KBES methodology; Section 4 shows the results; and Section 5 states conclusions and proposes future work.

2. Related Works

This section describes some works related with topics such as Reconfigurable Manufacturing Systems (RMS), Reconfigurable Machine Tools (RMTs) and Knowledge and Modeling for Manufacturing. Later, the Universal Numeric Control (UNC) project is introduced. This project was developed by the Monterrey Institute of Technology and Higher Education (ITESM in Spanish) and is part of the application described in this work.

2.1. RMS, RMTs, and Knowledge and Modeling for Manufacturing

The term RMS was introduced in the mid-1990s as a cost-effective answer to responsiveness and customization demands in the market. RMS has its origin in computer science in which reconfigurable computing systems try to cope with the inefficiencies caused by the unchanging hardware structures and software logic of conventional systems. In fact, the reconfiguration of the artifact usually requires two steps: the rearranging of components and their reprogramming. Other existing concepts that relate to reconfiguration are manufacturing paradigms like: Dedicated Manufacturing Systems (DMS), Lean Manufacturing, Cell Manufacturing Systems (CMS), Group Technologies, Virtual Cellular Manufacturing and Flexible Manufacturing Systems (FMS) [7], [8], [9], [10], [11].

Reconfigurable systems employ reconfigurable assembly lines, increasing the capacity of production through replication and modularity. They are modular semiautomatic systems composed of manual workstations with robots and other resources [12] [13]. The basic building blocks of a RMS are the Reconfigurable Machine Tools (RMTs). A RMT is designed to be customized to produce the desired product or product mix in the required quantities. RMTs are tailored to the initial operation requirements and when operation requirements change, RMTs may be cost-effectively converted and customized for the new requirements. Consequently, controllers for RMTs must be based on the concept of open-architecture [14] [15]. When referring to open-architecture control, the software architecture is modular, thus hardware components and software components can be easily added or removed and the controller can be cost-effectively reconfigured. RMTs are designed for a specific range of operation requirements and as a result, do not have wasted resources and functionality. It is important that the RMT mechanical structure is designed in such way that the machine tool's geometric errors will not compromise quality. However, two additional considerations must be taken into account. First, since RMTs are designed for a range of operation requirements, restraining aspects of tolerance requirements will dictate the geometric error requirements of the machine tool. Second, the structure of a RMT may need to be reconfigured. Therefore, for some applications, RMTs will require mechanical adapters that allow for the quick and accurate addition or deletion of mechanical modules.

Reconfigurable Manufacturing Systems (RMS), Reconfigurable Machine Tools (RMTs), and Knowledge and Modeling for Manufacturing, models and architectures are areas related to the main idea of the proposed model, however some differences are proposed (Table 1):

- A reconfiguration of the machine tool considering their components.
- The creation of rules based on the parts of the machine tool.
- The acquisition of knowledge for reconfiguration of machine tools.
- The application of the manufacturing model based on the design of the machine tool, including the use of software tools as CAD to accomplish the architecture for reconfiguration manufacturing system.

		Subjects						
		Knowledge	**Modelling**	**Ontology**	**RMS**	**RMT**	**Flexibility**	**Product Development**
References	**Molina and Bell 1999 [16]**						X	
	Mehrabi et al. [17]				X			
	Reuven and Yong-Mo 2000 [18]					X		
	Oldknow and Yellowley 2000 [6]		X		X		X	
	Landers et al. 2001 [19]	X		X			X	
	Szykman et al. 2001 [20]				X		X	
	Abdullah et al. 2001 [21]	X			X		X	X
	Merdan et al. 2005 [22]		X		X		X	
	Hakasson and Hartung 2006 [23]				X	X	X	
	Malec et al. 2007 [24]				X	X	X	
	Gurra-Zubiaga and Young 2008 [25]			X	X	X		
	Jardim – Goncalves et al. 2011 [26]			X	X	X		
	Rena 2010 [27]			X	X	X		
	Alsafi and Vyatkin 2010 [28]			X	X	X		
	Hincapié et al. 2014 [34]	X	X	X	X	X	X	X

Table 1: Literature Research on RMS, RMT and Knowledge and Modeling for Manufacturing

Several works have described how Information and knowledge structure aid to support manufacturing decisions, and how this knowledge can be modeled with the UML tool. The design diagrams included in UML aid to construct the software that supports any decision process [21]. Besides UML demonstrates how to acquire the knowledge and integrate it within this technological tool, configuration of manufacture system as well a machine tool.

UML also explains why RMTs are designed for customized range of operation requirements and may be cost-effectively for rapid adjustment of production capacity and functionality in response to new market conditions. Other works have proposed the use of software agents to solve the reconfiguration problem, similar as it is proposed in the described work [28]. However, most of the approaches described in the state of the art were considered to aid to integrate and develop the proposed model and sustain the main idea of the machine tool reconfiguration based on its components and employing KBES methodology to obtain, elicit, structure and manage the knowledge around the reconfigurable machine tool [19] [22] [27] [28].

2.2. Universal Numeric Control

When CNC machines appeared, manufacturing companies opted for generating self-developed codes with the objective of protecting their market. This led to the creation of the UNC, or unified alternative of the code used in the CNC machines. UNC makes the integration of the CNC language with conventional machine tools possible.

UNC is an open architecture based on low-cost software, created to reconvert machines in the manufacturing sector, such as turning machines, milling machines, drillers, cutters, punchers, etc., that do not have automatic control devices [29]. This architecture was proposed in the ITESM as a response to the detected need of the manufacturing sector in Mexico, since there are many companies that own machines to do different types of mechanical operations. The UNC project has several different development paths, such as interpolators, axis control and the human-machine interface, among others. Each one of them is a software module that is supported by one reference architecture. In order for the UNC to accomplish real-time functionality, QNX Neutrino was selected as its operative system.

The UNC architecture consists mainly of the hardware system and the software system. The hardware system is composed of the electronic control devices that comprise the controller (motor drivers, sensors, etc.), the circuit boards and peripheral devices of the PC, as well as the cards inserted into the PC buses. The software system, on the other hand, is composed of three internal layers: the OS layer, which allows for the efficient management of the computer resources; the internal communication layer which links the functional units so the design of the application modules is independent from the hardware; and the Application Programming Interface (API), which is a set of functions defined by the system integrator. The API makes it possible for software application programmers to develop their system modules with the required specifications to reach the desired functionality [30].

The proposed UNC architecture (Figure 1) has a modular software structure to allow for a division of the CNC functions and to give flexibility to the controller platform to handle a variety of tool machines like: turning machines, milling machines, drillers, cutters, punchers, and others. These machines in turn use different electric components, such as electric actuators (AC and DC), and different sensorial systems, gears, etc.

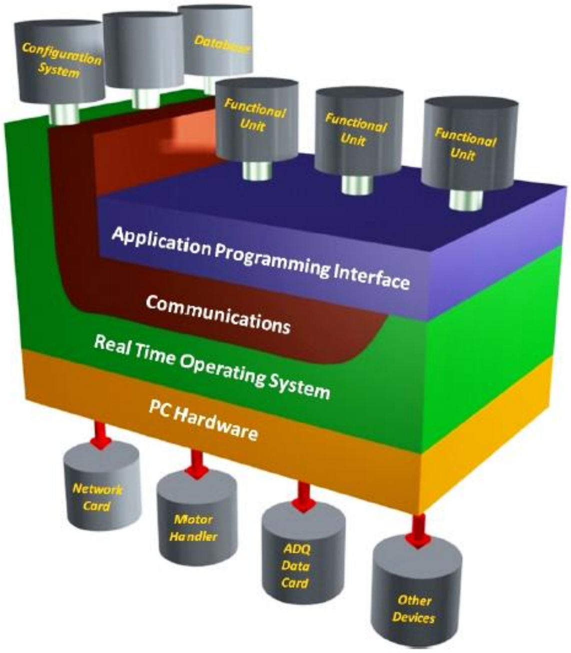


Figure 1: Universal Numeric Control Architecture.

In this article we describe the development of a software$_{240}$ application that allows for the reconfiguration of a manual machine tool to make it automatic. The software was developed applying the methodology described in [31] and using a knowledge-based engineering system approach.

