We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

186,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

A Knowledge Management System Embedded in the New Semantic Technologies

Vasile Mazilescu
Dunărea de Jos University of Galați
Faculty of Economics and Business Administration
Romania

1. Introduction

There is a large interest in organizational knowledge in the context of transition to knowledge economy, where knowledge is viewed as the main source of sustainable competitive advantage. Although knowledge management (KM) is primarily concerned with how people and organizations use their knowledge assets, one way to do this efficiently is to employ technology to facilitate the KM processes (Alavi, 1999). Consistent with the growing interest in organizational knowledge and KM, many ICT researchers have been promoting a class of information systems, referred to as Knowledge Management Systems (KMSs). The objective of a KMS is to support knowledge capturing, categorizing, storing, searching, distributing and application within organizations. Technical advances in computers' processing and storage capacity, together with linking these computers into networks of distributed nodes, have greatly increased the organizations' capability to deliver goods and services. Along with these capabilities we need quality, accuracy, responsiveness and capacity. Particular topics of interest on KMSs include among others: Organizational knowledge management approaches, Information management challenges, Service Oriented Architecture (SOA), software environments, Semantic web services environments, Information modeling and the representation of semantics, Intelligent software tools and services, Information management systems in practice.

Semantics is the study of meaning. Semantic Technologies (STs) are distributed software technologies that make the meaning more explicit, principally so that it can be understood by computers. New Semantic Technologies (NSTs) will dramatically influence enterprise's architecture and the engineering of new systems and infrastructure capabilities, so that they act as *disruptive technologies* (so innovative that they have the potential to completely change the way we do business) on capturing and sharing next generation knowledge among workers and organizations in the new economy. NSTs are tools that represent meanings, associations, theories, and know-how about the application of things, separately from data and program codes. These systems must be designed as distributed systems, with the ability to combine different knowledge-based techniques (with the purpose of acquiring and processing information and knowledge), based on approximate reasoning methods (Müller, 1996; Lin, 2008). NSTs will better emulate the human decision-making process, also characterized by imprecise and time-varying knowledge (Knight & Passino, 1987; Barachini, 1990; Dubois et al., 1991; Qian, 1992; Nebel & Bäckström, 1994). Time restrictions are not

excessive in common distributed applications. Critical time reasoning problems may occur in case of faulty operations and overloading. The reasoning depth developed for such systems is still poor (Iqbal et al., 2007; Durán &Aguilo, 2008; Marco & Marley, 2009).

The aim of this paper is to present a Knowledge Management System based on Fuzzy Logic (KMSFL), a real-time expert system to meet the challenges of the dynamic environment. The main feature of our integrated shell KMSFL is that it models and integrates the temporal relationships between the dynamic of the evolution of a technological process with some fuzzy inferential methods, using a knowledge model for control, embedded within the expert system's operational knowledge base. As important contributions of this work, we have integrated some elements of control theory, fuzzy and temporal logics and discrete event systems to increase the decision making capacity (Bylander, 1994; Kim & Lee, 2003; Davis, 2006). We also focused particularly on time, in its many facets (real-time, algorithmic complexity and reasoning over time), by using a time meta-equation. The closed-loop of our KMSFL starts from an initial state and allows planning a number of states to achieve a desired final state. At the end of its operation, the control expert system provides the human decider the possible actions, under specific conditions of the problem. All bibliographic sources are important, because each of the mentioned authors has used a series of concepts that we have integrated in KMSFL.

2. Related work

There is a need for incorporating aspects of time and imprecision into real-time KMSs, considering appropriate semantic foundations (Bobrowitz, 1993; Chen & Parng, 1996; Lau et al. 2008). In reality, it is a common practice for organizations to use one or more of the following (technical) systems and concepts to support their KM efforts (Binney, 2001; Wenger, 2001; Mazilescu, 2009b): Knowledge Maps, Taxonomies, Enterprise search engine, e-collaboration tools, Information repositories, Expert Systems, Data Mining / Knowledge Discovery systems, Case-based Reasoning / Question-Answering tools (for Helpdesk and/or Contact Centers), E-Learning and/or Learning Management Systems (LMS), Enterprise Information Portal, Intellectual Capital (IC) measurement tools. Expert systems are examples of relevant knowledge-based methodologies (as Knowledge Capture Systems) that have much to contribute to KMSs, because they manipulate knowledge in order to implement various tasks (Tsui, 2002; Wang & Lin, 2007; Schwartz, 2006; Omar, 2008). KMSs based on Agent Technology try to provide computers the ability to perform various intelligent tasks, for which their human users resort to their own knowledge and to collective intelligence. Currently, KMSs is a highly economically important field due to their ability of approaching new sets of problems, different from those tackled by the classical systems, such as: perception, decision making, planning, diagnosis, natural language comprehension, enterprise KM, learning, web service interfaces, etc. Conventional expert system shells are too slow for real-time environments, and their inference process is boundless. We need a reactive and interruptible system that can assimilate data and asynchronous events, and present the operator with a reasoned opinion in a timely manner. Only speed is not enough (Stankovic & Ramamritham, 1995; Lassaigne & Rougemont, 1996). While practitioners and researchers continue their efforts in designing and building complex intelligent systems, they became conscious of the fact that uncertainty is present not only in human knowledge. Allowing a certain degree of uncertainty in describing complex systems is perhaps the most significant way to simplify them (Zadeh, 1983; Dubois & Prade, 1992;

Luger & Stubblefield, 1993). Different types of uncertainty can be rigorously characterized and investigated in the context of fuzzy sets theory (Zadeh, 1978; Marco & Marley, 2009). Thus, the ability to operate in an uncertain or partially known environment is one of the basic performances of any real-time intelligent system (Passino & Antsaklis, 1989). Real-time calculation is an area of intense research, since the correctness of a system's functioning in a dynamic and distributed environment depends not only on its operating logic, but also on the temporal aspects involved. Such systems include various solutions of systems, subject to various complex time restrictions, with different granularity levels of the time. Temporal knowledge is an essential element for many applications (planning, process control, dynamic situations control). An intelligent system must have reasoning capabilities that take into account a series of events that may occur in the process: interruptions, limitations on processing time, synchronous and asynchronous nature of the new information occurrence. Considering time, we must highlight two complementary aspects: temporal information management and formalization of the temporal reasoning over time and in real-time (Lunardhi & Passino, 1995). Some approaches are based on numerical models and other on symbolic representations of time. Reasoning under real-time restrictions has specific characteristics. Real-time operations often involve a temporal reasoning, but conversely this is not always true. The control involves a close relation between the process and the control system, which must react to the occurring events. The act of intelligent control is interposed between the process and the various physical entities incorporated in the process' superstructure. In this context, the control system has certain Artificial Intelligence (AI) features, if, in the presence of minimal guidance information from a human expert, it can perform complex actions in response to the events coming from outside. In this case, intelligence includes the ability to accept abstract task specifications in a general form of *goals/restrictions* and to produce reasonable actions, which are consistent with the specifications (Mazilescu, 2009a). In any real-time system like KMSFL, there is a fundamental compromise between action and reasoning. We must notice that, logically, the human decider is firmly included in the intelligent control system, which works with certain specific knowledge. The inferential system's logical results can address differently the human operator, the different execution elements or the interfaces with other systems and users. Designing and testing the inferential subsystem for KMSFL, require the existence of certain scientific methods for knowledge acquisition, which, unfortunately, is a heterogeneous, difficult and time consuming process. For this reason, the synthesis of the knowledge management model incorporating human experience was iterative, during a considerably long time, being necessary the indirect development of some methods and environments for testing and simulating some crisp and fuzzy control models, permanently adjusting the inferential subsystem's parameters. For the KMSFL synthesis we adapted and aggregated some AI techniques (possibility theory, symbolic logics, expert systems, etc.) with certain models for technological process control (planning, discrete event systems, qualitative analysis, etc.), closer to human decider's natural way of understanding and operating. This objective was achieved starting from the essential predictability feature that the designed KMSFL must have.

In this respect, we effectively used the notion of microscopic predictability (adopting the Rete compiling technique for the fuzzy processing) and of macroscopic predictability (through KMSFL specification, design and implementation, as a discrete event system). We also introduced logical events. We analyzed and extended the knowledge compilation technique, in order to improve the filtering stage, for the case of fuzzy knowledge (Ghallab,

1988). Furthermore, were used the fuzzy rules along with the fuzzy variables and constants, in the form of possibility distributions, as a basic representation mode for elementary fuzzy knowledge. Choosing between probabilities or possibility distributions and fuzzy sets is not easy, since for finite spaces, the probabilities may have a greater flexibility, in terms of representativeness, but an increased computational complexity. For these reasons, for the KMSFL we have chosen as imprecision measures the possibility measure Π and the necessity measure N (Dubois & Prade, 1992). For the fuzzy variables linking, it was necessary to solve the composition of the fuzzy substitutions, which depends on the compatibility of the fuzzy sets involved in the antecedent of the rules. The necessity measure and the GMP scheme are not independent and the choice of the thresholds for the possibility and necessity measures must be consistent with the chosen inference scheme. This choice is particularly important at all imprecision processing levels within the KMSFL for fuzzy filtering, fuzzy unification, fuzzy conflicts solving, and determination of similar states (Mazilescu, 2011).

We conducted a qualitative analysis of KMSFL using the concepts introduced by (Passino & Antsaklis, 1994). For this, we justified the control system design, in terms of its closed-loop performances, using the traditional concepts of Lyapunov stability of the dynamic systems, applied particularly for KMSLF. Integrating the features of the fuzzy rules base compilation, designing an appropriate inference engine corresponding to the time meta-equation and the corresponding logical justifications are the basic elements in designing the KMSFL. A particularly important issue in designing a control expert system is how the operational knowledge can be acquired and loaded in the knowledge base. In this case, we integrated a model of the process, as a part of the knowledge base. KMSFL was designed so that to coordinate the use of process outputs and reference inputs, to decide how should be synthesized the inferential process' results. In relation to the integration of the expert system within the control expert system's structure, its results can be used by the human decider in decision-making, or can be applied directly on the process.

