Annals of "Dunarea de Jos" University of Galati
Fascicle I. Economics and Applied Informatics
Years XX – nº3/2014
ISSN-L 1584-0409 ISSN-Online 2344-441X
www.eia.feaa.ugal.ro

An Analisys for Intelligent Planning Applications

Vasile MAZILESCU*

ARTICLE INFO

Article history:
Accepted October 2014
Available online December 2014
JEL Classification
D80, D83

Keywords:
Planning, IKMSs, Logical Events.

ABSTRACT

We investigate some various planning complex applications. We can develop and justify thus a series of modeling and design techniques for intelligent management systems, as well as methods for the analysis of planning systems performance, and, of an expert system in particular, between which there are strong similarities. We will also outline a number of differences between conventional problem solving systems and Intelligent Knowledge Management Systems (IKMSs), the links between expert systems and those of structural and functional planning, the analogy between the model of the problem or business process and the field of the problem represented by a fuzzy knowledge system based on logical events. This way, the work reported in this paper serves to promote the development of some foundations on which to perform careful analysis for intelligent planning systems that operate in critical environments, like Virtual Organizations or Hierarchical Coalitions.

1. Introduction

A key issue for developing cognitive systems is the distinction between architected or human modeled ontology on the one hand, and emergent, largely machine – automated knowledge-base construction, on the other. Therefore, there are in IKMSS three large groups of problems that should approach in terms of decision-based applications: human-environment interface, qualitative knowledge modeling and time management. Such applications obviously require dated event operations the life-time of which should be managed by the system, which often works asynchronously with the acquisition and control system. A real-time expert system shell must also represent imprecise, time and temporal data, encode temporal knowledge, and manage temporal/fuzzy reasoning for allocating temporally interdependent tasks to homogeneous or heterogeneous cooperative agents in dynamic large-scale networks. We have developed a IKMSS as an agent that works as a planner and it is realized as a fuzzy real-time expert system shell to meet the challenges of the dynamic environment, like Virtual Organizations (VOs) or Hierarchical Coalitions (HCs). VOs provide an effective instrument to integrate a company's operations with those of other enterprises, to work with customers and create a better product or service, to achieve a faster time to market, and to acquire a higher degree of product customization (Burden 2009, Byrne 1993, Chen 2008, Norman et al 2004).

Section 2 refers to related works in the paper research problem. Section 3 is the comparative analysis of planning capabilities for new KMSs, i.e. Hierachical Coalition Systems and Distributed Control Systems. The planning process along hierarchies includes the steps of the individual plan. At higher levels or at the much more abstract levels, every step of the plan has more effects than a step of the plan at a lower level. Section 4 is dedicated for an analysis of our KMS based on logical events. Section 5 presents our conclusions and future developments for Distributed Knoweldge Management Systems (Vandaie 2008).

2. Related works

In the knowledge-based economy, the complexity in tasks and the demand for creative work from the workers are both very high (Becerra-Fernandez and Sabherwal 2004). These two trends have shifted management's focus from the traditional "proceed-based" framework to a "teams-based model" and the style of work from, historically, one of coordination and cooperation to collaboration. The advance of technology and the upsurge of the free workers have also exacerbated the need for collaboration. The wired, networked economy is the natural habitat of the free worker, and is opening up new markets for talent and new opportunities for networking outside company walls. Emerging agent-based systems offer new means of effectively addressing complex decision processes and enabling solutions to business requirements associated with VOs. Intelligent agents can provide more flexible intelligence/expertise and help the smooth integration of a variety of system types (i.e., Internet applications, Customer Relationship Management, Supplier Network Management, Enterprise Resources Management, Expert Systems). This section presents an overview of expert systems as the most widely-used approach for domain knowledge management today

*West University of Timisoara, Faculty of Economics and Business Administration, Romania. Email address: burca_valentin@yahoo.com (V. Burca)

as well as agent technology, and shows the latter as a superior systems development vehicle providing flexible intelligence/expertise and the integration of a variety of system types. Acquiring models means learning by observation, exploration and experiment, teaching and coaching, or reading. To use models means reasoning including mental simulations and tests, hypothesis, plausible inferences, logical thinking, and value-based trade-off.

Multiagent planning architecture (MPA) can be an option in the support systems for coalitions because it is a framework for the integration of various technologies into a system able to solve problems which cannot be solved by the independent systems. MPA is widely used in planning the applications, such as coalitions' operations. MPA agents are organized based on the concept of cells planning. Each cell has a managing member that performs the planning of the cell from a group of available agents and distributes the tasks among selected planning agents. Sharing the information generated during the planning process will be held by a central plan, and each agent of the cell can access it. MPA is important for the integration of the different planning solutions. This integration has limits for the development of support systems coalitions, like any other approach that tries to integrate existing instruments and possibilities in a simple framework. The main reason is collaboration between problem solving on different components and the need to be embedded from the beginning into systems. The problem is how to incorporate collaborative requirements into a distribution of processes planning, so that planning the final connection will not be a sum of independent plans (Stadtler and Kilger 2005). Considering that the most important function of the knowledge-based instruments is to support the human user, we must also understand how they interact in the planning the collaboration process. It should be noted that we do not take into account aspects of the interface which could improve the man-agent interaction. The main idea is that we can join up the groups, associated with the development of hierarchical coalitions as support for the agents, by means of a framework in line with an ontology based on constraints and on the appropriate functions. We can argument a MAP framework brings several advantages such as: a well-known environment to represent and build plans; a transparent manner to integrate collaborative concepts that complement the planning skills; opportunities to develop the human-agent mechanisms; support for customizing the intermediaries. Based on this statement, we can have specific problems.