3. Description of the mechanical tool: Turning tool

The turning process is the most common process used to reduce mass. It is employed to manufacture all types of cylindrical forms, reducing material with a cutting tool. Generally, lathes receive energy from an electric motor, which provides the rotary movement necessary to move and advance the tool [32].

The lathe is a machine that removes material by rotating the work against a cutter. Parts to be worked with can be held between centers, attached to a face plate, supported in a jaw chuck, or held in a draw-in collet chuck. Though this machine is particularly adapted to cylindrical work, it may also be used for many other purposes. Plain $_{210}$ surfaces can be obtained by supporting the work on a face plate or in a chuck.

The lathe is probably the oldest of all the machine tools as well as the most important machine in modern production. The size of a lathe is expressed in terms of the diameter of the piece it will swing; thus a 16-inch lathe is one having sufficient clearance over the bed rails to take pieces to be worked with which are 16 inches in diameter.

Numerically controlled turning machines make use of two kinds of tooling systems with fundamentally different tool carriers:

- Indexing tool turrets

- Tool magazines combined with tool changing devices.

Scientists originally assigned little importance to the numerically controlled turning machine because lathes had already been mechanically automated to a great extent. Although turning machines have always been quite versatile and highly automated, they have become even more universal and flexible through the use of numerical control (NC) and especially through computerized numerical control (CNC) [33]. Highly automated turning machines are also known as turning centers or turning cells. These names are based on the familiar terms ‘‘machining center” and ‘‘manufacturing cell”. The next section describes the KBES proposed for fast reconfiguration of machine tools using the KBES methodology, using the lathe as a case study.

4. Knowledge-based Engineering System

KBESs will aid in the recording of all the tacit knowledge (people), as well as the explicit knowledge (documents, videos, databases, among others) involved in the manufacturing system for the reconfiguration. To construct the proposed architecture, the Unified Modeling Language (UML) is chosen to define the structure of the reconfiguration of the machine tool in relation to its components and the KBES methodology, such as the main system, sub-systems and modules that comprise the proposed architecture. UML is used for the model description and the interactions between the elements, and object oriented principles like abstraction and inheritance are used for structuring the set of modules for systems development.

In KBES, once the available information on sources was collected, the definition and acquisition of knowledge were left to complete the process. It was required to identify the people who have knowledge on the subject. The people identified were professors, students, automators, mechanical experts and vendors, and it was necessary to interview them for validation of all the information gathered, and also to get more valuable information. In this case, with all the results obtained from gathering and elicitation, an analysis was done to define the most important features to consider about each lathe component.

Rules definition was done considering the required input and output from the reconfiguration process. A structure composed of three particular steps such as basic rules, math operations, and dependent rules (Figure 2) was defined from the inputs and outputs.

The purpose of basic rules is to receive the input data to process them for (i) defining some initial components of the reconfiguration or (ii) querying in a database to get information required by the following two steps. The second step applies the mathematical formulae that are required to define some components after having partial rules. The defined mathematical rules are based on formulas such as: (i) motor selection, (ii) production capacity of pieces involved in work (iii) precision involved in encoder (sensor), ratio of gears, ball screw advancement, and data input from customer, and (iv) power formula, which aids to define the spindle motor type with its features, driver type, and also advancement speed.

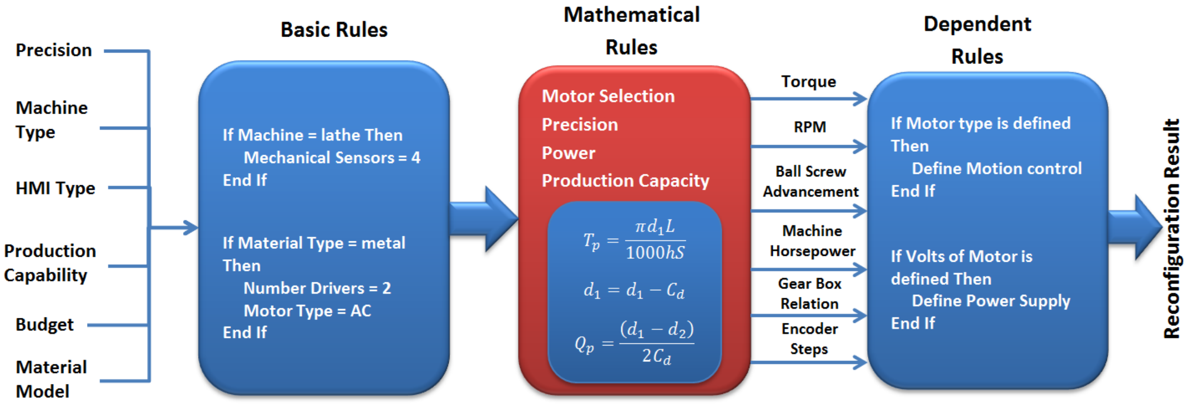


Figure 2: Flow of data and processing through three steps of rules defined.

Figure 2 shows an example of the formulas required to compute production capacity, where: T_p is the time of one pass (minutes), d_1 is the initial diameter, d_2 is the final diameter, L is the length, h is the feed, C_d is the depth of cut, S is the cutting speed, and Q_p is the quantity of passes. Finally, dependent rules are the rules that must be defined with the aid of other components that share their features in order to be calculated and defined.

Before implementing the software application using technological tools, it was required to structure the collected knowledge. UML diagrams were used to represent some of the collected knowledge. In this case, diagrams such as activity diagrams, use case diagrams, class diagrams, component diagrams, and sequence diagrams, among others, were developed. Some diagrams are specific for components or systems, while others represent the activities that they follow.

The activity diagram, which describes the Reconfiguration Process with its principal activities (Figure 3), is presented first. The whole process is composed of a series of procedures that are done both manually and with the help of the software tool.

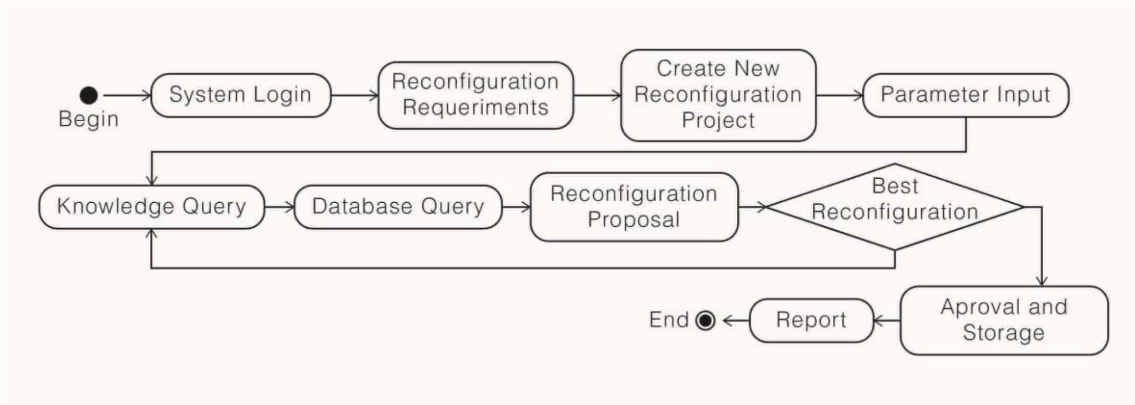


Figure 3: UML reconfiguration process as activity diagram.

The component diagram below shows how the systems of components interact. The component diagram defines how the classes, packages and other low level elements of the system group interact with the upper level components (Figure 4). The interface used between the components is also described. The components are compiled software artefacts that provide the behavior of the software within the restrictions imposed by the requirements or specifications.