It was necessary to highlight the knowledge representation in accordance with this system's formalism, to present the basic features of the compiled structure of fuzzy knowledge, the KMSFL parameters, and also to define other logical and computational features of the system. KMSFL-specific knowledge is represented in 1st order logic, aiming the knowledge factorization. We described the design of the fuzzy knowledge compiler, which includes two major parts: static discrimination structure (unification tree, fuzzy unification tree) and variable linking network (algorithms for generating the variable linking network for different rule topologies, fuzzy unification and propagation of the parameters during inferential process) and the system's inference engine algorithm. To highlight the KMSFL applicability, is presented an extended case study (both for the crisp and the fuzzy case). The flexible manufacturing system is composed of many subsystems connected so that they can transmit different amounts of material by means of bond wires, directed according to a given structure schema. For the crisp case, we used the same control model also tested with the G2 generator (Mazilescu, 2009b). This case allowed us to synthesize the fuzzy control model, because of the serious limitations underlined in the tests. KMSFL was implemented in C++. Simulation results have demonstrated the developed system's ability to deal properly with the allocation problem. In addition, we tested a series of parametric dependencies for fuzzy inferential process embedded in the KMSFL engine.

The solution of a control intelligent system as a multi-agent system, presented in *Section 3*, has the quality of emphasizing the place of our fuzzy expert system within a distributed

control structure, as well as the integration of other heterogeneous agents. *Section 4* defines KMSFL, designed and implemented by the author as a logical system with discrete logical events resulting from the inferential process. In this section we introduced the time meta-equation (in which each term has its variable part properly defined– according to relationship 1) and the specific model for the KMSFL as a control expert system (according to relationship 2). A very important problem is the way in which operational knowledge on the process control can be acquired and loaded into the knowledge base. For this, we built a model of the process, as a part of the knowledge base. The KMSFL was designed to coordinate the use of process outputs and reference input(s) and to choose the inferential process' results that will be synthesized. In relation to expert system's integration into the control structure, KMSFL results can be used by the human decider in decision-making, or can be applied directly on the process. *Section 5* describes the knowledge representation in accordance with this system's formalism, the basic features of the structure of compiled fuzzy knowledge, and KMSFL system parameters. We have developed and implemented a fuzzy knowledge compiler, similar to the classic Rete compiler, which includes two major parts: static discrimination structure (unification tree, unification fuzzy tree) and the variables linking network (algorithms for generating the variables linking network for different topologies of rules, fuzzy unification and spreading of parameters during inferential process). To highlight the applicability of KMSFL, we present in *Section 6* an extended case study relative to a balancing problem for a flexible manufacturing system (both for classical and fuzzy case), formulated as follows: the flexible manufacturing system consists of a set of components (machines, subsystems, etc.) connected so that they can transfer different amounts (parts) of material through bond wires, directed according to a given structure scheme. Such a system can be represented through a graph (M, A), where M={1, ..., N} is the set of identical components in its structure and A⊂M×M. Assume that (M, A) is strongly connected, i.e. for $(\forall)$ i∈M there is a path from i to $(\forall)$ j∈M and moreover, if (i, j)∈A, i≠j). Each component has a quantity of material that can be processed. We assume that any quantity of component i, denoted by $x_i \geq 0$, for $x_i \in N_+^*$ or $\mu_{xi} \in R_+^*$ (for the fuzzy case), can be partially transferred to component j. The control expert system for this problem area must be able to transfer the whole amounts of material (discrete case) or fractions of the quantity x_i (continuous case) from component i to another component j, if there exists (i, j)∈A. For the crisp case, we used the same control model also tested with G2 generator. This case allowed us to synthesize the fuzzy control model, due to the serious limitations outlined in its previously conducted crisp tests. Finally, s*ection 7*, concludes and introduces the future research work in order to synthesize KMSs more integrated into the NSTs.

3. KMSFL in a distributed control structure

KMSFL, as a control agent, is an expert system that integrates imprecise knowledge. The reasoning specific to this agent is performed by the inference engine based on fuzzy logic. This agent's architecture follows the general characteristics of any agent like in (Jennings & Wittig, 1992). However, there are several new elements which we emphasize in fig. 1.

It is noted that planning and coordination module (PC_{io}) comprises three levels: the level of evaluation of the global situation, the meta-planning level and the level of coordination of cooperation. Its main functions are dedicated to monitoring the activity of other agents, choosing the communication protocol, analyzing whether to respond to other agents and detecting global conflicts. The major difference between the monitoring function of PC_{io} and

the monitor is that the monitor supervises the activity of its own system based on fuzzy knowledge. The monitor is the only link between the system based on imprecise knowledge and the ISICMA$_{io}$ interface, and implicitly between the system based on imprecise knowledge and the whole multi-agent control system. Its basic role is of remote control of its own system based on fuzzy logic, meaning that the monitor will communicate through messages, because it is not a part of the AI system based on imprecise knowledge. Decisions concerning the control of the intelligent control system based on fuzzy knowledge are made at the level of Monitor$_{io}$, whose main functions are: retrieving information from various sources, granting access to MS$_{io}$, evaluating the local situation, updating the local information, describing high-level goals, planning the decisions and actions. In order to eliminate the possible conflicts between these functions and the ones of the AI system based on imprecise knowledge, Monitor$_{io}$ invokes the low-level behaviors stored in MS$_{io}$.
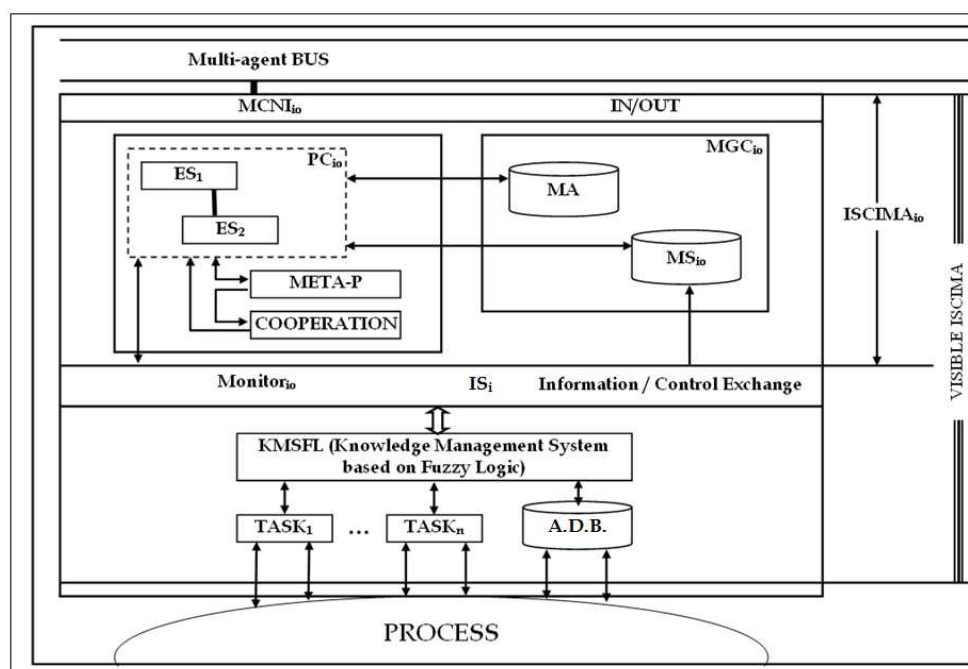

Fig. 1. KMSFL in a multi-agent structure

Fuzzy logic is a generalization of bivalent logic, replacing the discreet nature of the latter with one of continuous nature. While in bivalent logic, in order to demonstrate the validity of formulas, are used methods that use up all the possibilities of evaluation according to the interpretation function, in fuzzy logic this is no longer possible. A special feature of human reasoning is the effective use of natural language, even in the logical reasoning. According to this observation, we may conclude that the mathematical model of how a man thinks during a control process and at a certain level of decision synthesis may be based on fuzzy logic combined with modal temporal attributes (Mazilescu, 2010). Approximate reasoning theory, as a methodology for exploiting imprecise knowledge relative to the control expert system's state (denoted $x^{CES} \in X^{CES}$ and represented in the form of possibility distributions), allows, by means of logical inferences, to obtain rigorous characterizations of the values of linguistic variables within the structure of state x^{CES}, according to the control goal (Mazilescu, 2009a). The set X^{CES} can be defined as a Cartesian product $X^b \times X^{int} \times X$, where $x^b = [x_1^b, x_2^b, \ldots x_{k_1}^b]^t \in X^b$. For example, the component x_1^b indicates, through its values, the

potential command events for the process, $x_i^b \in U^{(i)}$, i=2, ..., k_1, where $U^{(i)}$ are the universes of discourse attached to linguistic variables $x^{(i)}$ (chosen to characterize the state $\mathbf{x}^{ES} \in X^b \times X^{int}$), X^{int} is the set of internal states of the inference engine and X is the set of states of the process. Thus, we can model KMSFL and the corresponding reasoning as a possibilistic expert system, that allows us to characterize a state $\mathbf{x}^{CES} \in X^{CES}$ based on imprecise information relative to the state $\mathbf{x}^{CES}$, i.e. with a subset $E \subseteq X^{CES}$, for which $\mathbf{x}^{CES} \in E$. We assume that there may be components of state $\mathbf{x}^{CES}$, defined as predicates that have firm truth values. Moreover, in this case is met the condition of membership of truth values to the interval [0,1] and thus we can work only with interval [0,1]. The control expert system manages knowledge specific to a state of the closed-loop system $x^{CES} \in X^{CES}$, characterized at time k by $x_k^{CES} = (x_k, x_k^{ES})$. The class of possibilistic expert systems can encompass also temporal reasoning. In this case, the rules base consists no longer of relationships, but of multi-dimensional possibility distributions to which are attached temporal descriptors, modeled in turn by means of possibility distributions, in order to attach the fuzzy statements to temporal features. Attaching the fuzzy temporal descriptors is specific to AI techniques, while in terms of control, this corresponds to fuzzifying the moments of time in the theory of discrete event systems.