Computer based information systems (IS) in manufacturing planning and control, have since their introduction evolved to cope with the needs of manufacturing firms. It is possible to identify a route of evolution starting with the introduction of manufacturing requirement planning (MRP) systems in the 1960s. Through the manufacturing resource planning (MRPII) systems in the 1980s and on through enterprise resource planning (ERP) systems in the 1990s ending in recent supply chain management (SCM) software such as Planning and Sceduling Systems (PSS) (APICS 2007, Bergamaschi et al 2006, Bhatt 2001, David et al 2006, Jonsson et al 2007). PSS are either add-ons or direct integral components of ERP systems which create the support mechanism for planning and decision making at the strategic, tactical and operational planning levels (Kreipl and Dickersbach, 2008). In general, PSS are said to create better plans than its predecessors as they include advanced functionality such as optimization and simulation. It is many times argued that PSS systems are especially well-suited for environments where simple planning methods cannot adequately address complex trade-offs between competing priorities (Jonsson et al 2007). Are proposed advanced algorithms in computer system to deal with complicated planning questions such as: What is the most adequate resource to use when more than one can be picked? What is the best assignment of product quantities to resource so that changeover can be minimal? What if some products could only be scheduled after others? What if the lines have different processing rates? What if some products cannot be scheduled simultaneously because they have the same tools, pallets, or fixtures? Investing a PSS is expensive and time consuming and it is important that the investment generates a profit. For a considerable time there has been much discussion on the value of IS support in planning and controlling material flows and production (Berglund and Karlun 2007, Chen 2008, Jonsson and Mattsson 2006, Kim el al 2003, Li and Xiao 2006, Lin el al 2007). The understanding of when and how to use PSS in order to obtain benefits, is limited. For practitioners, the potential value of knowing why, when and how to successfully use PSS and which benefits to expect is obvious. For academia in manufacturing planning and control, it is important to understand weather the PSS dominant approach really solves the real-world planning and scheduling problems considering the vast volumes of mathematical models and algorithms developed during the years. It is also interesting, for researcher in manufacturing planning and control, to understand how benefits are created when using PSS.

An expert system is a type of planner, since it emulates the way in which experts make decisions, in the presence of abstract, qualitative and approximate knowledge, in a limited expertise field. Currently, what is necessary is the formulation of a mathematical theory of intelligent planning systems, which operates in a dynamic environment, with real time properties. For systems of Artificial Intelligence applied to management, the fundamental concepts of control theory have a special significance. The *state* of a planner or an expert system describes the situation that defines the problem solving strategy at a given time. In closed loop planning systems, the planner detects the exists of the problem field and uses them in the decision making process. Closed loop planners undergo the execution, monitoring and re-planning, being able to continue their activities on the basis of contingency plans that may appear due to the disturbances which act on the problem field. Planning systems, or expert systems, may be designed with three hierarchical levels: the *execution* level, the *coordination* level, and the *control* level. The models used at superior levels are more abstract. A *planning problem* for a system may be defined thus: given an initial state, a desired state and an aggregate of possible actions, let it an aggregate of actions (either partially or totally ordered) be determined, which applied to the initial state leads the system to the desired state. Planning is a difficult problem and may become much simpler if certain restrictions are applied. The *problem of plan validation* is reduced to the verification, for a plan, of an initial state and of a given final state, that all actions mentioned in the plan may be executed successfully, meaning that all preconditions are fulfilled and the actions mentioned in the plan lead to the desired state. Planning is an argumentation task, with a view to determine a sequence of operators which allow the reaching of the goal, starting from a given initial state. Different planning problems may be defined by restricting the type of operators, imposing a series of limitations to the number of preconditions and post conditions (finite aggregates of elements which may be either variable or constant), by using certain logical operators within the operator structure. P For these types of restrictions, the planning problem is polynomial or NP-complete. Chapman proves, with the TWEAK system, that the planning problems is un-decidable, thus pointing out the difficulty of the planning problem in general, but doesn't prove what proprieties must be fulfilled for the reduction of complexity (Bylander 1994, Luger and Stubblefield 1993, Lunardhi and Passino 1995, McKay and Wiers 2003). Erol analyzes the planning proprieties of the TWEAK system, thus proving that the planning problem is un-decidable only if th e number of constants is infinite. In other words, if the aggregate of constants is finite, the operators are fixed and negated preconditions or post-conditions are not allowed, then the TWEAK system type planning problem is polynomial. This is a very specialized planning problem and it isn't clear how a number of these conditions may be relaxed. There are other results regarding the decidability of some types of specialized planners, but they provide little information regarding the overall complexity of the planning problem. The computational complexity of planning was investigated. Bylander analyzes the general problem of deciding the existence of a solution for the planning task in the context of the STRIPS system and it is demonstrated that this general problem is PSPACE-complete (Bylander 1994). A certain type of temporal reasoning is represented by the evaluation of consequences of an aggregate of events. This problem is also linked to the other aspects of the temporal arguments, such as the planning and validation of a plan. Seeing as the planning is done incrementally, of interest is not only the validation of the plan but also the determining of the causes due to which the planning is not yet valid. When we compare the characteristics of a conventional system as opposed to an Artificial intelligence system, the following observations may be made: **i)** *the decision rate* with conventional systems is typically slower than in Artificial Intelligence systems; **ii)** *the abstract and generalization character* of models used in Artificial Intelligence systems is larger when compared to the fine grain of models used in conventional systems; **iii)** *the synthesis level* of decisions and learning proprieties, similar to those of human beings, exist in Artificial Intelligence systems at a much higher degree than in the conventional ones. The result is a high degree of autonomy which exists within non-conventional systems (Nebel and Bäckström 1994).