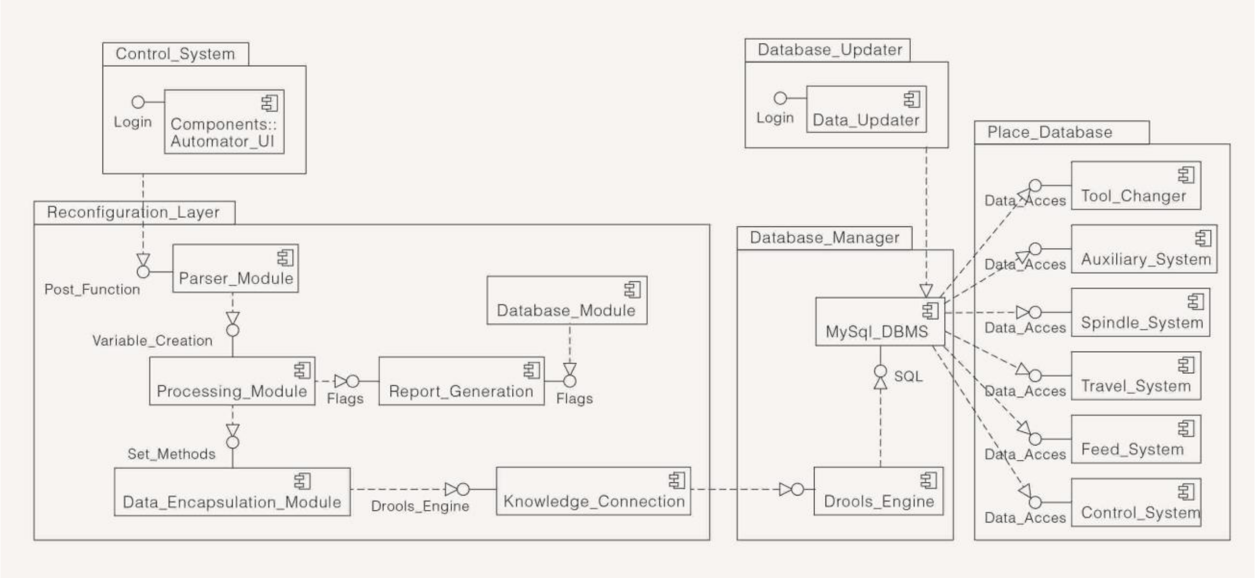


Figure 4: UML interactions between systems as component diagram.

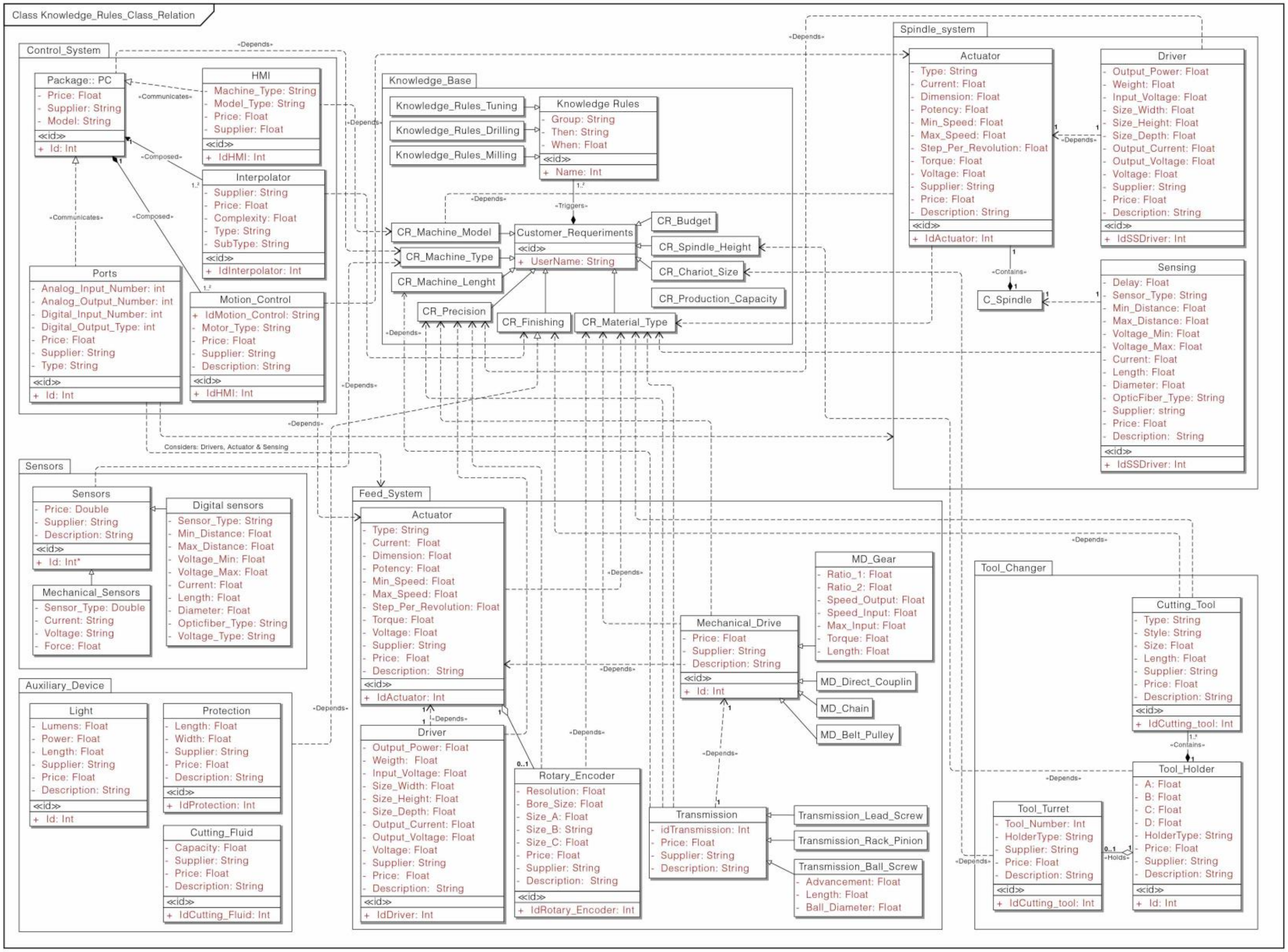


Figure 5: UML General components as class diagram.

Then there are class diagrams, which show the collection elements and describe their types, content, and their relationships between each other, as well as the detailed and specific features of each system (Figure 5). The most relevant packages part of this diagram are the subsystems required for reconfiguring a machine tool, such as: the control system, sensor, feed system, tool charger and spindle system. Several classes are part of each package, which abstracts the components used for characterizing each subsystem. For example, the feed system is composed by actuator, driver, rotary encoder, mechanical drive, among others. The UML standard allows representing each feature of a component using the attributes of a class (red color words in the figure 5). Besides, the association between each component is described using UML relations of dependency, inheritance or composition. This relation could be between components that are part of different subsystems or between components in the same subsystem. Finally, the knowledge base package represents the structure used for implementing the knowledge base proposed including the rules and the customer requirements.

5. Development of the Software Application

The Information Technology tools are derived from research in which a series of rules and knowledge about how reconfigurations work for different types of manufacturing machines are defined. In this case, a structure that facilitates easy interaction between the different levels of architecture and the knowledge database are defined. A system for Knowledge Bases is a special type of system where the different instructions and data processing are defined in a knowledge database instead of directly implemented statements in the classes that the program comprises. The requirements defined for the software application are defined in the following user stories:

- As a user, I want to record all the information related with the state of the reconfiguration required (budget, time, material, precision, production capacity) so that the software have appropriate information to perform the reconfiguration.
- As a user, I want to analyze or visualize all the reconfigurations possible for a machine tool ordered so that I can chose the most appropriate.
- As a user I want to observe the details of the components recommended by the software to perform the reconfiguration so that I can chose the most appropriate.

Additionally, there are several functional requirements necessary for the development of the user stories:

- The application must provide a way to record and store all the information related to the machine tools and its components, categorized by: Sensors, control system, feed system, tool charger, spindle system.
- The application must provide a structure for storing expert rules that using reconfiguration information such as budget, time, material, precision and production capacity is able to chose several option of reconfiguration, that is, commercial details of sensors, control system, feed system, tool charger and spindle system recommended.

The KBES is composed of several layers that interact, as shown in Figure 6.