4. Defining KMSFL as a discrete system based on logical events

Temporal aspects are important in areas such as planning, qualitative simulation, cognitive modeling, and natural language semantics. There are two basic approaches to the integration of temporal aspects in terms of AI logical systems: **i)** first-order logic can be directly used to formulate statements that contain symbols for time positions; **ii)** first-order logic can be extended with modal operators. Introducing the temporal aspects aims at designing the means for solving the meta-equation:

$$time = complexity \oplus real\_time \oplus temporal\_reasoning \qquad (1)$$

which was proposed and used for integrating time in an application of AI, dedicated to technological process control. This meta-equation will be particularly applied for the inference engine developed, which is able to exploit knowledge specific to control applications. Operator $\oplus$ is an aggregation symbolic meta-operator, which can be instantiated in different classes of specific operators. Relation 1 can be viewed as a meta-equation as it contains several variable elements, such as: i) the first term has as variable the time component defined in the algorithms' analysis, and the compiled knowledge structure specific to KMSFL system tries to improve exactly this value; ii) the second term includes the time variable in the form of the length of the control expert system's inferential chain, which depends on how the events occur; iii) the last term was included to highlight the possibility of adding temporal attributes, which was not clearly necessary for KMSFL system, at least not for the case study solved. We define all elements describing KMSFL as a Control Expert System (CES). Defining the model for KMSFL as a closed-loop system is important, if we consider some of its features (Passino & Antsaklis, 1994). At this level, we emphasize a series of features of the control expert system, the elements of qualitative analysis of the control expert system, as well as a crisp version of a simplified example, to demonstrate the theoretical aspects.

Definition The control expert system (CES) is a formal system, defined as:

$$CES = (X^{CES}, E^{CES}, f_e^{CES}, \delta_e^{CES}, g^{CES}, x_o^{CES}, E_v^{CES}) \qquad (2)$$

where: $X^{CES} = X \times X^{ES}$ is the set of states x^{CES} of the control expert system

$E^{CES} = E_u \cup E_d^{CES} \cup E_o$, with $E_d^{CES} = E_d \cup E_r^{ES} \cup UI$. Sets E_u, E_d, E_r^{ES} and UI are the input events of the control expert system and E_o the output events of the process;

$g^{CES} : X^{CES} \rightarrow \mathcal{P}(E_u \cup E_d^{CES}) - \{\varnothing\}$ is the activation function of the CES;

$f_e^{CES} : X^{CES} \rightarrow X^{CES}$, $e \in \mathcal{P}(E_u \cup E_d^{CES}) - \{\varnothing\}$ is the set of state transition functions;

$\delta_e^{CES} : X^{CES} \rightarrow E_o$, $e \in \mathcal{P}(E_u \cup E_d^{CES}) - \{\varnothing\}$ is the output function; $x_o^{CES} \in X^{CES}$ is the initial state; $E_v^{CES} \subset E^{CES}$, where E^{CES} is the set of all trajectories (finite or not) of events in closed loop, which can be generated by the control expert system, based on g^{CES} and f_e^{CES}, and E_v^{CES} is the set of all trajectories of allowed events in closed loop (a subset of trajectories of events that may result, knowing the trajectories of events of the process and of the expert system, connected together). Thus, E_v and E_v^{CES} are viewed as some restrictions in the structure of E_v^{CES}. Based on the trajectories of allowed events, we can highlight additional restrictions which are possible sequences of events within the closed loop system. The output events of fuzzy expert system are considered, in this case, logical events. These issues are particularly important for the future possibility of a qualitative analysis of the closed-loop system (admissibility, cyclical behavior and stability). Conventional knowledge-based systems can ignore the dynamic behavior of the control expert system caused by user inputs and process outputs. Many expert systems' evaluation is done either through difficult simulations, or by comparing its behavior with the one of human experts. The fuzzy control expert system is like a planner, because it can predict a number of states in the evolution of the process. An expert system (fig. 2) should be designed to remove unwanted behaviors of closed-loop system. Initial state x_o^{CES} is necessary both from theoretical and practical considerations. From the theoretical point of view, it is the beginning of a formal system, in which derivation relations will be the inferential processes, and from the practical point of view, it is necessary to define the initial state of the control expert system in order to reduce possible unwanted combinations of states that could unduly complicate the model. If the initial state of the closed loop system is known, the state transitions can be restricted to the acceptable states of the system. Once specified the initial state $x_o^{CES} = (x_0, x_o^{ES})$ for the state $x_k^{CES} = (x_k, x_k^{ES})$ at time k, we get, based on the definition of activation function $g^{CES}(x_k^{CES})$, the following form:

$$g^{CES}(x_k^{CES}) = [g^{ES}(x_k^{ES}) \cap E_r^{ES} \cap UI] \cup \delta^{ES}(x_k^{ES}) \cap g(x_k) \cap HC] \cup [g(x_k) \cap E_d], \qquad (3)$$

where:

i. $g^{ES}(x_k^{ES}) \cap E_r^{ES} \cap UI = (\mathcal{P}(E_1^{ES} \cup \mathcal{R}) \setminus \{\varnothing\}) \cap E_r^{ES} \cap UI$ is the set of input and internal events of the expert system, allowed for the state x_k^{ES};

ii. $\delta^{ES}(x_k^{ES}) \cap g(x_k) \cap HC = E_0^{ES} \cap \mathcal{P}(E_u \cup E_d) \cap HC$ is the set of command input events of the process, allowed for the state x_k^{ES};

iii. $g(x_k) \cap E_d = (\mathcal{P}(E_u \cup E_d) \setminus \{\varnothing\}) \cap E_d$ is the set of input disturbance events of the process, allowed in the current state x_k of the process.

The input events allowed in the process for the control expert system in closed-loop are the events allowed both by the current state of the process $\mathbf{x}_k$ or by the states x_k^{ES}, and the events caused by input disturbances of the process, allowed in the state $\mathbf{x}_k$. The expert system must control only the activation of the events of type $E_u \cup HC$. It is built in such a way that its transition from a current state in a future state is achieved in response to any output event of

the process. Notice that the control expert system's dynamic must be properly defined, even for the particular case in which the expert system also comprises imprecise knowledge, and its reasoning must reflect the temporal characteristics, specific to process (according to relationship 3). Assume that $e_k \subset g^{CES}(x_k^{CES})$ is an allowed event of the control expert system in closed loop, currently in the state $x_k^{CES} = (x_k, x_k^{ES})$. Under these conditions, if the events $e_{r_k}^{ES} \in g^{ES}(x_k^{ES}) \cap E_r^{ES} \cap UI$, $e_{uk} \in \delta^{ES}(x_k^{ES}) \cap g(x_k) \cap HC$ and $e_{dk} \in g(x_k) \cap E_d$, then e_k can be defined in various ways. These depend on the type of input command events of the process, or on the type of disturbance events, which may occur simultaneously. There is thus a finite number of ways $i_0 \in N^*$, so that for $(\forall)$ $i \in \{1, ..., i_0\}$, e_k^i can be properly defined. Corresponding to each type of event e_k^i, $i = 1, ..., i_o$, then:

$$f_{e_k^i}^{CES}(x_k^{CES}) = x_{k+1}^{CES}, \text{ where } x_{k+1}^{CES} = (x_{k+1}, x_{k+1}^{ES}).$$
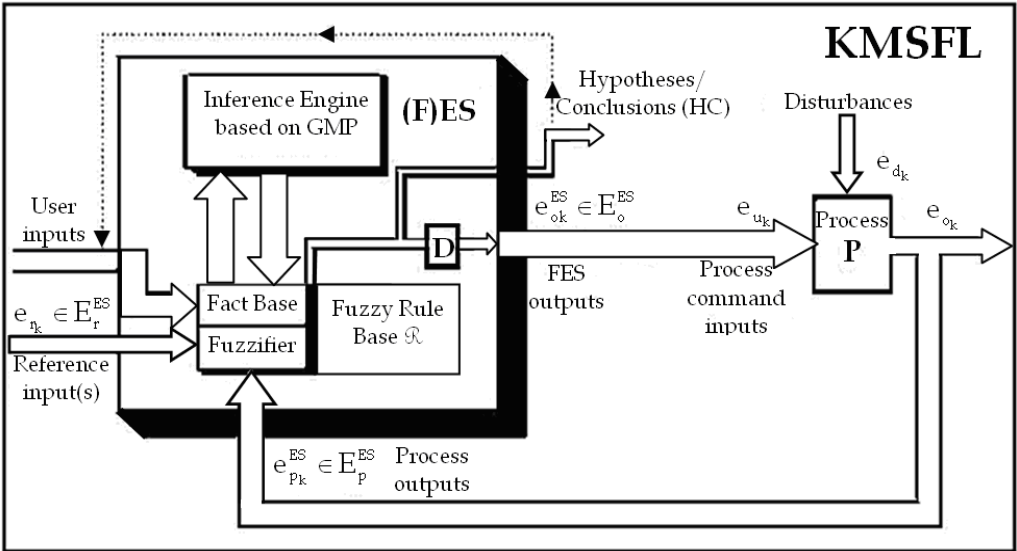

Fig. 2. The basic architecture for KMSFL

5. The characteristics of KMSFL as a fuzzy expert system

Knowledge representation and exploitation within an expert system are rather conflicting characteristics, whereas the increase in knowledge representation power reduces system's efficiency and increases the difficulty of developing it. Many AI problems are difficult to solve from the computational point of view. An observation which may help to reduce this complexity is that often these problems have the following property: inputs can be divided into two parts, of which, a part is relatively constant long time, compared with the second part. In such situations, seems right to make some changes in the constant part, in order to reduce the time of obtaining the solution for the AI problem, if the second part varies, but is known at certain moments of time. Transformations made in advance are called pre-processing or knowledge compilation. Using variables in an expert system allows knowledge factorization. First order predicates language facilitates expressing complex knowledge rigorously, imposing appropriate reasoning techniques. Definition of certain propagation and inference procedures for real-time expert systems, involves the development of powerful reasoning mechanisms, as well as adapting the control algorithms

to the state spaces, which are often very large. KMSFL is conceptually based on all the properties summarized above and consists of the compiled fuzzy rules base (control model) and the inference engine. In order to highlight the characteristics of this system, the following elements must be described: i) *The formalism,* specifying the types of knowledge supported by the system. Are presented, in order, the fuzzy knowledge syntax and the basic features of the compiled linguistic models, system's parameters, the elementary fuzzy filtering, the compatibility of possibility distributions for GMP inference scheme; ii) *Compiler properties,* which include the static structural discrimination component of the fuzzy state x^{CES}, the fuzzy unification tree as the basic element of the structure of compiled fuzzy knowledge (aimed at checking the consistency of fuzzy substitutions), the algorithm that generates the variables linking network.