3. A Comparative Analysis of Planning Capabilities for New KMSs

Planning systems based on Artificial Intelligence use *specific* models for the problem field, called *problem* representations and logic argumentation models. Current expert systems have many characteristics in common with planning systems (representation of knowledge, heuristic inference strategies), conceived specifically for communicating with the exterior, while conventional expert systems are strongly encapsulated. Planning systems execute actions in a dynamic manner to produce modification in the state of the problem field. The planner monitors the problem field to progressively obtain information useful for decision synthesis. An explicit loop is being crossed between the actions done by the planner, the problem field, the measured outputs and the planner that uses the outputs to decide the control actions, with a view of reaching the goal. Within expert systems, a similar loop exists: the knowledge base represents the problem field, while the inference engine represents the planner.

Expert systems inherently have a *purpose*: diagnosis, system configuration. Certain expert systems have more planning elements than others. A planning system based on models specific to Artifical Intelligence is made of a planner, the problem field, the connections between the two and the exogenous outputs. The planner's outputs are inputs for the problem field, representing control actions. The outputs of the problem field are inputs for the planner. They are measured by a planner and used to determine evolution in the problem solving process. Furthermore, there are non-measurable exogenous inputs for the problem field (disturbances), which represent the uncertainty associated with it. The measured exogenous input of the planner represents the goal. It is a task of the planner which examines the outputs of the problem field, it compares them to the goal function and determines what actions must be undertaken to reach it. Not all planners are completely autonomous. Some have a user interface, through which the goals may be generated, allowing for certain degrees of intervention of the human element in the planning process. The planner acts on the problem field through inputs, with a goal to solve a specific problem (Stoop and Wiers 1996, Vollmann et al 2005).

The *solution for the problem* is a sequence of inputs and outputs (possible states), generated with a view to fulfill the purpose. A model for the field of the real problem is thus developed, called the representation of the problem. The problem field is, in a certain sense, infinite, since one can never model all of the aspects that are specific to it. The representation of the problem becomes, thus, incomplete, through the simplifying of certain characteristics, either known or not. However, simpler models are desired, since there exists an inverse proportionality relation between the complexity of the modeling and the strength of the analysis. Planners based on Artificial Intelligence techniques are made from the following important components: the plan generator, the plan simulator (uses plans in the shape of heuristic decision rules), the execution model for the selected plan, the situation evaluator (optional). The Artificial Intelligence techniques used in the design of planning or expert systems are diverse. They refer especially to representation problems (on must take care in selecting the degree of detail for the mathematical structure used or the allowed modeling power, since too much modeling power may prevent the development of certain components of the planner, the verification and validation of the system), the type of approach (dependant or independent of the field), the type of the planner (hierarchical or not, linear or non-linear, reactive, distributed, encompassing a meta-planner etc.), the type of interactions that may appear in the synthesis of decisions, the forms of searching and re-planning.

The fundamental characteristics of a planning type or expert type Artificial Intelligence system, such as the one designed in the current paper, are (Lunardhi and Passino 1995): a) It can be considered a planning model of the acting human expert; b) The State of the planner represents the situation in which the problem solving strategy exists at a given time, being made up of the information necessary for predicting the future behavior of the system. c) The Controlability represents the ability of inputs to modify the state of the system, considering for analysis a deterministic system. The problem field is completely controllable at a given time i, if there exists a finite time j, with j>i, so as for any x_i and any state x, there is a finite input sequence $u_i,..., u_j$ (produce which the planner can produce) which transfers state x_i to state x at time j, identified through $x=x_j$. In case the problem field is completely controllable, then for any state there exists a planner which designs any specific state, called purpose; d) The observability of the problem field is the ability of determining its state, using inputs (sequence of actions), outputs and the asociated model; e) Minimality reflects how well the system was modeled; f) Internal stability reflects the ability to remain in an aggregate of states (invariant) when there are no goal-inputs and in the presence of disturbances.