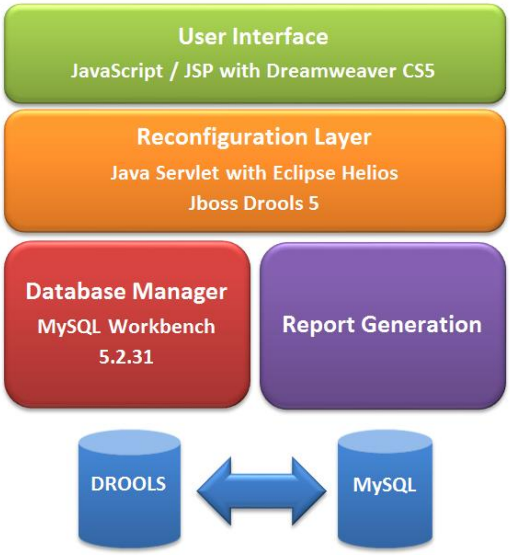


Figure 6: Software Architecture.

According to Figure 6, the sections defined by the architecture and its tools are described below:

- User interface: The user interface represents the form in which the software communicates with users. In this case an automation user permits easy interaction and data capture. By using the interface the user is able to register and release parts, register users, configure rules and access the database to obtain the desired results. The user is also able to permit data requests and obtain historical reports and their reconfigurations. The tools selected to work were: Adobe Dreamweaver Design Suite in its CS 5 version and the languages used were JSP (Java Server Page) and JavaScript.
- Reconfiguration Layer: This layer provides methods and actions for the data introduced on a superior level, as well as methods for returning obtained results. Several modules are concentrated within the layer. Its functions are the automation of processes and database searches, and the elaboration of calculations. The programming language selected was JAVA. Its Servlets modality permits work in multiplatform web environments with programming flexibility. Java is integrated with a database of business rules which contains instructions and statements for the realization of consults and the obtaining of parameters according to what the automation user has introduced. These rules are designed once a series of investigations and analyses are done. They are then passed to a Java language in order to write them in the DROOLS system for the administration of Business Rules.
- Database manager: The database is comprised of a database Entity-Relation mounted

in the MYSQL System for the administration of Databases (BDMS) which allows for the creation of tables and attributes for each of the different machine components. Database consulting can be done with the SQL language.

- Report generation: When a reconfiguration process is finished, a copy of the reconfiguration obtained is stored for later analysis. This copy will be generated through the same code in Java with a backup in the MYSQL database which will permit future access and the use of data. The database will be built on MySql and with the DROOLS tool.

To have a better understanding of building architecture and connections between tools, Figure 7 shows the functionality of the software application in more detail.

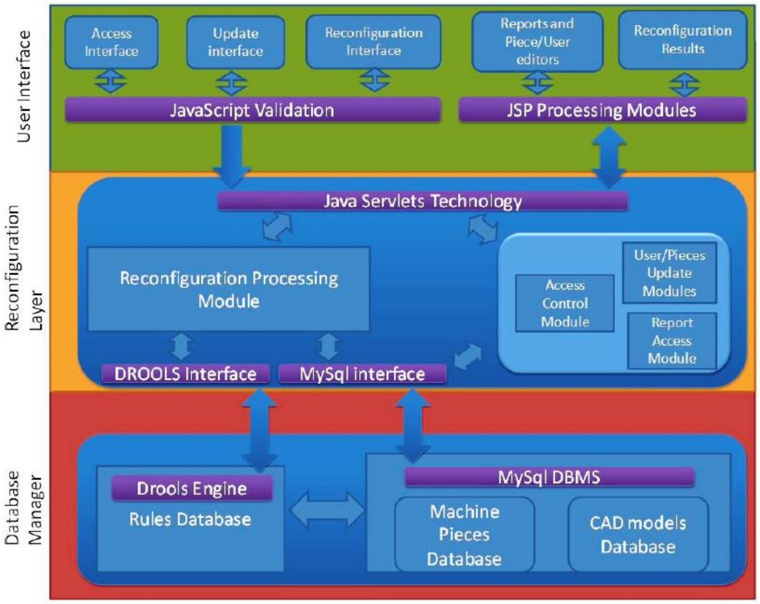


Figure 7: Detailed architecture functionality with its tools.

As explained above, the system is divided into different layers which communicate in order to achieve the expected results for a reconfiguration process. In this case, the function of each section is:

- Web-Based User Interface: In this case, the user selects from the different menus displaying the tasks or options to update, edit, modify, delete or create new components in the database, create new users in the system, or start a new reconfiguration. This allows for interaction with the final user, capturing data within the system in different forms by employing the JSP and JavaScript technology. Java Server Pages (JSP) is a technology from Java that permits the generation of dynamic content for the web and data display from a database or Servlet process in HTML or similar format. This technology can work in an independent manner due to support from JAVA's programming language. This language allows for the creation of small to medium scale web applications without the necessity of intermediaries. JavaScript is a language from Script which is oriented to objects. It has the capacity to handle functions and functional programming that allows the programmer to create and control the flow of data from the customer side. It also allows calculations and validations to be done before being delivered to the server for further processing.

- Access Interface: The interface is the one in charge of capturing and preparing the access data of the different system users. This interface is validated by a JavaScript that is sent to the next layer of the architecture for processing.
- Update Interface: The system is provided with a series of interfaces especially designed for registering, deleting, changing and editing pieces to store in the database. These interfaces are displayed to the user through different menus and are validated using JavaScript to avoid the insertion of false data or data types not permitted in the data capture forms. There is also the interface that permits the capture of all data necessary to process the business rules designed for the system. This interface shows the user a series of fields to fill out. Some fields are already defined with fixed values, and others with values to be introduced manually. The capture form is validated with JavaScript in order to send them to the reconfiguration step.
- JavaScript Validation: The system must avoid erroneous or incoherent data in its database, and the JavaScript tool was selected to prevent that from happening. A module has been implemented which is presented in all the data capture forms and helps to avoid the most common user errors. For example, introducing text chains in fields designed for numeric data, or filling out some other important data for a reconfiguration are common errors.
- Reports, Pieces and User Editors: Additionally, there are some interfaces for the display and editing of stored data. Three types of interfaces are identified:

  (i) Users: Provides a tool with which it is possible to update and register new users to the system. Some personal data must be registered and are processed later. In the same manner, in the case of an already existent user, all data required to be processed from the Reconfiguration Layer are shown in the screen utilizing JSP in order to display information as well as editing information.

  (ii) Pieces: The database will be consulted in order to obtain the information of determined pieces or system of pieces. This information is shown as data tables or lists of stored data, and will be shown to the user in order to offer changes or deletion options for registered pieces. They also allow the user to update the stored data easily.

  (iii) Reports: After a successful reconfiguration is achieved, a report is produced. In this case all results are stored in the database for further consultation. This information will be processed by JSP and will be displayed for editing or deletion as required.
- Reconfiguration Results: The interface for the reconfiguration results is designed to display a list of pieces and components that represents a successful reconfiguration. In this interface, the data will be captured by the JSP and displayed in a list of pieces where the user will be able to select if the configuration will be stored, or deleted in

order to create a new one.