Network's terminal nodes correspond bijectively to the fuzzy rules; iii) *Inference engine algorithm based on fuzzy logic,* which includes techniques for reasoning in the presence of compiled imprecise knowledge.

As a first step in the practical implementation of an expert system, knowledge representation aims to describe the problem domain as a model that includes relational entities and symbols, according to an appropriate formalism. The types of knowledge accepted by KMSFL are: **i)** variables (symbols always preceded by '?', such as ?x, ?y, and which will occur only in rules); **ii)** atomic constants (numbers or strings); **iii)** possibility distributions or fuzzy constants (symbols always preceded by the character '*' and used to represent imprecision); **iv)** logical operators. Possibility distribution can take any form. This complexity can cause a number of difficulties for the application of possibility theory. In practice, when the variable is numeric, it appears that a trapezoidal possibility distribution on continuous referential is well suited. It can be represented through four parameters (g, d, φ, δ). The trapezoidal form of possibility distributions is preserved in most of the inference and calculation operations. All the fuzzy constants used in knowledge representation and modeling, for the synthesis of fuzzy reasoning algorithms, are represented by trapezoidal possibility distributions, such as g≤d, φ, δ≥0, called T-numbers. Fuzzy constants can occur both in facts and rules, and are always associated to fuzzy sets (T-numbers) through *constfaz* function. Within KMSFL, we can equate the fuzzy set to a fuzzy constant. Undefined fuzzy constants are not allowed.

A fuzzy constant has always a value corresponding to a continuous, trapezoidal and normalized fuzzy set. Using possibility distributions provides an unified framework for representing imprecision and uncertainty. Parameter ζ is used to measure fuzzy sets' uncertainty (0≤ζ≤1). If a fuzzy set is uncertain, parameter ζ must be defined in *constfaz* function through a list (uncertain ζ). We admit that a completely uncertain fuzzy set (ζ=1) has no effect on system's behavior. In contrast to facts, a motive is a structured list in which variables may occur. This indicates the presence of variables, atomic constants and of fuzzy constants within motive's structure. In addition, the motives may occur in both the conditional part and in rules' conclusion. Uncertainty is allowed in the conditional part and in the consequent of GMP inference scheme, only if a particular linguistic model requires it. In order to increase the knowledge representation capacity, are introduced *predicates* that appear as motives in the left side of the rules. We emphasize the presence, within the knowledge model, of predicates F(a), which are flags that emulate human reasoning sequences, in order to achieve the control expert system's goal. The knowledge representation formalism for KMSFL is:

Antecedent	:: = condition*
Condition	:: = motive | motive motive_index | predicate
motive_index	:: =\<atomic constant\>
motive	:: = expression in which are allowed the three types of data;
predicate	:: = (predicate_sym predicate_arg predicate_arg)
predicate_arg	:: = predicate | atomic constant | fuzzy constant | variable
predicate_sym	:: = {=*, *\<, \>*, *Π, *N}
consequent	:: = conclusion*
conclusion	:: = motive | motive motive_index | predicate | procedure

Predicates =*, *< and >* have binary values, while *Π, *N are fuzzy predicates. If fuzzy linguistic models are introduced in a expert system, it becomes more complex due to considering the fuzzy processing at all system's levels, such as: fuzzy filtering, imprecise sets compatibility, fuzzy unification, calculation of the inferred conclusion together with the calculation of spreading for the parameters that manage imprecision, selection of strategies in which are naturally embedded also elements of factual knowledge imprecision. The fuzzy pattern-matching aims to determine the instantiations set of the causes. It is stronger than the classic one due to its capacity of processing the fuzzy knowledge. It is a matter of evaluating the degree of this pattern-matching between a fuzzy cause and a fuzzy fact (the fact filters more or less the cause). In order to put a fact in relation with a cause, we can build up a recursive algorithm, comparing the two associated trees step by step. It follows beyond doubt that the knowledge pattern-matching is the basic operation. Generally speaking, it is a matter of pattern-matching between a model P and a data D, to which we attach μ_P respectively Π_D ($\mu_P(u)$ is the degree of the compatibility between the value u and the meaning of P, while $\Pi_D(u)$ is the degree of possibility that the value u represents the value of the attribute which describes an object modeled through the data D). The degree of compatibility has the membership function $\mu_{P|D}$ defined through the extension principle. Though it translates relevant information related to the degree of the pattern matching between P and D, it is difficult to use $\mu_{P|D}$. We prefer two scalar measures in order to evaluate the compatibility: $\Pi(P,D)$ and $N(P,D)$. Let us consider the most simple case $((*f, *m \rightarrow *c),*c')$, where *m is the cause of the rule, *m→*c, *f is the fact, each of them being expressed by fuzzy sets. In order to deduce the conclusion *c', it is to be known if the fact is compatible with the rule condition. We can try to calculate GMP for the inferred conclusion *c'. The theory of possibilities provides two measures, which are very useful to evaluate the compatibility of the fuzzy sets:

$$\Pi(*m,*f)=\sup_u \min(\mu_{*m}(u), \mu_{*f}(u)), N(*m,*f)=1-\Pi(\neg*m, *f)=\inf_u \max(1-\mu_{*m}(u), \mu_{*f}(u))$$

Generally, it is much complicated to calculate N than Π. A simple calculating method is based on the separation of the complementary of *m. Analyzing the form of ¬*m, we find that this can be divided into two fuzzy sets L_s and L_d. The fuzzy set $L_s=(-\infty, g_m-\varphi_m, -\infty, \varphi_m)$ is always on the left of *m, while $L_d=(d_n+\delta_m, \infty, \delta_m, \infty)$ is always on the right of *m, and $L_s \cap L_d = \varnothing$. It follows that ¬*m=max(L_s, L_d). We get:

$$N(*m,*f)=1-\Pi(\neg*m,*f)=1-\Pi(\max(L_s,L_d),*f)=1-\max(\Pi(L_s,*f),\Pi(L_d,*f)).$$

Having Π and N, defined and calculated this way, we distinguish several classes of decreasing compatibility. Even if the measures Π and N correctly estimate the degree of compatibility between the fuzzy constants, these measures cannot be used directly to infer

the conclusions in the case of an inference engine based on GMP. If the measures Π and N satisfy some thresholds, then the pattern matching is successful. To calculate GMP we need the parameters θ and K, in the following form:

$$\theta=(*m,*f)=\max(\mu_{*f}(g_m-\gamma_m),\mu_{*f}(d_m-\varphi_m)), \quad K=(*m,*f)=\min(\mu_{*m}(g_f),\mu_{*m}(d_f))$$

At the end of the fuzzy condition/fact pattern-matching stage for the cause C and the fact F, if the degrees of the pattern matching satisfy the chosen thresholds and if there is a consistent substitution σ, then the pattern matching is successful. The substitution σ is a particular case when the variables in the causes can be associated to some fuzzy constants present in the facts. If the instance $\sigma \cdot C$, obtained through the application of the fuzzy substitution σ to the condition C, is not totally equal with F, i.e. the expression $F=\sigma \cdot C$ is not always true, then σ is fuzzy. We can take into account the problem of finding the proper thresholds for measures Π and N in order to determine the facts that do not filter the causes at all. The choice is not made randomly, as between the two parameters of GMP it must be a tight link. Because of all these remarks and in order to correctly solve the problem, there are the links between Π, N, θ, K.

Fuzzy variables linking. The fuzzy condition/fact pattern matching is the first stage in the running of the inference engine, which takes into account the imprecision. After this stage, it results a lot of instantiations of the causes. Each motive's instantiation will be associated to a fuzzy substitution and to the four parameters Π, N, θ, K. The second stage is represented by the linking of the variables and it aims to determine the consistent instantiations at the level of rules' full conditions.

Fuzzy unification. The purpose of the fuzzy unification is to verify the consistency of the fuzzy substitutions, where the variables can be associated with fuzzy sets. Let's consider a rule $(*D\ *H\ ?x)\ (B\ ?x)\rightarrow(act(C\ *E\ ?x))$. In the antecedent of the rule there are two causes $C_1=(*D\ *H\ ?x)$ and $C_2=(B\ ?x)$. We suppose the facts to be specified: $F_1=(*d_1\ *h_1\ *w)$ and $F_2=(B\ *r)$. For certain chosen fuzzy sets, the fuzzy constant $*d_1$ filters $*D$ and $*h_1$ filters $*H$. The only result for the pattern-matching between C_2 and the fact F_2 is the fuzzy substitution $\sigma=(*\ /?x)$ and the pattern-matching parameters. If all the parameters satisfy the designed thresholds, then the facts unify totally with the causes. After the fuzzy condition/fact pattern-matching, we obtained two fuzzy substitutions: $\sigma=\{*w/?x\}$ and $\sigma = \{*r/?x\}$, where $*w$ and $*r$ are fuzzy sets. The fuzzy unification contains, on the one hand, the evaluation of the consistency degree of the fuzzy substitutions on a certain norm and, on the other hand, the fuzzy substitutions composition.