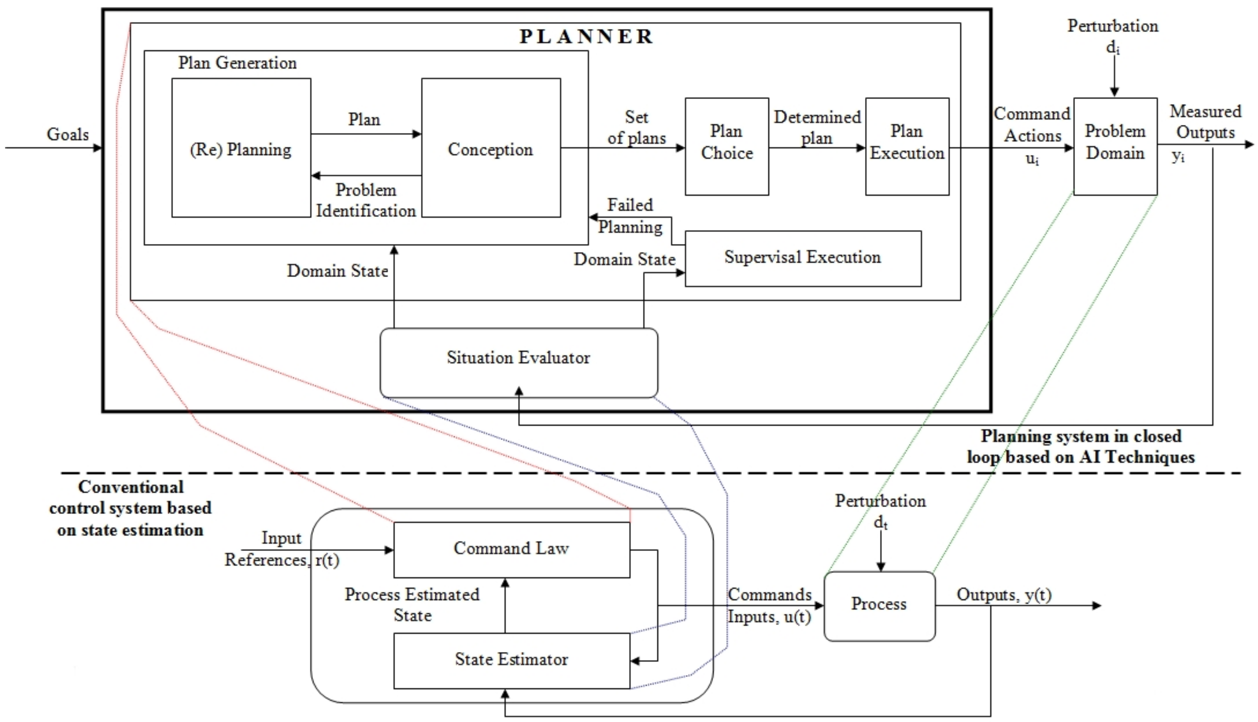

Figure 1. *Closed loop planning system*

We will now present the important problems that can be outlined for open loop systems. If the problem field is uncontrollable, then there is no planner capable to solve the problem, and if it is controllable, the existence of such a planner is assured. An important problem with regards to designing an expert system of management for a business process is represented by the analogy between the problem field and the knowledge model. The problem field represents the aggregate of real problems which the Artificial Intelligence systems must solve, using a series of arguments, modeled as formal systems (automated or quantified logic, temporal, dynamic, multi-valued, nuanced, etc.). Inputs and outputs are marked by u(t), y(t) for conventional control systems and u_i, y_i, for Artificial Intelligence systems. Obviously, inputs (outputs) in the problem field are more abstract and will be described by symbols. These symbols may signify any type of action that a planner may use over the problem field. With regards to the input u(t) of the process, input u_i of the problem field is a sequence of symbols. The physical system, both for the problem field and for the process, is represented by a part of the real that we wish to study. The difference between them is in the type of systems and, thus, of the modeling techniques used. Of note is the fact that the problem field cannot always be described by equations, therefore conventional control techniques are not adequate for Artificial Intelligence systems (Barachini 1990).

Due to the strong similarities between the process and the problem field, an analogy may be outlined between the models used for the process and the problem field, on the one side, and between the fundamental systemic concepts on the other. The process is described through stochastic (possibly non-linear) or differential equations, called state and output equations, which describe the dynamics of the process, the structure and its inter-connections. The problem field may be described through symbolic equations. There are still strong mathematical analogies between the two types of systems, studying certain proprieties of special significance in the classic control theory: controllability, observability, stability, system rate. There is a structural analogy between classic control systems and Artificial Intelligence systems (planning, expert system), both in an open loop as well as a closed loop. *Generating a plan* is the process of synthesis of an aggregate of candidate plans, to fulfill the g_i purpose at moment i (the goals can remain fixed or be modified). During the process of plan generation, the system *designs* (through a simulation based on a model of the problem field) the plausible future plan.

The system uses heuristic plans based on decision rules, resource management, success probabilities, in order to choose the plan to execute. *The plan execution module*, translates the chosen plan into actions over the problem field, using techniques of resource allocation. S*ituation evaluator* uses the inputs, outputs and the representation of the problem in order to determine the state of the field. The estimate state of the field is used for updated the model implicated in the design and generation process of the plan. The term of *situation* is preferred in this case to designate an abstract, general view of the state. *The monitoring of the execution* uses the estimated state of the problem field, in order to determine the viable nature of the elaborated plan, in cases of failure launching re-planning actions. In the structure of a planner based on techniques of Artificial Intelligence may also be integrated a *world modeler*, which does the permanent updating of the representation of the problem field.

The evaluation of the situation and the monitoring of the execution in open loop systems cannot be assured, being no possibility of re-planning. In this case, the planner is sensible to the variations which appear in the problem field, open-loop planners not being able to reduce this sensibility. These planners are of interest only for insignificant disturbances, which is not the case with real problems, which need complex and detailed problems. Seeing as the outputs of the system are not detected with inputs, if the problem field is unstable (input-output or internal), then it will never be able to be stabilized in open-loop planning. Open-loop planning absolutely imposes an exact amount of knowledge of the problem field. Since this cannot be obtained, any disturbance may be catastrophic. Open-loop planners present advantages due to their simplicity. If the problem field is stable and the disturbances are insignificant, then these planners may be useful. They also have the advantage of reduced costs, not being necessary measurements of states and outputs related to the problem field. Closed loop planning systems (figure 1) are analogous to closed loop conventional control systems and most of the time, do not use situation evaluation. The monitoring of the execution as well as the re-planning are thus permitted.

The problem field may not be completely controllable, either because of the elements of executions which seriously affect the state, either because of the reconsidering of the model corresponding to the problem field, seeing as controllability is a propriety of the mathematical used model. Controllability and observability studies can be useful as a guide for the synthesis of the model for the problem field. The evaluations of the situation is not necessary for planning if the whole state of the problem field is measurable, this propriety being analogous to the total sate reaction. In case the study of the observability outlines certain non-observable states of the problem field, additional translators can be implements which provide the necessary information about the state. Thus, the existence of a compromise between the costs of implementing a planning system and its complexity, becomes apparent. The solution of implicating several translators for the evaluation of the entire state may be more expensive. In this case, the existence of a situation evaluator is no longer necessary, which leads to a simpler planner structure. If the problem field is completely controllable and observable with regards to inputs and outputs chosen, then a closed loop planner exists which stabilizes the problem field, should it be unstable. The planning system examines the difference between the current output situation and the desired goal, with a view to executing certain actions. The error in Artificial Intelligence Planning Systems, or in expert systems, is not so easily obtained, compared to the classics systems of control, seeing as the distance between the symbols is much harder to quantify. Ever more difficult is the evaluation of the similarities between states, based on a given criteria in the case of expert systems of monitoring based on imprecise knowledge. These aspects will clearly come into the design of the IKMSS. Planning systems can have, at any given moment in time, the stop function constantly (regulator type planning systems) or not (goal-following planning systems). In case a closed loop system or an expert monitoring system is being designed, certain proprieties are outlined through a qualitative analysis, similar to the one present in chapter five and in the work. Stability is a propriety of the closed loop system, also valid for knowledge-based planning systems. Even if the closed loop planning system is thought out with the ability to stabilize any system that can be stabilized, it can destabilize the specific field of a different problem. Such is the case, for example, in the application of a correct plan with a certain delay. Systems are destabilized when the rise of the system rate over a certain safety value, this being a specification of the closed loop system, is attempted. A major advantage of closed loop planning systems as compared to open loop systems is their ability to reject disturbances which may appear in the problem field, being insensitive to variations of the problem field (reaching a goal, even if the model corresponding to the problem field incorporates objectively approximate knowledge). In classic control systems theory, this objective is accomplished using robust systems. Closed loop planning systems with situation evaluation are analogous to conventional controllers with state estimation (Figure 1).