- JSP Processing Module: All the graphic parts of the system are displayed to the user in JSP (pages), allowing for the inclusion of Java code in the normal HTML pages. This module is included in practically all the pages that require the capture and display of information. When the majority of the internal processes are finished, certain data to be displayed are given to the interfaces, where JSP is involved. The capture and display processes utilize several language tools.
- Reconfiguration Layer: In this level (due to the easy interaction between JSP and JAVA), a system based on Servlets was implemented which captures the data that are sent from the interface and that are further sent to respective modules for processing. Servlet is a type of JAVA language designed to take advantage of the capacities of servers that has host applications based on the Request-Reply system. These requests are used to process data from a data capturing form and then an adequate reply is returned according to a series of programmed operations considering the Servlet. Java programming language was selected for this project because it is a multiplatform language with the ease of working in practically any operative system. It also offers a wide variety of additional libraries which allow for easy communication between other systems and interfaces.
- Reconfiguration Processing Module: The reconfiguration processing module is the principal module of the system. It is where all the operations, consultations and actions necessary to create a machine reconfiguration take place. It is based on the data from the automator or principal user. In this module the data sent from the reconfiguration interface are captured, and these data are encapsulated by the Servlets technology in order to be placed in internal variables which will pass through different filters and operations. Also, this module has two additional interfaces; the DROOLS interface and the MYSQL interface. These permit interaction with the rules databases and with the machines pieces databases. According to the data previously processed, they can also perform the requests to said databases and generate the expected results.
- Access Control Module: This is the module designed for capturing data from the Access Interface in order to make sure that the user that wishes to access the software is previously registered in the system database. The access control module is based on the Servlets technology and the MYSQL interface to obtain and process the data previously sent from the user interface, and to redirect the user to the welcome selection or deny the access to the user.
- User/Pieces Update Module: There are two modules: (i) Users: Modifies and updates the information referring to the user of the system and is capable of registering and eliminating users; (ii) Pieces: Controls three principal tasks; registering new pieces in the database, editing, and eliminating the current content of the database. This module receives the parameters of the pieces to process from the user interface. Then, using the Servlet technology, these parameters are prepared to be inserted in the database or to be eliminated if required. This is achieved through the connection interface of MYSQL and with internal access methods, as well as data control flux.

- Report Access and Generation Module: These modules are the ones in charge of storing the data generated for later reference (from a successful reconfiguration), and accessing and displaying the stored results of previous reconfigurations. This allows users to control all that is done and has been done in the system, making it easy to repeat searches with similar data. Servlets and MYSQL interface is used to provide access and display the stored data.
- DROOLS Interface: In this interface, the Business Rules Management System (BRMS) called DROOLS is used. It is specifically designed to work with the JAVA programming language and it offers an attractive solution for business rules administration. JBoss provides a series of libraries which work as an interface between the Java programming language and the established rules in the DROOLS database. This interface is completely free and supported by the JBOSS Community.
- MYSQL Interface: The database selected for reconfiguration software was MYSQL, which is an OpenSource Database. One of its principal characteristics is that it can be easily employed in different environments. A special library serves as the interface between the Servlet Data and the MYSQL database which is free and supported as well.
- Database Administration: Furthermore, the database has been selected to be divided into two forms. The first is managed by DROOLS, which provides a structured way to easily edit all the rules that are necessary to work with the machines. These rules are interpreted from the common language and are converted to programming language statutes in order to be introduced to the DROOLS system for evaluation and control. DROOLS is a Business Rules Management System (BRMS) with a rule engine based on the interference of forward chaining, also known as a production rules system using an advanced implementation of the Rate algorithm. It is free distribution software according to the Apache licensing terms. DROOLS supports the JSR-94 standard for its business rules motor and a framework for the construction enterprise, maintenance and enterprise politics reinforcement of an organization, application or service. DROOLS employs JCR (JackRabbit) to administer the rules repository, and the Java Authentication and[630] Authorization Service (JAAS) standard for authentication and authorization. After analyzing the application context, and according to previous investigations regarding the form in which work must be done to achieve the desired results, DROOLS is selected because the way in which business rules are administered is very similar to the way in which the KBES rules are contemplated. This tool eases the editing of business rules without the necessity of recompiling the system code in every update because the rules are an independent editable archive. They are editable from each text editor and any web interface.

  The machines pieces database is handled by MYSQL. MYSQL is a relational, multithread and multi-user database administering system with more than six million completed implementations. MYSQL is a subsidiary of Sun Microsystems, which is an Oracle Corporation subsidiary, and since April 2009 MYSQL has been developed as free software under a dual licensing scheme. One of the main advantages of using this tool is that it complies with SQL lineaments, which makes it easy to learn and implement at a reconfiguration level. Using the MYSQL Workbench tool has made editing data tables so they conform to the database easier. They are consulted according to the rules and data previously introduced. Then a series of data is obtained which in conjunction with DROOLS, returns a series of results for processing to the Reconfiguration Layer, which is then displayed to the final user.

- Machine Place Database: This is one of the databases that contains data tables with data of pieces from different machines and their attributes. These are accessed by different searches (queries) in MYSQL language, allowing the required data to be obtained. Also, it is in this database where all the updated transactions and the registrations are stored. They communicate with the highest levels of the architecture (vital internal aspect of the system).
- DROOLS engine: The database will contain a series of system statutes and rules which will be called from the reconfiguration level. Once they are executed, the parameters and the information contained in the piece database will be read in order to return results for the machine reconfiguration. These files have a format in which every rule is delimited from the others. A proper name is assigned which makes its comprehension and execution easier. The programmer may also insert comments.

6. Discussion and Results

The software described can handle the entire reconfiguration of a machine with enough precision to compete with the current automation process. It is worth to mention that the application was developed as an experimental prototype in order to test a proof of concept and it is not intended to be used as a tool for real scenarios in companies yet. With this premise some examples were developed in order to test the reconfiguration software. These examples were then compared with a real reconfiguration made by an expert to identify discrepancies.

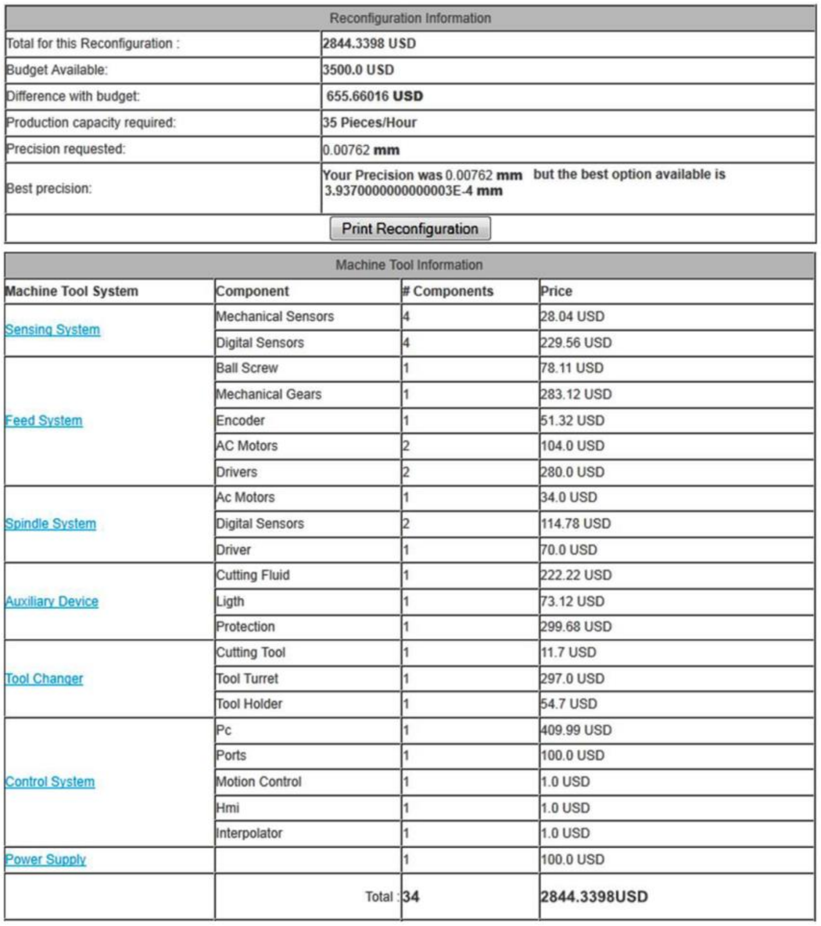
6.1. Case 1: Complete Reconfiguration

The machine used for the test is a generic product with multiple dimensions and very high precision. The software will generate a list of components required for the complete reconfiguration of the machine with this information. The machine used for these examples is an EMCO Maximat Super 11. Table 2 describes the specifications of the machine.

Model Lathe	EMCO Maximat Super 11
Material	Aluminium
Spindle height	44 mm
Chariot Width	70 mm
Machine Lenght	800 mm
Budget	3500 USD
Precision	0.00762 mm
Cariage weight	100 Kg
Tool Number	1, Type: AR
Death Time	1 minute

Table 2: Test parameters of the mechanical tool.