Let us consider a rule R with k conditions, under the form $COND(R)=(C_1,...,C_k)$. After the fuzzy condition/fact pattern-matching, if each condition C_i, filters a fact F_i, then there are a fuzzy substitution σ_I, so that $F_i=\sigma_i \cdot C_i$, and the four parameters Π_i, N_i, θ_i, K_i. Let us consider a variable $?v$ within the rule; assume that it appears n times in the conditional part of the rule. $?v_i$ is used for the representation of ith of the variable $?v$. In this case, all the occurrences of the variable $?v$ within the global condition of the rule can be represented through the following list: $\{?v_1,\ ?v_2,...,?v_n\}$. Each $?v_i$ will be certainly associated with a term t_i, which can be an atomic or a fuzzy constant, denoted: $\{t_1/?v_1,\ t_2/?v_2,..., t_n/?v_n\}$. All the various variables present within a rule are independent. Each variable can occur in a rule several times. Each occurrence of the variable is independent of the other occurrences. Nearly all expert systems preserve this hypothesis. The fuzzy unification consists of: **i)** The consistency verification of the element in list $\{t_1/?v_1,\ t_2/?v_2,...t_n/?v_n\}\rightarrow\{t_p/?v_p\}$, as against a certain norm; **ii)** The

composition of the fuzzy substitutions. In order to eliminate any confusion, $?v_p$ is used to represent the variable $?v$ after the fuzzy unification. Finally, the fuzzy unification can be represented through the following expression: $\{t_1/?v_1, t_2/?v_2,...t_n/?v_n\}\{t_p/?v_p\}$, where t_p is going to be calculated. Let us consider a simple case. If t_i is a fuzzy set, i.e. $t_i={*t(i)}$, $(i=1,2)$, then the symbolic or numerical comparison is no longer sufficient to evaluate the consistency between $*t(1)$ and $*t(2)$. When $?v_1$ and $?v_2$ are independent, the cartesian product $*t(1) \times *t(2)$ is defined by:

$$*t(1) \times *t(2)=\{((x_1, x_2), \mu_{*t(1) \times *t(2)}(x_1,x_2)/x_1 \in X_1, x_2 \in X_2, X_1, X_2 \subset R\}, \mu_{*t(1) \times *t(2)}(x_1,x_2)$$
$$=\min(\mu_{*t(1)}(x_1)_{*t(2)}(x_2))$$

The compatibility between $*t(1)$ and $*t(2)$ can only be clarified through a reasonable explanation of the criterion relative to which compatibility is judged. In the classic situation, the criterion is given by the equality relation. It is quite natural to introduce appropriate criteria for fuzzy unification in both stages: to check the consistency and to make up the fuzzy substitutions. These criteria should be more general; the equality relation can be defined through a binary fuzzy relation R. Defining the fuzzy set $*t(1)$ and the relation R, we obtain $\mu_{R \circ *t(1)}(x_2)$, defined by:

$$\mu_{R \circ *t(1)}(x_2)=\sup_u \min(\mu_R(x_1,x_2),\mu_{*t(1)}(x_1))$$

Since we know both the relation R and the Cartesian product $(t(1) \times *t(2)$, we can use measures Π and N to estimate the consistency of fuzzy sets $*t(1)$ and $*t(2)$ relative to R. Thus we have:

$$\Pi(R, *t(1) \times *t(2)) = \sup_{x1,x2} \min(\mu_R(x_1,x_2),\mu_{*t(1)}(x_1),\mu_{*t(2)}(x_2)),$$

$$N(R, *t(1) \times *t(2)) = \inf_{x1,x2} \max(\mu_R(x_1, x_2), 1-\mu_{*t(1)}(x_1),1-\mu_{*t(2)}(x_2))$$

It is interesting to note that the fuzzy binary relation R, can be interpreted in various ways. The equality relation may be regarded as a particular case of relation R. A last important problem is the parameters spreading. At the end of the elementary fuzzy pattern-matching stage, if the pattern-matching degree satisfies the chosen threshold and if there is a consistent substitution σ, then the pattern-matching process is successful. The fuzzy condition–fact pattern-matching process is the first stage, part of the overall cycle of the inference engine, able to take into consideration the imprecision. Each instance of a fuzzy motive is associated with a fuzzy substitution σ and with the parameters Π, N, θ, K. Following all these remarks and in order to correctly solve the problem, there are the links between Π, N, θ, K. As already shown, GMP verifies the following proposition:

Proposition i) $K = 0 \Leftrightarrow \theta = 1$; $K > 0 \Leftrightarrow \theta < 1$; ii) The conclusion $*c'$ inferred through GMP is uncertain: $(\mu_{*c'}=1) \Leftrightarrow \theta = 1$; iii) $N(*m, *f) > 0 \Leftrightarrow \theta < 1$.

The second stage in the pattern-matching process, on a global scale of the fuzzy rules, is the fuzzy linking of variables. This conducts the fuzzy unification, whose main purpose is to verify the consistency of fuzzy substitutes, for which we have already presented a series of theoretical results. Using the tests present in the linking nodes, we can build a dynamic tree that allows adding or suppressing facts. Within each test node of this tree, the values of the variables are tested. If two facts go on the same path, then it is possible that the two facts are consistent. We may use this tree in order to avoid combination challenges. This tree is called *linking tree* and it is associated to the linking nodes. It has some difficulties for the

discrimination of the fuzzy sets within its linking nodes, since certain parts (leaves) of the tree may contain multiple fuzzy facts.

That is why the efficiency of this solution decreases, being similar to the use of the unification tree, in order to discriminate the fuzzy motives. The main inefficiency factor is related to the disorder of the fuzzy facts in the tree leaves.

In order to improve this situation, we may use the characteristics of the fuzzy sets in order to sort out the facts. This approach was used to adapt the unification tree for the processing of fuzzy motives. The major difference that appears between the two situations is the fact that the unification tree is a static tree, i.e. the discriminator motives do not change, whereas the linking tree is dynamic, the discriminator facts being updated during the functioning period of the inference engine.

The fuzzy variables linking process consists of the fuzzy unification and the spreading of Π, N, θ, K parameters, evaluated on a global scale of the antecedent of the rule. We will further insist on the parameters spreading process, obtained at the end of the filtering stage, in the consequent of the rules. The Π possibility and N necessity degrees represent the extent to which a rule is satisfied in the current state within the facts base. During the selection stage, the system selects the rule that satisfies best these conditions in order to activate it, and the parameters θ and K serve to applying the GMP inference scheme. In the conclusion part of a rule there may be multiple motives (some may be added, others may be deleted, once the rule has been activated).

6. Case study

For the synthesis of fuzzy knowledge control model for the flexible production system, we considered the loads as fuzzy T-numbers, expressed linguistically as "around" or "approximately", and we introduced intermediate variables in the knowledge model's structure, such as: global balance degree *bd* (for the whole flexible manufacturing system) with the fuzzy values *satisfactory* and *unsatisfactory*, partial balance degrees pbd_i, i = 1, ..., 5 (per groups of subsystems), corresponding to certain situations unresolved in the crisp model, and fuzzy variables d_{56}, d_{42}, d_{13}, d_{21}, d_{43} and d_{35}, whose values can be the fuzzy T-numbers *small, large* or *zero*. The currently used partial balance degrees have as imprecise characteristics the linguistic values *good* and *unsatisfactory* (*unsatisfactory* value, in this case, is similar, in fuzzy number, to the one associated with the linguistic value *unsatisfactory* of the global balance degree). It was also useful to introduce knowledge in the form of imprecise facts of type (X1 X2 X3 X4 $?v_q$), (X1 X2 X4 $?v_q$), (X3 X4 X5 X6 $?v_q$), (X1 X3 X4 X5 X6 $?v_q$), (X2 X4 X5 X6 $?v_q$), ensuring that the balancing is continuously accomplished, partially and gradually, and a number of *meta-rules* in which these facts are present. They will be activated chained, as (R13, R14) (R13, R15) (R16, R17) (R16, R24) (R18, R19) (R18, R25) (R20, R21) (R22, R23)) and support the synthesis of fuzzy decision. We may notice that, for this control model, we followed the stages for the synthesis of a linguistic model, in terms of knowledge acquisition process, and the knowledge and meta-knowledge are represented according to KMSFL formalism.

Exploiting this model requires calculating the fuzzy unification and the partial conclusion inferred through the GMP scheme, the call of procedures in rules' consequent and of a control module, dynamically updating the priorities of the rules according to the current imprecision of all knowledge involved at some point in the decision synthesis, chaining the fuzzy meta-rules and demonstrating the global asymptotically stable behavior of the closed-

loop system. Intermediate fuzzy variables pbd_1, pbd_2, pbd_3, pbd_4 and pbd_5 are degrees of partial balance between groups of subsystems (1, 2, 3, 4), (1, 2, 4), (3, 4, 5, 6), (1, 3, 4, 5, 6) and (2, 4, 5, 6). These variables were used to eliminate the unbalance situations obtained in the crisp case, these being the control meta-knowledge of the expert in the field. They depend essentially on the structure of flexible manufacturing system and are actually imprecise knowledge used in decision synthesis. For each inference cycle it must calculate all the corresponding fuzzy values and generate the updated facts, thus, the current state of the control system.

The fuzzy control model has the following properties: *i*) variable pbd_1 appears in rules 14 and 15, and its value is recalculated after the pattern-matching between its current fuzzy value and the motive (pbd_1 *b) by applying the GMP inference scheme. The same calculation method is applied to other partial balance degrees; *ii*) for meta-rules 14, 16, 18, 20 and 22 fuzzy unification is applied differently, requiring the calculation of the resulting fuzzy value for the variable $?v_q$ from the consequent; *iii*) variable $?v_q$, together with variables pbd_1, pbd_2, pbd_3, pbd_4 and pbd_5, allow chaining the fuzzy rules while the control model is used by the KMSFL inference engine; iv) within the structure of meta-rules 13-25 are present fuzzy variables d_{56}, d_{35}, d_{13}, d_{21}, d_{43}, d_{42}, representing the fuzzy instant differences between machine loads, referenced by the corresponding indexes, in the appropriate order. The values of these variables are fuzzy numbers tp *ma(2 pbd_j 1 1), j=1 ... 5 and tp *mi(0 1 0 1). Fuzzy value *b is generated similarly. To clarify how the fuzzy model for flexible production system is initialized and exploited, follow the *Algorithm_KMSFL*, in which are also presented some functional aspects of the control expert system.

This algorithm associated with the inference subsystem within the KMSFL was developed in close relation with fuzzy knowledge representation formalism. In addition, we tested the knowledge model using the generator G2 (only for the crisp case) to highlight some of the advantages and limitations of the knowledge management model and of the inference engine. However, fuzzy processing is specific to the expert system developed, this being the major objective of this work.