If the problem field is observable, then a situation evaluator exists which determines the state of the problem field using inputs, outputs, and the model of the problem field. On case this condition is not satisfied, the situation evaluator cannot, at any moment in time, accomplish its objective. Situation evaluation is particularly useful in stabilizing the problem field, when it is detectable. The state estimator (observer) is analogous to the situation evaluator. There are, also, similar analogies between classic adaptive control systems and adaptive planning systems, based on Artificial Intelligence techniques. Designing a planning systems is a difficult problem, through the variety of the models used (satisfactory from a computational perspective), which must be a reflection of the complexity of the environment in which they operate.

4. An analysis of our KMS based on logical events

Our management expert system (Mazilescu 2009, Mazilescu 2010, Mazilescu 2011, Mazilescu 2012), modelled as a discrete event system using the relationship CES = (X^{CES}, E^{CES}, f_e^{CES}, Γ_e^{CES}, g^{CES}, x_0^{CES}, E_v^{CES}), is extended with the cost function ζ_e: $X^{CES} \times X^{CES} \rightarrow R^+$, attached to the logical event that occurs at moment $k \in T \subseteq \mathbf{R}^+$, and with the non-empty set of final states $X_f \subset X^{CES}$. The management expert system model thus obtained, is a formal description of deterministic discrete event systems, meaning that for a given input event there is exactly one subsequently state. State transitions can occur deterministically in relation to time, so this model can highlight a certain type of asynchronism. The set:

$$E(CES) = \{(x, x') \in X^{CES} \times X^{CES} \mid x' = f_0^{CES}(x), e \in P(E_u \cup E_d^{CES}) - \{\varnothing\}\} \quad (1)$$

denotes the set of events (possibly infinite) for the management expert system, events obtained as a result of logical inferences, hereinafter logical events. Cost function $\zeta_e(x, x')$ of a logical event is defined for any $(x, x') \in E(CES)$, and indicates the cost of each event that occurs at the process input (e.g. fuzzy expert system's output based on logical inference), which allows the state transition of closed-loop system. This function has the property that there must be a $\delta' > 0$ so that $\zeta_e(x, x') \geq \delta'$ for any $(x, x') \in E(CES)$, where δ' is calculated on the number of inferences performed by the fuzzy expert system incorporated into the management expert system structure.

The fuzzy expert system is described by SEF = (X^{SE}, E^{SE}, f_e^{SE}, Γ_e^{SE}, g^{SE}, x_0^{SE}, E_v^{SE}), where $E^{SE} = E_i^{SE} \cup R \cup E_o^{SE}$ is the set of events of the expert system, for which takes place inclusion $E_i^{SE} \subset P(E_p^{SE} \cup E_r^{SE} \cup IU) - \{\varnothing\}$. E_i^{SE} is the set of all input events of the expert system consisting of the reference inputs E_r^{SE}, the process output events E_p^{SE} that may occur and in relation to which the system must be able to respond in real time, and the user inputs, which are logical events. Output events of the process, E_p^{SE}, can be, in certain situations, the changes in process' state. Function f_e^{SE} is the transition function of the expert system, and x_0^{SE} is its initial state. Closed-loop system is characterized by the state $(x^{SE}, x) \in (X^{SE} \times X)$. If (x^{SE}, x) is a valid state of the closed-loop system, then the next new state is given by $((x')^{SE}, x')$ with $(x')^{SE} = f_e^{SE}(x^{SE})$ and $x' = f_e(x)$.

We note by X^{CES*} the set of all finite strings over X^{CES} which also include the null element. A string $s \in X^{CES*}$ is called *state trajectory* of the management expert system, if for any successive states $xx' \in s$, $x' = f_e^{SE}(x)$ for a given state of the expert system $x^{SE} \in X^{SE}$ (xx' is the chaining of state x with state x'). Let $E_s(CES) \subset$

$E(CES)$ the set of all logical events necessary to define a particular state trajectory $s \in X^{CES*}$. Set $E_s(CES)$ can be determined by generating pairs (x, x'), (x', x''), ... An input sequence for the management expert system that produces a state trajectory $s \in X^{CES*}$, is built so that $x' = f_e^{SE}(x)$ for any $xx' \in s$. Let $X_z \subset X^{CES}$ and $T(CES, x, X_z) \subset X^{CES*}$ denotes the set of all finite state trajectories $s = xx' \ldots x''$ of the management expert system starting with $x \in X^{CES}$ and ending in $x'' \in X_z$, or $T(CES, x_0, X_f)$ is the set of all finite state trajectories starting in initial state x_0 and ending in the final state $x \in X_f$.