Figure 8 describes reconfiguration according to the software. This summary is intended to describe the capabilities of the reconfiguration after its implementation, as seen in the best precision type. It also shows the difference in cost according to the expected budget.

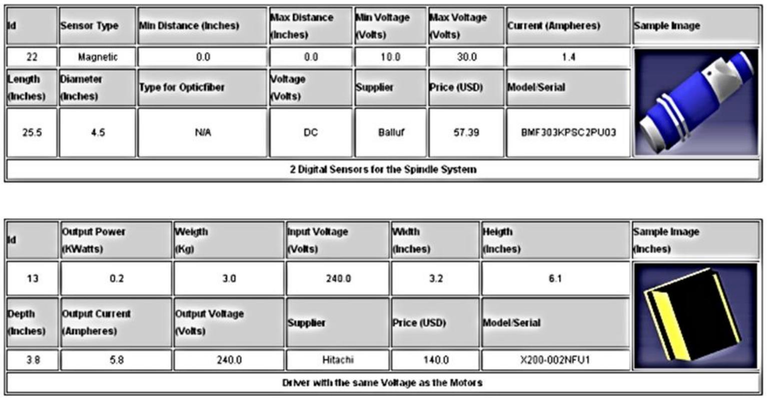


Reconfiguration Information	
Total for this Reconfiguration :	**2844.3398 USD**
Budget Available:	**3500.0 USD**
Difference with budget:	**655.66016 USD**
Production capacity required:	**35 Pieces/Hour**
Precision requested:	0.00762 **mm**
Best precision:	**Your Precision was** 0.00762 **mm but the best option available is 3.9370000000000003E-4 mm**

Machine Tool Information			
Machine Tool System	**Component**	**# Components**	**Price**
Sensing System	Mechanical Sensors	4	28.04 USD
	Digital Sensors	4	229.56 USD
Feed System	Ball Screw	1	78.11 USD
	Mechanical Gears	1	283.12 USD
	Encoder	1	51.32 USD
	AC Motors	2	104.0 USD
	Drivers	2	280.0 USD
Spindle System	Ac Motors	1	34.0 USD
	Digital Sensors	2	114.78 USD
	Driver	1	70.0 USD
Auxiliary Device	Cutting Fluid	1	222.22 USD
	Ligth	1	73.12 USD
	Protection	1	299.68 USD
Tool Changer	Cutting Tool	1	11.7 USD
	Tool Turret	1	297.0 USD
	Tool Holder	1	54.7 USD
Control System	Pc	1	409.99 USD
	Ports	1	100.0 USD
	Motion Control	1	1.0 USD
	Hmi	1	1.0 USD
	Interpolator	1	1.0 USD
Power Supply		1	100.0 USD
	Total :	**34**	**2844.3398USD**

Figure 8: Summary reconfiguration for case 1 with budget details.

As shown, it does not provide the details in the bill nor the summary. This is because one can simply go through the page and change the components as desired (Figure 9).



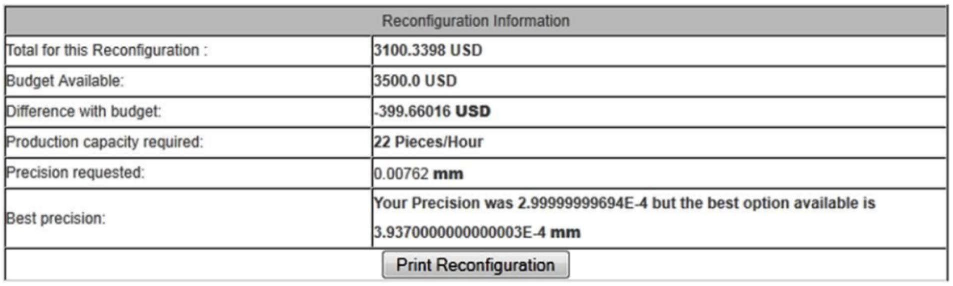
Id	Sensor Type	Min Distance (Inches)	Max Distance (Inches)	Min Voltage (Volts)	Max Voltage (Volts)	Current (Ampheres)	Sample Image
22	Magnetic	0.0	0.0	10.0	30.0	1.4	
Length (Inches)	Diameter (Inches)	Type for Opticfiber	Voltage (Volts)	Supplier	Price (USD)	Model/Serial	
25.5	4.5	N/A	DC	Balluf	57.39	BMF303KPSC2PU03	
2 Digital Sensors for the Spindle System							

Id	Output Power (KWatts)	Weigth (Kg)	Input Voltage (Volts)	Width (Inches)	Heigth (Inches)	Sample Image (Inches)
13	0.2	3.0	240.0	3.2	6.1	
Depth (Inches)	Output Current (Ampheres)	Output Voltage (Volts)	Supplier	Price (USD)	Model/Serial	
3.8	5.8	240.0	Hitachi	140.0	X200-002NFU1	
Driver with the same Voltage as the Motors						

Figure 9: Component detail as seen by the user.

6.2. Case 2: Complete reconfiguration, different material and dead time

In this second case, most of the parameters will stay the same but the material and the dead time will change in order to prove the mathematical rules that use different types of comparisons to provide the best components. This time the type of material was changed to steel and the dead time was changed to two minutes. Figure 10 shows the reconfiguration summary provided by the software.



Reconfiguration Information	
Total for this Reconfiguration :	**3100.3398 USD**
Budget Available:	**3500.0 USD**
Difference with budget:	**-399.66016 USD**
Production capacity required:	**22 Pieces/Hour**
Precision requested:	0.00762 **mm**
Best precision:	**Your Precision was 2.99999999694E-4 but the best option available is 3.9370000000000003E-4 mm**

Figure 10: Summary reconfiguration for case 2.

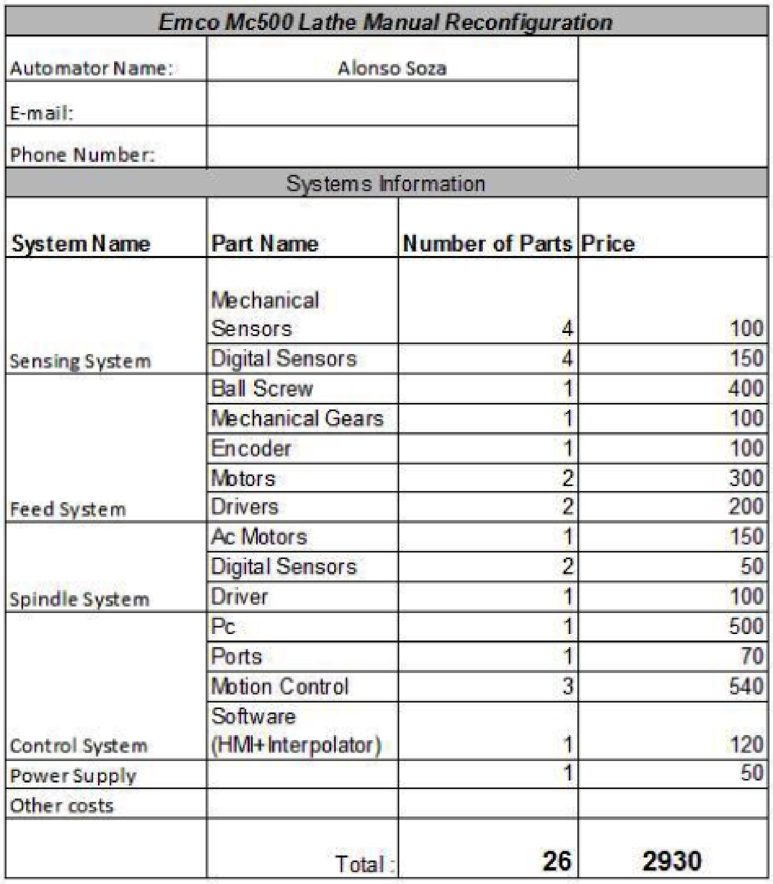
6.3. Case 3: Partial reconfiguration

In this particular case, an example is given using partial reconfiguration. It is partial reconfiguration because the lathe is semi-automatic or some components already exist. An expert made a real reconfiguration bill for this case using the EMCO MC500 from the Mechatronics Lab at the ITESM. This bill is compared with the one produced by the software. Table 3 describes the specifications that were used as the inputs for the reconfiguration process.