Algorithm_KMSFL

1. Enter the initial fuzzy loads $\mu_{x_{i_0}}$;
2. Calculate the expected average value $\mu_{c_0} = \sum_{i=1}^{6} x_{i_0} / 6$, the fuzzy distances $\mu_{d_{i_0}} = \mu_{x_{i_0}} - \mu_{c_0}$ and the initial balance degree $bd_0 = \max_{i=1,\dots,6} \{ \mu_{d_{i_0}} \}$;
3. Generate sets *s (the characteristics of fuzzy sets *s are checked in tests, in relation to the initial loads), *n = tp(3 bd_0 2 2) (d and δ are checked in tests in relation to the initial loads) and the facts (bd *n) and (bd *s);
4. Initialize $x^b=(x^{b1}, x^{b2})$ where $x^{b1}=0$ and $dim(x^{b1})=12$, x^{b2} contains the initial facts attached to all linguistic variables involved in the control model, such as: ((X1 ?x), (X2 ?y), (X3 ?z), (X4 ?v), (X5 ?w), (X6 ?ξ), (X1 X2 X3 X4 $?v_q$), (X1 X2 X4 $?v_q$), (X3 X4 X5 X6 $?v_q$), (X1 X3 X4 X5 X6 $?v_q$), (X2 X4 X5 X6 $?v_q$), (X1 X2 X4 $?v_q$), (X12b 1), (bd *n), (CFS *zero), (pbd_1 *b), (d_{56} *ma), (d_{35} *ma), (pbd_2 *b), (d_{13} *ma), (pbd_3 *b), (d_{21} *ma), (pbd_4 *b), (d_{42} *ma), (pbd_5 *b), (d_{43} *ma), to evaluate the effect of all fuzzy variables that appear in structure of this state component of KMSFL. As initial fact attached to the reason (bd *n), is launched the fact (bd $*v_0$), where $*v_0$ is generated as a fuzzy set around the value of bd_0, of the form (*constfaz* $*v_0$(tp bd_0-1 bd_0+1 2 2)). Is also initialized $x^{int} = 0$.

5. In the consequent of rules R1-R10, along with the deduction of fact (xib i), corresponding to a procedure call, is attached the calculation of the balance degree bd with its new fuzzy value, i.e. is generated the new fact (bd $*v_k$), $k \geq 1$. Value $*v_k$ can filter or not $*n$ or $*s$, and properly the fuzzy values of partial balance degrees and the fuzzy distances between the physically interconnected subsystems;

6. If the inference engine stops on a different event than on the one corresponding to the activation and execution of rule 11, then are recalculated the fuzzy differences $\mu_{d_{i_k}} = \mu_{x_{i_k}} - \mu_{c_0}$, along with determining the subsystem (i = 1, ... 6) and the corresponding rule j (j = 1 ,..., 25) to be activated, to satisfy the balancing goal at the current time, by using meta-rules (R13-R25) or by launching the *control module*.

For each inference cycle must calculate all the corresponding fuzzy values, and generate the updated facts and thus the current state of the control system.

Implementation results for KMSFL

Test A. This test aims to prove the functioning of the KMSFL prototype for a first set of knowledge. Fuzzy values involved in testing the control model by using the corresponding inference engine are: (*constfaz* $*s$(tp 0 1 0 1)), (*constfaz* $*n$(tp 2 ∞ 1 1)), (*constfaz* $*ma$(tp 2 ∞ 1 1)), (*constfaz* $*mi$(tp 0 1 0 1)), (*constfaz* $*b$(tp 0 1 0 1)), ε=1, η= 0, $\mu_{x_1} = \mu_{x_2} = \mu_{x_3}$=(199.5 200.5 1 1), $\mu_{x_4} = \mu_{x_5} = \mu_{x_6}$=(-0.5 0.5 1 1), "around x_0"=(x_0-0.5 x_0+0.5 x_0-1.5 x_0+1.5)="x_0", x^{b1}=**0**. We get the following results: the first complete execution of the inference engine implies a number of 118 inferences and the values of subsystems' loads and of the global balance degree are: μ_{x_1}="101.33"=(100.83 101.83 0.5 0.5), μ_{x_2}="102.60"=(101.56 102.56 0.5 0.5), μ_{x_3}="101.33"=(100.83 101.83 0.5 0.5), μ_{x_4}="102.04"=(101.54 102.54 0.5 0.5), μ_{x_5}="100.61"=(100.11 101.11 0.5 0.5), μ_{x_6}="92.627"=(92.127 93.127 0.5 0.5), bd=(6.87 7.87 0.5 0.5). In this case, the balance degree bd="7.3731" is an unsatisfactory value, which means that the balancing problem is not completely solved (fig. 3, fig.4). For inferences 42, 43, 54, 55, 58, 59, 70, 71, 82, 83, 86, 87, 95, 96, 100, 101, 103, 104, 105, 106 are activated meta-rules 16, 17, 13, 14, 13, 15, 16, 17, 13, 14.13, 15, 16, 17, 16, 17, 13, 14, 13, 15 respectively, the rest of the inferences activating the basic rules R1-R10.
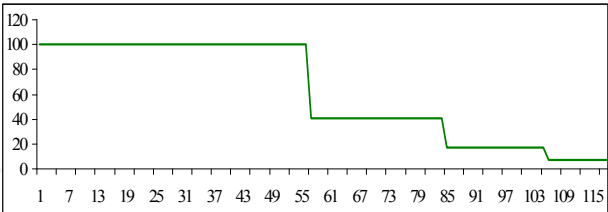

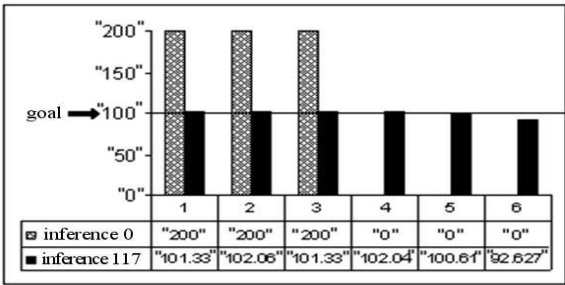Fig. 3. The evolution of balance degree bd

Fig. 4. The loads evolution for test A

After resetting the inference engine with the values previously obtained, we get a total of 18 inferences, whose result is presented in Table 1 (SC=Set of Conflicts, ER=Executed rule):

No.	μ_{x1}	μ_{x2}	μ_{x3}	μ_{x4}	μ_{x5}	μ_{x6}	x_{b1}	x^{b1}	SC_k	ER	Event	bd
0	101.33	102.01	101.33	102.00	100.61	92.627	0	00000000000	7	7	e_{43}	7.3575
1	101.33	102.01	101.67	101.67	100.61	92.627	7	00000010000	4	4	e_{35}	7.3575
2	101.33	102.01	101.14	101.67	101.14	92.627	4	00010010000	8	8	e_{56}	7.3575
3	101.33	102.01	101.14	101.67	96.883	96.883	8	00000011000	4	4	e_{35}	3.1018
4	101.33	102.01	99.011	101.67	99.011	96.883	4	00010010000	2	2	e_{13}	3.1018
5	100.17	102.01	100.17	101.67	99.011	96.883	2	01010010000	3	3	e_{21}	3.1018
6	101.09	101.09	100.17	101.67	99.011	96.883	3	00100010000	13	13		3.1018
7	101.09	101.09	100.17	101.67	99.011	96.883	3	00100010000	13,14	14	$[e_{56}]$	3.1018
8	101.09	101.09	100.17	101.67	97.947	97.947	8	00000011000	13	13		2.0378
9	101.09	101.09	100.17	101.67	97.947	97.947	8	00000011000	13,15	15	$[e_{35}]$	2.0378
10	101.09	101.09	99.058	101.67	99.058	97.947	4	00010010000	2	2	e_{13}	2.0378
11	100.07	101.09	100.07	101.67	99.058	97.947	2	01010010000	3	3	e_{21}	2.0378
12	100.58	100.58	100.07	101.67	99.058	97.947	3	00100010000	6	6	e_{42}	2.0378
13	100.58	101.12	100.07	101.12	99.058	97.947	6	00000100000	7	7	e_{43}	2.0378
14	100.58	101.12	100.60	100.60	99.058	97.947	7	00000010000	4	4	e_{35}	2.0378
15	100.58	101.12	99.829	100.60	99.829	97.947	4	00010010000	13	13		2.0378
16	100.58	101.12	99.829	100.60	99.829	97.947	4	00010010000	13,14	14	$[e_{56}]$	2.0378
17	100.58	101.12	99.829	100.60	98.888	98.888	8	00000011000		0	$e \neq e_{00}$	1.1394

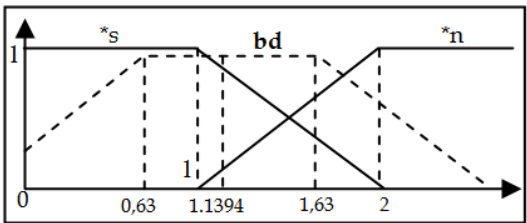Table 1. Inference process results for test A (case a2)

Fig. 5. The fuzzy final balance degree

We notice that, at the end of inference engine execution, in accordance with the reset, the balance degree is bd="1.1394", which corresponds to linguistic value *satisfactory* (fig. 5) because $\Pi(*s, "bd")=1$ and $N(*s, "bd")=0246>0$. The balance degree is satisfactory but the fuzzy differences between the final values of subsystems' loads and the expected average value are not similar (i.e. they are not all around zero). This is proved by the inference engine stop after the execution of meta-rule 14 (after which it cannot infer anything) and not of rule 11, as is conceived the control model. This is due to improper choice of the fuzzy set for the linguistic value *unsatisfactory*.