A valid behaviour of a logical discrete events system modelled with the relationship $CES = (X^{CES}, E^{CES}, f_e^{CES}, \zeta_e^{CES}, g^{CES}, x_0^{CES}, F_v^{CES})$ can be characterized by the set of all valid state trajectories $T(P, x, X_f)$, $x \in X^{CES}$ with respect to the sequence of input events (logical). Input events of the process, in terms of inferential processes results, are either the result of fuzzy expert system's action $E_o^{SE} \subset P(E_u \cup IC) - \{\emptyset\}$ as the output events of the expert system, or hypotheses / conclusions (IC) obtained from the inference process, which are addressed to human decision maker. It is clear that the outputs of the system based on imprecise knowledge are calculated according to a dependence such as $\delta^{SE}: X^b \times X^{int} \rightarrow E_o^{SE}$, in relation to the current state x^{SE}.

Let $Q = E_u \cup E_d^{CES}$ a finite set of input events of the management expert system, which also includes logical events such as those listed above. The valid behaviour of the closed-loop system is characterized by the set of valid state trajectories, i.e. the set of physically realizable trajectories. Let $CES = (X, Q, f_e, \zeta_e, x_0, X_f)$ the *reduced representation* of the valid behaviour of closed-loop system and $A = (X_a, Q_a, f_a, \zeta_a, x_{a0}, X_{af})$ the model of another discrete events system that specifies the *allowed behaviour* of the same system, which must also be valid.

Formally, the allowed behaviour A is contained in the allowed behaviour of CES and is noted with A[CES], and are met the following conditions in relation to A: **i)** $X_a \subset X$; **ii)** $Q_a \subset Q$; **iii)** $f_a: X_a \rightarrow X_a$, at the occurrence of a logical event is given by $f_a(x) = f(x)$ if $f(x) \in X_a$, else is undefined; **iv)** $\zeta_a: X_a \times X_a \rightarrow R^+$ is a restriction of $\zeta_e: X \times X \rightarrow R^+$; **v)** $x_{a0} = x_0$; **vi)** $X_{af} \subset X_f$. Furthermore, $E(A) \subset E(CES)$ denotes the set of allowed logical events of A. The model of A is specified by designer and is the desired behaviour of the management system that includes a system based on imprecise knowledge, contained in the valid behaviour of the management system.

As the default graph of A is locally finite, there is a finite number of finite length state trajectories and generally there may exist several state trajectories for which the minimum is reached. Let $X_z \subset X_a$. The set of minimum cost state trajectories of A, starting in state $x \in X_a$ and ending in state $x \in X_z$, is denoted by $T^*(A, x, X_z) \subset T(A, x, X_z)$. We are interested in the set of optimal allowed state trajectories starting in x_0 and ending in X_{af}, denoted $T^*(A, x_0, X_{af})$. We define the performance index $J: X_a^* \rightarrow R^-$ which, in terms of event cost, becomes

$$J(s) = \sum_{(x, x') \in E_s(A)} \zeta_e(x, x')$$

for any $s \in T(A, x, X_a)$, $x \in X_a$ and $e \in P(E_u \cup E_d^{CES}) - \{\emptyset\}$. By definition, $J(s) = 0$ if $s = x$. The goal is to find a sequence of input events to produce a state trajectory s which minimizes $J(s)$ and ends in X_{af}.

Assume that A is (x_0, X_{af})-admissible, the problem of synthesis of a management expert system is to determine the state trajectory $s^* \in T(A, x_0, X_{af})$, so that $J(s^*) = \min\{J(s): s \in T(A, x_0, X_{af})\}$, using a system based on imprecise knowledge and implicitly an appropriate inference algorithm. The approaches specific to symbolic AI systems use heuristic search techniques, which successively generate candidate state trajectories until it can be determined an optimum, in order to reduce computational complexity (Lunardhi and Passino 1995). Heuristic information is called *evaluation function*, in whose structure there is a heuristic function. Defining the evaluation function depends on the performance index and on the information available for the search algorithm (of A* type) at different calculation levels of the evaluation function.

Assume a state trajectory $s = s_x s_{x>}$, with $s \in T(A, x_0, X_{af})$ for which $J(s) = J(s_x) + J(s_{x>})$.

If $s^* \in T^*(A, x_0, X_{af})$ and $s^* = s_x^* s_{xx'}^*$, then $J(s^*) = J(s_x^*) + J(s_{xx'}^*)$, where:

$J(s_x^*) = \min\{J(s_x): s_x \in T(A, x_0, x)\}$ and $J(s_{xx'}^*) = \min\{J(s_{xx'}): s_{xx'} \in T(A, x, X_{af})\}$ (2)

If $x_b \in X_a$ is a particular state that is not on any optimal state trajectory and $x_b \in s$ for $s \in T(A, x_0, X_{af})$, then $J(s) > J(s^*)$ for $s^* \in T(A, x_0, X_{af})$. This shows that $J(s)$ satisfies the optimality principle of dynamic programming. $J(s)$ is a perfect discriminant because it provides accurate information on the situations in which states are on an optimal trajectory or not. Using $J(s)$ together with an A* algorithm is a search strategy that can generate optimal state trajectories. The present problem depends categorically on the necessary information which is not available for algorithm A*, in order to calculate the index $J(s^*) = J(s_x^*) + J(s_{xx'}^*)$. For algorithm A* index $J(s^*)$ is estimated with an evaluation function $f: X_a^* \rightarrow R^+$,

defined for any $s \in X_a^*$, so that $s \in T(A, x, X_{af})$, where $x \in X_a$. Evaluation function f is obtained by approximating $J(s_x^*)$ and J([illegible]) with appropriately defined functions. Let $s = s_x s_{x>}$ a given state trajectory. The value of $J(s_x^*)$ will be estimated using the function g: $X_a^* \rightarrow R^+$, where $g(s_x) = J(s_x)$ for any and $s_x \in T(A, x_0, X_a)$. Notice that $g(s_x) = 0$ if $s_x = x_0$.
Function $g(s_x)$ is used to highlight the expansion cost for reaching the current state x, and is used for the entire state trajectory from x_0 to x, to estimate its cost. It is not necessary that $g(s_x) = J(s_x^*)$, with $s_x^* \in T(A, x_0, x)$. $J(s_x^*)$ estimates the cost until the current state x. In order to estimate J([illegible]), we use function h: $X_a^* \rightarrow R^+$, with $h(x) = 0$ if $x \in X_{af}$.