Machine Type	Generic
Material	Bronze
Resistance of Tool	H10
Machine Length	1.98 Feet
Budget	5000 USD
Production Capacity	13
Precision	0.00025 mm
Complexity	High
Ports type	DAQ
Control System	CNU

Table 3: Test parameters for case 3.

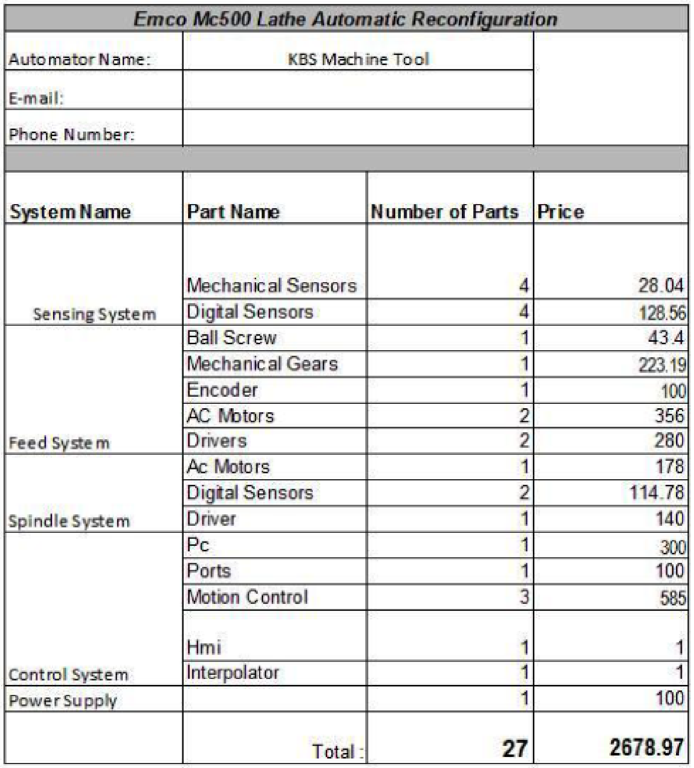
Figure 11 presents the real reconfiguration with its results that the expert has developed:



Emco Mc500 Lathe Manual Reconfiguration			
Automator Name:	Alonso Soza		
E-mail:			
Phone Number:			
Systems Information			
System Name	**Part Name**	**Number of Parts**	**Price**
Sensing System	Mechanical Sensors	4	100
	Digital Sensors	4	150
Feed System	Ball Screw	1	400
	Mechanical Gears	1	100
	Encoder	1	100
	Motors	2	300
	Drivers	2	200
Spindle System	Ac Motors	1	150
	Digital Sensors	2	50
	Driver	1	100
Control System	Pc	1	500
	Ports	1	70
	Motion Control	3	540
	Software (HMI+Interpolator)	1	120
Power Supply		1	50
Other costs			
	Total :	**26**	**2930**

Figure 11: Real reconfiguration resulting by expert.

The results that the reconfiguration software displays, considering the input of the real reconfiguration, are shown in Figure 12.



Emco Mc500 Lathe Automatic Reconfiguration

Automator Name:	KBS Machine Tool
E-mail:	
Phone Number:	

System Name	Part Name	Number of Parts	Price
Sensing System	Mechanical Sensors	4	28.04
	Digital Sensors	4	128.56
Feed System	Ball Screw	1	43.4
	Mechanical Gears	1	223.19
	Encoder	1	100
	AC Motors	2	356
	Drivers	2	280
Spindle System	Ac Motors	1	178
	Digital Sensors	2	114.78
	Driver	1	140
Control System	Pc	1	300
	Ports	1	100
	Motion Control	3	585
	Hmi	1	1
	Interpolator	1	1
Power Supply		1	100
	Total :	27	2678.97

Figure 12: Results of reconfiguration software.

In comparison, it can be observed that the results of the reconfiguration software have savings of 8.91 % compared with the reconfiguration proposed by the expert. In the software case, all the mathematical requirements about machining processes are considered against the empiric knowledge. There are 27 pieces in this case versus 26 in the real results. If prices of components are observed, motion control is more expensive according to the software results.

7. Conclusions

A model for a low-cost automated intelligent assistant was successfully created, and the model covers all the required characteristics in terms of the clients' requirements, which were defined in the first stage of the methodology. The KBES methodology that was employed to create the entire base for reconfiguration process was developed with the aid of the Unified Modeling Language (UML) to define the modules of the architecture, and to structure the interaction between the knowledge and resources involved in the process. Through the model it is possible to give a highly detailed description of all the machine tools' components and how these components can be transformed through knowledge modeling techniques, such as UML, into functional software from its abstract models. All the particular objectives were accomplished because it was possible to identify diverse open source technological tools that were employed to build the software architecture.

From the suggested architecture (as seen in Figure 6), the first layer known as Web-Based User Interface was developed with Javascript and JSP. It was designed to be accessible from the Internet and it shows an intuitive interface. The current interface only gives the option for lathe reconfiguration, since it was the only one chosen to implement as a case study, but the other machine tools can be easily added. To match necessities and components, the Reconfiguration Layer was designed in JAVA, consequently allowing a modular approach and easy implementation. The database is easily adapted from the ODBC Driver and the Graphic Interface, and the Rules Database does not need to be changed at this point but can be updated if needed.

At the bottom of the architecture is the Database Administration Layer which uses DROOLS and MySql to provide the best matches. DROOLS has the capacity to manage several complex rules, making this layer smart enough to handle the necessities of the Reconfiguration Layer even though it was designed for business administration but for this case was used for knowledge management. Additionally, several rules were created and became a net of interconnected rules where some of them are dependent, basic and mathematical. DROOLS allowed complex if-then rules and integrated easily with the MySql database system. At this time the system does not have an automatic update system, thus the administrator must directly update the data through several forms and edit the existing component database. All the information needed for each component was carefully selected and evaluated to ensure its compatibility with all industry standards and specifications. The architecture was designed to be a robust, complete, stand-alone, centralized application that runs on a web server.

Finally, the software application was tested using three different cases: (i) Case 1: Complete reconfiguration, (ii) Case 2: Complete reconfiguration, different material and death time, and (iii) Case 3: Partial reconfiguration. The results of the machine reconfiguration obtained were successful, fitting the constraints imposed and proposing the required components.

8. Acknowledgments

This research article is published as a product of PhD thesis of Mauricio Hincapié titled "Model design for reconfigurable machine tools in a manufacturing system using knowledge based engineering systems", funded by Instituto Tecnológico y de Estudios Superiores de Monterrey, Campus Monterrey, Mexico, so we want to acknowledge their support throughout the research project.

9. References

[1] A. Bouri, M. Breij, M. Diop, R. Kempner, B. Klinger, K. Stevenson, Report on support to SMES in developing countries through financial intermediaries, Tech. rep., Dalberg (November 2011).

[2] U. N. I. D. Organization, World manufacturing production statistics for quarter, Tech. rep., United

Nations (September 2013).

[3] G. Research, The world machine tool output & consumption survey, Tech. rep., Gardner & Associates (September 2013).

[4] H. ElMaraghy, Reconfigurable process plans for responsive manufacturing systems, in: P. Cunha, P. Maropoulos (Eds.), Digital Enterprise Technology, Springer US, 2007, pp. 35–44.

[5] N. R. Counsil, Visionary Manufacturing Challenges for 2020, Academy Press, 2002.

[6] R. Landers, B. Min, Y. Koren, Reconfigurable machine tools, Annals on Manufacturing Technology 50 (1) (2001) 269 – 274.

[7] S. Saad, The reconfiguration issues in manufacturing systems, Journal of Materials Processing Technology 138 (13) (2003) 277– 283. [29]

[8] H. M. Selim, R. G. Askin, A. J. Vakharia, Cell formation in group technology: Review, evaluation and directions for future research, Computers & Industrial Engineering 34 (1) (1998) 3– 20.

[9] V. Kannan, S. Ghosh, A virtual cellular manufacturing approach to batch production*, Decision Sciences 27 (3) (1996), 519–539.