Test B. The purpose of this test is to highlight that, for other initial subsystems' loads, the control model achieves its intended objective, in terms of balance degree (which must be satisfactory at the end of the execution), and that the final loads are similar to the expected average value, given by $\mu_{\bar{x}} = (constfaz * \mu_{\bar{x}} (tp\ 32.83\ 33.83\ 1\ 1))$, according to the similarity criteria adopted within the KMSFL. In this case $\mu_{x_1} = "200"$, $\mu_{x_2} = ... = \mu_{x_6} = "0"$, $x_{b1}=0$, and the

other fuzzy values of the knowledge model are initialized and are similar to those in the test A. After the first execution, the inference engine performs 67 inferences and the values of fuzzy loads are: μ_{x_1} ="33.419", μ_{x_2} ="34.289", μ_{x_3} ="33.419", μ_{x_4} ="34.961", μ_{x_5} ="32.55", μ_{x_6} ="31.361". The last rule activated is meta-rule 13, and the balance degree at this stage is bd="1.9725". It is noted that the balance degree is satisfactory (Π(*s, "0.967") = 1 and N(*s, "0.967")=0.266>0), the fuzzy differences between the final loads and the expected average value are approximately zero, but the loads of the subsystems analyzed in pairs, according to the reasons within antecedent of rule 11, are not all similar to each other, because the execution does not end with rule 11. The observation above is based on the following calculations: for $\mu_{\bar{x}}$ =(32.83 33.83 1 1), μ_{x_1} =(32.919 33.919 1 1), μ_{x_2} =(33.789 34.789 1 1), μ_{x_3} =(32.87 33.87 1 1), μ_{x_4} =(33.69 34.69 1 1), μ_{x_5} = μ_{x_6} =(31.865 32.865 1 1) we get respectively the following pairs: $\Pi(\mu_{x_1},\mu_{\bar{x}})$ = 1, N($\mu_{x_1},\mu_{\bar{x}}$)=0.455, $\Pi(\mu_{x_2},\mu_{\bar{x}})$=1, N($\mu_{x_2},\mu_{\bar{x}}$)=0.02, $\Pi(\mu_{x_3},\mu_{\bar{x}})$=1, N($\mu_{x_3},\mu_{\bar{x}}$) =0.48, $\Pi(\mu_{x_4},\mu_{\bar{x}})$=1, N($\mu_{x_4},\mu_{\bar{x}}$)=0.07, $\Pi(\mu_{x_5},\mu_{\bar{x}})$=1, N($\mu_{x_5},\mu_{\bar{x}}$)=0.01 that satisfy the condition of similarity between any of the final states of the loads and the expected average value.

Test C. This test emphasizes the *dynamic circularity.* All subsystems are loaded with fuzzy values different from "0", and the characteristics of the associated fuzzy numbers are closer to the case of natural perception, i.e. have been increased the cores and the imprecision areas of fuzzy values, as follows: *s = *b = *mi = (-1 1 2 2), *n =* ma = (3 ∞ 3 3), "around x_0" = (x_0-1 x_0+1 x_0-3 x_0+3), ε = 1, η = 0, μ_{x_1} = (1222 1224 2 2) = "1223", μ_{x_2} = (309 311 2 2) = "310", μ_{x_3} = (444 446 2 2) = "445", μ_{x_5} = (37 39 2 2) = "38", μ_{x_6} = (741 743 2 2) = "742", x^{b1} = 0. We may notice the intensive use of meta-rules in the second part of the execution and the correct ending of the balancing problem, in terms of final state. The expected average fuzzy value is $\mu_{\bar{x}}$ = "610.8(3)". After the first execution of the inference engine of KMSFL, starting from inference 39, the system goes in a cyclical operation (dynamic circularity), since all inferences starting from inference 40 do not change the KMSFL state so that to achieve the intended objective correctly. This is justified by the value of bd = "3.7213", with Π(*n, "3.7213") = 1 and N = (*n, "3.7213") = 0.54426. In this example is not activated the control module from rule 12 consequent, whose purpose is to eliminate the cycles from inference engine operation. To solve this problem, which is not allowed in the operation of an expert system, in the next we will highlight the behavior of inference engine when activating the control module. Solving the above listed deficiencies is founded on resetting the inference engine operation, based on user inputs UI, in order to achieve the goal. Reset is done by positioning the vector x^{b1} = 0, the subsystems' loads and the other fuzzy values being equal to the last values previously used.
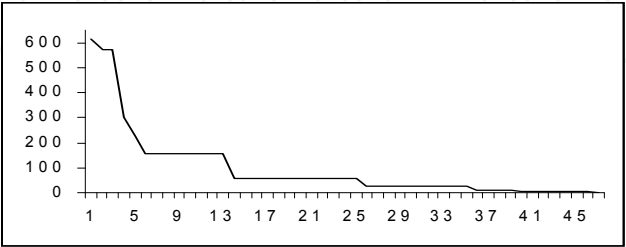

Fig. 6. Balance degree evolution for case C

After the reset, are used the meta-rules from the structure of knowledge management model to achieve the goal state. We get the following final fuzzy values of subsystems' loads: μ_{x_1} = (609.87 611.87 2 2), μ_{x_2} = (611.76 613.76 2 2), μ_{x_3} = (609.34 611.34 2 2), μ_{x_4} = (611.71 613.71 2 2), μ_{x_5} = μ_{x_6} = (608.16 610.16 2 2). The final balance degree is satisfactory because Π((-1 1 2 2),

(0.9311 2.9311 2 2)) = 1 and N((-1 1 2 2), (0.9311 2.9311 2 2)) = 0.017225 > 0. The evolution of the balance degree is shown in fig. 6. Final loads above are similar to each other because: $\Pi(\mu_{x_1}, \mu_{x_2})$ = 1, $N(\mu_{x_1}, \mu_{x_2})$ = 0.275 > 0, $\theta(\mu_{x_1}, \mu_{x_2})$ = 0.945, $K(\mu_{x_1}, \mu_{x_2})$ = 0.055, $\Pi(\mu_{x_1}, \mu_{x_3})$ =1, $N(\mu_{x_1}, \mu_{x_3})$ = 0.367 > 0, $\theta(\mu_{x_1}, \mu_{x_3})$ = 0.265, $K(\mu_{x_1}, \mu_{x_3})$ = 0.75, $\Pi(\mu_{x_1}, \mu_{x_4})$ = 1, $N(\mu_{x_1}, \mu_{x_4})$ = 0.04>0, $\theta(\mu_{x_1}, \mu_{x_4})$= 0.92, $K(\mu_{x_1}, \mu_{x_4})$ = 0.145, $\Pi(\mu_{x_1}, \mu_{x_5})$ = $\Pi(\mu_{x_1}, \mu_{x_6})$ = 1, $N(\mu_{x_1}, \mu_{x_5})$ = $N(\mu_{x_1}, \mu_{x_6})$ = 0.725 > 0, $\theta(\mu_{x_1}, \mu_{x_5})$ = $\theta(\mu_{x_1}, \mu_{x_6})$ = 0.855, $K(\mu_{x_1}, \mu_{x_5})$ = $K(\mu_{x_1}, \mu_{x_6})$ = 0.145.

It is noted that all criteria regarding the correct stop off the inference engine are met, thus obtaining the solution, whose value must be calculated through unification according to rule 11 antecedent. Thus, is revealed the inference engine behavior in order to avoid dynamic circularity presented in this example. The KMSFL can effectively predict a number of states that characterize the process, for a finite time horizon, as a result of its own inferences and of its knowledge management model.

7. Conclusions and future work

Designing and testing KMSFL system requires the existence of some scientific methods for knowledge acquisition, which unfortunately is a heterogeneous and time-consuming process. The synthesis of the control model for the case study presented, encompassing human control experience, was iterative, being necessary to indirectly develop some methods and testing and simulating environments for some crisp and fuzzy control models, permanently adjusting the inferential subsystem.

The importance of this paper consists of demonstrating the possibility of employing an expert KMS in problems of process control and planning, using imprecise knowledge. It was necessary to continuously adapt known models (e.g. theory of possibilities, discrete event systems) to synthesize a control structure based on fuzzy knowledge. We also tried to conceptually develop a multi-agent real control structure, which is a solution to meet a series of demands on the complexity of the control process. Such systems, especially those based on communication between agents by sharing memory, bring up features well suited for real-time applications, such as: integration of heterogeneous agents, interaction between activities of acquisition, reasoning and action on the external environment, fusion of data coming from sensors of different nature and operation, flexibility and efficiency in the integration of new data needed for reasoning, by simply writing them in the common memory. Real-time applications require the cooperation of elaborate reasoning processes. The essential point in integrating the cognitive/reactive aspects is the modeling of the relations between the process evolution and certain inferential methods, to allow the closed-loop system to achieve a range of required performances. Temporal aspects are an important dimension of any KMS embedded in NSTs. There is a distinction between real-time reasoning and reasoning over time, the latter being a feature of the knowledge-based systems. For this reason, it was absolutely necessary to explain the significance of time in the design and implementation of KMSFL, emphasizing its implications in the specific case of the developed application.

Following the tests presented and the knowledge representation and exploitation structures within KMSFL, we may outline the following conclusions:

i. the formalism chosen for knowledge representation is strong enough to support the representation of some types of knowledge underlying the synthesis of control decisions. It has the advantage of factorizing the knowledge, which substantially reduced the size of fuzzy rules base. Also, this knowledge representation method is more appropriate to express some types of knowledge similar to those commonly used in the decision synthesis through natural language;

ii.   the knowledge fuzzy model for the problem presented was developed incrementally, as was embedded in the model sufficient domain knowledge, resulting from the limitations observed in the crisp case. Is it possible to constantly adapt the fuzzy model through simulations;

iii. the inferential subsystem based on fuzzy logic solves the control situations correctly, both from the computational point of view and in terms of the semantics of the conclusions inferred through the chosen inference scheme;

iv.   modeling the process and the expert system as systems with logical events allowed the qualitative analysis of KMSFL;

v.   the control module integrated into the inferential subsystem automatically adapts the problem-solving process, being equivalent to closed-loop system input, denoted UI (user input). In this way, through the control meta-rule 12, we can simulate practically the activation of the relation between output Hypotheses/Conclusions and user inputs. This component is not activated in every case (initial loads). This justifies a major feature of the designed control expert system: its ability to correctly solve the problem, under the conditions specified for each case; vi) The case when $c_k \in \mathbf{N^+}$ is not a particular case for $\mu_{ck} \in \mathbf{R^+}$ due to certain restrictions on the events trajectories and to system's lack of flexibility in achieving the balance, thus leading to partial balance. For the discrete case, there is a tolerated unbalance value, denoted ψ and for which $|x_i - x_j| \leq \psi$, which defines the invariant set. For the fuzzy case, is achieved a good balancing solution, regardless of the initial subsystems' loads. In this case the system is broadly asymptotically stable relative to event trajectories.