For h, the only information available for determining an estimation of J([illegible]) is the current state and the final set X_{af}. The function h is called heuristic function, since it offers the possibility to apply the algorithm A* using specific information relative to a particular search problem and allows focusing the search with an algorithm of A* type. The appropriate choice of functions for the structure of algorithm A* provides an effective search, i.e. a good solution for the synthesis of management expert system. Evaluation function is chosen so that $f(s_x) = g(s_x) + h(x)$, where $x \in X_a$ is the current state of the management expert system for A*. Correct interpretation of $f(s_x)$ is that it estimates the cost of a state trajectory from x_0 to $x' \in X_{af}$, passing through the state x. It does not estimate exactly the cost of s_x.

The algorithm A* operates by generating candidate state trajectories characterized by two sets of events (equivalent to pairs of states according to relation 1): $PC \subset E(A)$ and $QO \subset E(A)$, where PC is the set of events that are not candidates for development, and QO is the set of events likely to be developed. The content of sets PC and QO changes during the execution of A*, but cannot occur when $(x^1, x^2) \in PC \cup QO$ and $(x^3, x^4) \in PC \cup QO$, so that $x^2 = x^4$ and $x^1 \neq x^3$.

Let the set of state trajectories of A, investigated by A*, denoted by T(A, PC, QO). Each part of the state trajectory $s_{x'} \in T(A, PC, QO)$ starts with the initial state x_0 and has as final state $x' \in X_a$, so that $(\cdot, x') \in PC \cup QO$. For $s, s' \in X_a^*$ let $s \leftarrow ss'$ be the replacement of state s with ss'. To determine $s_{x'} \in T(A, PC, QO)$ using sets PC and QO, we choose $(x, x') \in PC \cup QO$ and note $s = xx'$.

The set $K(x) = \{x' \mid x' \in X_a \text{ and } x' = f_a(x)\}$ defines the expansion of state $x \in X_a$. A* produces an allowed optimal state trajectory $s^* \in T^*(A, x_0, X_{af})$, assuming that A[P] is (x_0, X_{af})-admissible, according to the following algorithm:

Algorithm. *(search algorithm for the state space)*

i) Let PC = {} and QO = **{(x**$_0$**)}**;

ii) *If* |QO| > 0 *then* **iii),** else *stop* (the algorithm has no solution);

iii) Choose $(x, x') \in QO$ so that $f(s_x x')$ is minimum. Let $QO \leftarrow QO - \{(x, x')\}$ and $PC \leftarrow PC \cup \{(x, x')\}$;

iv) *If* $x' \in X_{af}$ *then stop*, with $s_{x'} \in T^*(A, x_0, X_{af})$, an optimal state trajectory;
else for each state $x'' \in K(x')$:

1) *if* for $(\forall)\ \bar{x} \in X_a$, $(\bar{x}, x'') \notin PC \cup QO$ *then* $QO \leftarrow QO \cup \{(x', x'')\}$;

2) *if* $(\exists)\ \bar{x} \in X_a$, so that $(\bar{x}, x'') \in QO$ and $f(s_{x'} x'') < f(f_{\bar{x}} x'')$ *then* $QO \leftarrow QO - \{(\bar{x}, x'')\}$ and $PC \leftarrow PC \cup \{(x', x'')\}$;

3) *if* $(\exists)\ \bar{x} \in X_a$, so that $(\bar{x}, x'') \in PC$ and $f(s_{x'} x'') < f(f_{\bar{x}} x'')$ *then* $PC \leftarrow PC - \{(\bar{x}, x'')\}$ and $QO \leftarrow QO \cup \{(x', x'')\}$.

Under these conditions, for all $x \in X_a$, must be specified a function h(x), from the structure of function $f(s_x)$, to satisfy the conditions (Bylander 1994): **i)** *admissibility,* $0 \leq h(x) \leq$ J([illegible]), for any $x \in X_a$, so that [illegible] $\in T^*(A, x, X_{af})$; **ii)** *consistency* (or *monotony),* $h(x) \leq \zeta_e(x, x') + h(x')$, $(\forall)\ (x, x') \in E(CES)$.

5. Conclusions

The use of temporal aspects refers to the design of those tools to integrate time in control economic applications (e-commerce, Knowledge Management Systems, Virtual Organizations, Multi-agent Systems). These aspects are formally found on the inference engine algorithms, able to make full use of the specific knowledge to the process control. We assume that the process operates like finite nondeterministic state machine, while the expert system will operate like a finite deterministic state machine. The problem domain must be defined as a collection of problems that the expert system desires to solve. In conventional control, the plant is a dynamical system, described with linear or non-linear differential/ difference equations. A IKMSS based on knowledge control and planning models, consists of the planner or the inference engine, the problem domain, the exogenous inputs, and their interconnections. An expert system can be modelled using predicate or temporal logic or other symbolic techniques such as finite state machine. Although the KMSs are frequently being used to perform complex control functions, most often it is the case that no formal analysis of the dynamics is conducted because mathematical analysis of such systems is often considered to be beyond the scope of conventional control theory. The research work reported in this paper serves to promote the development of a firm mathematical foundation on which to perform careful analysis for the verification and validation of KMBSs. There are important another future directions for this work, investigating the dynamics of other reasoning systems that utilize learning and planning in various complex applications, studying computational complexity issues relative to conflict resolution strategies and meta-knowledge representation, and modelling realistic industrial or economic applications that involve Distributed Knowledge Management Systems.