[10] M. Rheault, J. Drolet, G. Abdulnour, Physically reconfigurable virtual cells: A dynamic model for a highly dynamic environment, Computers & Industrial Engineering 29 (14) (1995) 221– 225.

[11] M. Mehrabi, A. Ulsoy, Y. Koren, P. Heytler, Trends and perspectives in flexible and reconfigurable manufacturing systems, Journal of Intelligent Manufacturing 13 (2) (2002) 135–146.

[12] J. Heilala, P. Voho, Modular reconfigurable flexible final assembly systems, Assembly Automation 21 (1) (2001) 20 – 30.

[13] W. Chow, Assembly Line Design: Methodology and Applications, CRC Press, 1990.

[14] G. Pritschow, K.-H. Wurst, C. Kircher, M. Seyfarth, Control of reconfigurable machine tools, in: Changeable and Reconfigurable Manufacturing Systems, Springer Series in Advanced Manufacturing, Springer London, 2009, pp. 71–100.

[15] Y. Koren, Reconfigurable Manufacturing Systems, John Wiley & Sons, Inc., 2010, pp. 227–252.

[16] A. Molina, R. Bell, A manufacturing model representation of a flexible manufacturing facility, Journal of Engineering Manufacture 213 (1) (1999) 225 – 245.

[17] M. Mehrabi, A. Ulsoy, Y. Koren, Reconfigurable manufacturing systems: Key to future manufacturing, Journal of Intelligent Manufacturing 11 (4) (2000) 403–419.

[18] K. Reuven, M. Yong-Mo, Virtual arch type reconfigurable machine tool design, principles and methodology, Tech. rep., The University of Michigan NSF ERC for RMS Ann Arbor (2000).

[19] K. Oldknow, I. Yellowley, Design, implementation and validation of a system for the dynamic reconfiguration of open architecture machine tool controls, International Journal of Machine Tools and Manufacture 41 (6) (2001) 795 – 808.

[20] S. Szykman, W. Regli, R. Sriram, The role of knowledge in next-generation product development systems, Journal of Computing and Information Science in Engineering 1 (1) (2001) 3 – 11.

[21] M. Abdullah, R. Paige, C. Kimble, I. Benest, A UML profile for knowledge-based systems modelling, in: Software Engineering Research, Management Applications, 2007. SERA 2007. 5th ACIS International Conference on, 2007, pp. 871–878.

[22] M. Merdan, I. Terzic, A. Zoitl, B. Favre-Bulle, Intelligent reconfiguration using knowledge-based agent system, in: Emerging Technologies and Factory Automation, 2005. ETFA 2005. 10th IEEE Conference on, Vol. 1, 2005, pp. 4 pp.–1054.

[23] A. Hakansson, R. Hartung, Reengineering for knowledge in knowledge-based systems, in: Knowledge-Based Intelligent Information and Engineering Systems, Vol. 4251 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2006, pp. 342– 351.

[24] J. Malec, A. Nilsson, K. Nilsson, S. Nowaczyk, Knowledge-based reconfiguration of automation systems, in: Automation Science and Engineering, 2007. CASE 2007. IEEE International Conference on, 2007, pp. 170–175.

[25] D. A. Guerra-Zubiaga, R. I. M. Young, Design of a manufacturing knowledge model, International Journal of Computer Integrated Manufacturing 21 (5) (2008) 526–539.

[26] R. Jardim-Goncalves, J. Sarraipa, C. Agostinho, H. Panetto, Knowledge framework for intelligent manufacturing systems, Journal of Intelligent Manufacturing 22 (5) (2011) 725–735.

[27] P. Renna, Capacity reconfiguration management in reconfigurable manufacturing systems, The International Journal of Advanced Manufacturing Technology 46 (1-4) (2010) 395–404.

[28] Y. Alsafi, V. Vyatkin, Ontology-based reconfiguration agent for intelligent mechatronic systems in flexible manufacturing, Robotics and Computer-Integrated Manufacturing 26 (4) (2010) 381 – 391.

[29] M. Ramírez, J. P érez, M. Guitiérrez, Tecnología de CNC a bajo costo para la automatización de la pequeña y mediana empresa de países en vías de desarrollo, in: 8vo Congreso Iberoamericano de Ingeniería Mécanica, 2006, p.18.

[30] J. Pereda, D. Romero, M. Hincapié, M. Ramirez, A. Molina, Developing a universal numerical control machine based on an enterprise multilevel framework and its (IPPMD) reference map and methodology, Annual Reviews in Control 34 (1) (2010) 145 – 154.

[31] M. Hincapié, Reconfiguration model using knowledge-based engineering systems: A machine tool case study, Ph.D. thesis, Monterrey Institute of Technology in Higher Education (November 2011).

[32] L. Alting, Procesos para Ingeniería de Manufactura, Alfaomega Grupo Editor, 1996.

[33] H. Kief, CNC for Industry, Hanser Gardner publications, 2000.

[34] Hincapié, M., Guemes-Castorena, D., Contero, M., Ramirez-Cadena, M., Diaz, C. Reconfiguration model using knowledge based engineering systems. Journal of Manufacturing Technology Research, Volume 6, No. 1-2, pp. 63-81, 2014.

05-02-2016

Dr. Edward Szczerbicki
Associate Editor
Knowledge-based and Intelligent
Engineering Systems

Ref.: Iteration 1. Manuscript “Development of a Software Application for Machine Tool Reconfiguration using a Knowledge-Based Engineering System Approach”.

Dear Dr. Edward,

We thank the reviewers and you for the first round of comments to our article Development of a Software Application for Machine Tool Reconfiguration using a Knowledge-Based Engineering System Approach and the KES *Journal* for the opportunity to seek its publication.

Based on the comments by the reviewers and you, we have rewritten the article and answered all the concerns expressed. Attached herewith you will find: (1) the original article before any changes were made, (2) a table containing each one of the comments by the reviewers (left column) with changes and responses from us (right column), and (3) the rewritten article after corrections highlighted with red color text.

Please let us know of any additional requirements to satisfy in order to have the article published. We appreciate the time and effort that you and the reviewers have devoted to our paper.

Kind regards,

Prof. Dr. Eng. Mauricio Hincapié
Research Director
Research Center
Institución Universitaria Salazar y Herrera (IUSH)
Medellin, Colombia

Referee: 1		
Recommendation: Accept.		
	Comments	
	OBSERVATION	***AUTHORS' RESPONSES***
1	The paper is a good description of a software tool.	The authors thank the referees' positive opinions and seek to satisfy their requests to improve the article.
2	However, the status of the tool remains unclear. Is it in real use? Is it just an experimental prototype?	We have addressed the observation proposed by the referee doing the following modification in the manuscript (Section Discussion and Results, Page 19): In the manuscript was added the following paragraph in which the authors make clear that the prototype developed was for experimental purposes: "It is worth to mention that the application was developed as an experimental prototype in order to test a proof of concept and it is not intended to be used as a tool for real scenarios in companies yet."
3	Figure is unreadable and suppose in print it will still be unreadable.	The figure 3, 4 and 5 describing the structure of the software using UML standard were edited and modified in order to make readable the text and easy to understand.
Referee: 2		
Recommendation: Accept subject to revision		
	Comments	
	OBSERVATION	***AUTHORS' RESPONSES***
4	Interesting Idea.	The authors thank the referees' positive opinions and seek to satisfy their requests to improve the article.
5	The specific requirements on the application are unclear.	The most important application requirements were added in the manuscript (Page 13).
6	It would be much easier to read if the figures were placed right next to explanation and not after references.	We have addressed the observation rewriting several paragraphs in order to describe more deeply the

		figures (Red color text in the manuscript next to the figures). The figures were placed after the description text as was recommended by the referee.
7	Authors mention for example the Access Control module without further explanation of how it is implemented.	The description of the Access Control module was extended describing the implementation performed (Page 17).
8	Related work should be revised. As it is now it is more a background information than related work.	The related works section was complemented with a discussion of the works compared in table 1, and with the enumeration of the model proposed contributions (Pages 5 y 6).