We also summarize possible further development, which inherently can be obtained starting from KMSFL: identifying stronger planning characteristics, the logical specification of the real-time expert systems so that to describe how the statements change their truth values, depending on time or in order to meet some strong real-time restrictions, knowledge acquisition (as a difficult and insufficiently formalized problem), as well as the correct choice of the inference operators that lead to consistent results with regard to the well-defined knowledge semantics, integrating a control expert system like the one developed (agent) into a multi-agent system structure, identifying real problems in different application areas that can be solved using the expert system developed.

8. Acknowledgments

9. References

Alavi, M. (1999), Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues

Barachini, F. (1990), Match-Time Predictability in Real-Time Production Systems, Lecture Notes in Artificial Intelligence, nº 462, Expert Systems in Engineering, Principles and Applications, International Workshop, Viena, Austria, September, 1990, Proceedings, Springer-Verlag, p. 190-203, ISBN 3-540-53104-1

Binney, D. (2001), The knowledge management spectrum – understanding the KM landscape, Journal of Knowledge Management, 5, 1, 33-42

Bobrowitz , O. (1993), La représentation et l'utilisation des connaissances imprécises pour l'aide à la conduite des procédés, Thèse en Automatique et Informatique, Franche-Comté, 1991

Bylander, T. (1994), The computational complexity of propositional STRIPS planning, Artificial Intelligence, nº 69, 1994, p. 165-204

Chen, H.P.; Parng, T.M. (1996), A new approach of multi-stage fuzzy logic inference, Fuzzy Sets and Systems, Vol. 78, nº 1, feb. 26, 1996, p. 51-72

Davis, M. (2006), Semantic Wave 2006, Part-1: Executive Guide to Billion Dollar Markets, A Project10X Special Report

Dubois, D.; Lang, J.; Prade, H. (1991), Fuzzy sets in aproximate reasoning - Part 2: Logical approaches, Fuzzy Sets and Systems, nº 40, 1991, p. 203-244

Dubois, D.; Prade ,H. (1992), Fuzzy rules in knowledge-based systems, An introduction to fuzzy logic applications in intelligent systems (edited by R. Yager, L. A. Zadeh), Kluwer Academic Publishers, p. 45-68

Durán, O.; Aguilo, J. (2008), *Computer-aided machine-tool selection based on a Fuzzy-AHP approach*, Expert Systems with Applications, Volume 34, Issue 3, p. 1787-1794, 2008, www.elsevier.com

Ghallab, M. (1988), Compilation de bases de connaissances, PRC GRECO IA, Actes des Journées Nationales, p. 231-253

Iqbal, A.; He, N., Ullah Dar, N.; Li, L.- *Comparison of fuzzy expert system based strategies of offline and online estimation of flank wear in hard milling process*, Expert Systems with Applications, Volume 33, Issue 1, p. 61-66, 2007, www.elsevier.com

Jennings, N. R. and Wittig, T. (1992) ARCHON: Theory and Practice. In: Distributed Artificial Intelligence: Theory and Praxis. Kluwer Academic Press, pp.179-195

Kim, YG.; Yu, SH.; Lee, J.H. (2003), *Knowledge strategy planning: methodology and case*, Expert Systems with Applications, Volume 24, Issue 3, 1 April 2003, p. 295-307, 2003, www.elsevier.com

Kingston, J. (2001), *High Performance Knowledge Bases: four approaches to knowledge acquisition, representation and reasoning for workaround planning*, Expert Systems with Applications, Volume 21, Issue 4, p. 181-190, 2001, www.elsevier.com

Knight, J.F.; Passino, K.M. (1987), Decidability for temporal logic in control theory, Proceedings of the Twenty-Fifth Allerton Conference on Communication, Control and Computing, University of Illinois at Urbara-Champaign, p. 335-344

Lassaigne, R.; Rougemont, M. (1996), Logique et complexité, Hermes, Paris, 1996

Lau, H.C.W.; Cheng, E.N.M.; Lee, C.K.M.; Ho, G.T.S. (2008), *A fuzzy logic approach to forecast energy consumption change in a manufacturing system*, Expert Systems with Applications, Volume 34, Issue 3, April 2008, p. 1813-1824, 2008, www.elsevier.com

Lin, C.J.; Wu, W.W. (2008), *A causal analytical method for group decision-making under fuzzy environment*, Expert Systems with Applications, Volume 34, Issue 1, p. 205-213, 2008, www.elsevier.com

Luger, G.F.; Stubblefield, W.A. (1993), Artificial Intelligence, Structures and Strategies for Complex Problem Solving, Second Edition, The Benjamin/Cummings Publishing Company, Inc., ISBN 0-8053-4780-1

Lunardhi, A.D.; Passino, K.M. (1995), Verification of qualitative properties of rule-based expert systems, in Applied Artificial Intelligence, p. 587-621

Marco A.C. Pacheco, Marley M.B.R. Vellasco (2009), Intelligent Systems in Oil Field Development under Uncertainty, Studies in Computational Intelligence 183, Springer

Mazilescu, V. (2009a), A Real Time Control System based on a Fuzzy Compiled Knowledge Base, Proceedings of The 13th WSEAS International Conference on COMPUTERS, CSCC Multiconference, Rodos Island, Greece, July 23-25, Conference track: Artificial Intelligence. Computational Intelligence, p. 459-464

Mazilescu, V. (2009b), Knowledge Management & Intelligent Economic Systems, Europlus Publishing House, ISBN 978-973-1950-79-2

Mazilescu, V. (2010) Characterizing an Extended Fuzzy Logic System with Temporal Attributes for Real-Time Expert Systems, The 10th WSEAS International Conference on Systems Theory and Scientific Computation (ISTASC'10), Taipei, TAIWAN, p.159-165

Mazilescu, V. (2011), An Intelligent System for a Resource Allocation Problem based on Fuzzy Reasoning, Computer Technology and Application, David Publishing Company, From Knowledge to Wisdom, ISSN 1934-7332, p. 247-255

Müller, J.P. (1996), The design of intelligent agents: a layered approach, Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science), no 1177, ISBN 3-540-62003-6

Nebel, B.; Bäckström, C. (1994), On the computational complexity of temporal projection, planning and plan validation, Artificial Intelligence 66, p. 125-160

Omar, L.O. (2008), *Java Fuzzy Kit (JFK): A shell to build fuzzy inference systems according to the generalized principle of extension*, Expert Systems with Applications, Volume 34, Issue 1, p. 796-804, 2008, www.elsevier.com

Passino, K.M.; Antsaklis, P.J. (1989), A system and control theoretic perspective on artificial Intelligence planning systems, Int. Journal of Applied Artificial Intelligence, Vol. 3, p. 1-32

Passino, K.M.; Antsaklis, P.J. (1994), Lyapunov stability of a class of discrete event systems, IEEE Transaction on Automatic Control, Vol. 39, p. 269- 279

Ponnurangam, D.; Uma, G.V. (2005), *Fuzzy complexity assessment model for resource negotiation and allocation in agent-based software testing framework*, Expert Systems with Applications, Volume 29, Issue 1, p. 105-119, 2005, www.elsevier.com

Qian, D. (1992), Representation and use of imprecise temporal knowledge in dynamic systems, Fuzzy Sets and Systems, Vol. 50, no 1, Aug. 25, p. 59-77

Schwartz, D.G. (2006), Encyclopedia of Knowledge Management, Idea Group Inc., ISBN 1-59140-573-4 (hc)

Stankovic, J.A.; Ramamritham, K. (1995), What is Predictability for Real-Time Systems?, Advances in Real-Time Systems, IEEE Computer Society Press, p.26-33, ISBN 0-8186-3792-7

Tsui, E. (2002), The role of Artificial Intelligence in Knowledge Management, Knowledge-Based Systems, 13, 5, October, 235-239, URL= http://www.aifb.uni-karlsruhe.de/ ~sst/Research/ Publications/kbs-editorial-note2000.pdf

Wang, K.J.; Lin, Y.S. (2007), *Resource allocation by genetic algorithm with fuzzy inference*, Expert Systems with Applications, Volume 33, Issue 4, November 2007, p. 1025-1035, 2007, www.elsevier.com

Wenger, E. (2001), Supporting communities of practice: a survey of community-oriented technologies, 2001, URL=http://www.ewenger.com/tech/index.htm

Zadeh, L.A. (1978), Fuzzy sets as a basis for a theory of possibility, Fuzzy Sets and Systems, no1, 1978, p.3-28

Zadeh, L.A. (1983), Knowledge Representation in Fuzzy Logic, in An introduction to fuzzy logic applications in intelligent systems (edited by R. Yager, L. A. Zadeh), Kluwer Academic Publishers, p. 1-26

New Research on Knowledge Management Technology

Edited by Dr. Huei Tse Hou

Due to the development of mobile and Web 2.0 technology, knowledge transfer, storage and retrieval have become much more rapid. In recent years, there have been more and more new and interesting findings in the research field of knowledge management. This book aims to introduce readers to the recent research topics, it is titled "New Research on Knowledge Management Technology" and includes 13 chapters. In this book, new KM technologies and systems are proposed, the applications and potential of all KM technologies are explored and discussed. It is expected that this book provides relevant information about new research trends in comprehensive and novel knowledge management studies, and that it serves as an important resource for researchers, teachers and students, and for the development of practices in the knowledge management field.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Vasile Mazilescu (2012). A Knowledge Management System Embedded in the New Semantic Technologies, New Research on Knowledge Management Technology, Dr. Huei Tse Hou (Ed.), ISBN: 978-953-51-0074-4, InTech, Available from: http://www.intechopen.com/books/new-research-on-knowledge-management-technology/a-knowledge-management-system-embedded-in-the-new-semantic-technologies

INTECH
open science | open minds