References

1. *Alavi, M. (1999). Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues. Retrived October 26, 2012, from http://knowledge.emory.edu/papers/1005.pdf.*
2. *APICS. (2007). Using Information Technology to Enable Supply Chain Management, APICS Certifies Supply Chain Professional Learning System, APICS, Alexandria, VA USA.*
3. *Barachini, F. (1990). Match-Time Predictability in Real-Time Production Systems in Springer-Verlag, Lecture Notes in Artificial Intelligence (pp. 190-203). Retrived May 8, 2012, from http://link.springer.com/chapter/10.1007%2F3-540-53104-1_42*
4. *Becerra-Fernandez, I., and Sabherwal, R. (2001). Organizational knowledge management: A contingency perspective. Journal of Management Information Systems, Armonk, Summer, Vol. 18 No. 1, 23-58.*
5. *Bergamaschi, S. G. Gelati, F. Guerra, M. Vincini. (2006). An intelligent data integration approach for collaborative project management in virtual enterprises, World Wide Web 9 (2006) 35–61. doi 10.1007/s11280-005-2322-7*
6. *Berglund, M. and Karlun, J. (2007). Human, technological and organizational aspects influencing the production scheduling process, International Journal of Production Economics, Vol. 110, No. 1/2, 160-174.*
7. *Bhatt, G. (2000). Organizing knowledge in the knowledge development Cycle, Journal of Knowledge Management, Vol. 4 No. 1, 15–26.*
8. *Bhatt, G. (2001). Knowledge management in organizations: examining the interaction between technologies, techniques, and people, Journal of Knowledge Management. Kempston: Vol. 5 No. 1, 68-75.*
9. *Bylander, T. (1994). The computational complexity of propositional STRIPS planning, Artificial Intelligence, nº 69, 165-204.*
10. *Burden, D.J.H. (2009). Deploying embodied AI into virtual worlds, Knowledge-Based Systems 22, 540–544.*
11. *Byrne, J. (1993) The Virtual Corporation, Business Week, Feb. 8, 98-104.*
12. *Chen, T.-Y. (2008). Knowledge sharing in virtual enterprises via an ontology-based access control approach, Computers in Industry 59, 502–519.*
13. *David, F., Pierreval, H., and Caux, C. (2006). Advanced planning and scheduling systems in the aluminium conversion industry, International Journal of Computer Integrated Manufacturing, Vol. 19, No. 7, 705-715.*
14. *Jonsson, P., Kjellsdotter, L. and Rudberg, M. (2007). Applying advanced planning systems for supply chain planning: three case studies, International Journal of Physical Distribution & Logistics Management, Vol. 37, No. 19, 816-834.*
15. *Jonsson, P. and Mattsson, S-A. (2006). A longitudinal study of material planning applications in manufacturing companies, International Journal of Operations and Production Management, Vol. 26, No. 9, 971-995.*
16. *Kim, Y. G., Yu, S. H. Lee, J. H. (2003). Knowledge strategy planning: methodology and case, Expert Systems with Applications, Volume 24, Issue 3, 1 April 2003, 295-307.*
17. *Li, H., R., Xiao, (2006). A multi-agent virtual enterprise model and its simulation with Swarm, International Journal of Production Research 44, 1719–1737.*
18. *Lin, C-H., Hwang, S-L. and Wang, M-Y. (2007). A reappraisal on advanced planning and scheduling systems, Industrial Management & Data Systems, Vol. 107, No. 8, 1212-1226.*
19. *Luger, G.F., Stubblefield, W.A. (1993). Artificial Intelligence, Structures and Strategies for Complex Problem Solving, CESond Edition, The Benjamin/Cummings Publishing Company, Inc.*
20. *Lunardhi, A.D., Passino, K.M. (1995). Verification of qualitative properties of rule-based expert systems, in Applied Artificial Intelligence, 587-621.*
21. *Mazilescu, V. (2009). 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, 459-464.*
22. *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, 159-165.*
23. *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, pp. 247-255.*
24. *Mazilescu, V. (2012). A Knowledge Management System Embedded in the New Semantic Technologies, Chapter 1, in New Research on Knowledge Management Technologies, edited by Huei-Tse Hou, Intech, Rijeka, pp.1-22.*
25. *McKay, K.N., and Wiers V.C.S (2003). Planning, scheduling and dispatching tasks un production control, Cognition, Technology and Work, Vol.5, No. 2, 82-93.*
26. *Nebel, B., Bäckström, C. (1994). On the computational complexity of temporal projection, planning and plan validation, Artificial Intelligence 66, 125-160.*

27. *Norman, T.J., Preece, A., Chalmers, S., Jennings, N.R., Luck, M., Dang, V.D., Nguyen, T.D., Deora, V., Shao, J., Gray, W.A., Fiddian, N.J. (2004). Agent-based formation of virtual organisations, Knowledge-Based Systems 17, 103–111.*
28. *Stadtler, H. and Kilger, C. (2005), Supply Chain Management and Advanced Planning-Concepts, Models, Software and Case Studies, 3rd ed., Springer, Berlin.*
29. *Stoop, P.M. and Wiers, C.S. (1996). The complexity of scheduling in practice, International Journal of Operations & Production Management, Vol. 16. No. 10, 37-53.*
30. *Vandaie, R. (2008). The role of organizational knowledge management in successful ERP implementation projects, Knowledge-Based Systems 21, 920–926.*
31. *Vollmann, E.T., Berry, W.L., Whybark, C.P., Jacobs, C.P. (2005). Manufacturing Planning and Control for Supply chain management, McGraw-Hill Education, Singapore.*