New Concept of Multi-Agent CAPP System in Inteligent Manufacturing Systems

Ljubomir Lukić, Mirko Djapić, Aleksandra Petrović[1], Veda Kilibarda[2]
[1]Faculty of Mechanical and Civil Engineering in Kraljevo, University of Kragujevac (Serbia)
[2]Belgrade Business School, Belgrade (Serbia)

Modern computer technology has been widely applied in the field of production engineering, design and construction, engineering calculations and analysis, generating NC code, while application of CAPP systems (computer-aided process planning) is still at initial development phase in the area of production processes design in manufacturing systems. The biggest problem in development of CAPP systems for industrial applications is modeling knowledge in the field of metal-cutting(processes.

This paper explores the modeling of technical expertise in metal-cutting processes in a form suitable for the development of CAPP system in intelligent manufacturing systems using agent-oriented software technologies. Focusing on the selection of tools and cutting parameters in the design of machining operations, we first introduce the ontology for the knowledge domain, and then in that context identify and analyze some of the challenges that CAPP presents to the multiagent system architect. In particular, we investigate interactions between operation design and setup design, examine issues arising from global impacts of local decisions in plan construction, and discuss differences between software agents and humans in comparable planning roles. The analysis leads to several multiagent design patterns that help capture domain-specific know-how and integrate it into efficient team behavior.

Keywords: Inteligent manufacturing systems, Multi-agent systems, CAPP systems, CNC systems, Technology design

1. INTRODUCTION

An effective solution for the engineering task of globally planning distributed flexible production while cutting metal can be realized by applying multi-agent architecture [1], as a new area of artificial intelligence [2]. With such systems, distributed computer agents of different specializations work cooperatively to generate a machining plan. The basis for developing a multi-agent CAPP system is formalizing technological knowledge, from technological work piece analysis, clamping, choice and order of technological operations to designing all the technological parameters of the manufacturing process and generating an NC program for tool paths during the cutting process [3], which occurs in intelligent technological systems (IMS).

2. CURENT STATE OF RESEARCH

Of late, experts dealing with looking for an efficient solution for the complex problem of planning globally distributed flexible production see a solution in applying multi-agent architecture [4], a special area of artificial intelligence that has been rapidly developing for the past years. With these systems, distributed computer agents of different specializations work cooperatively to generate a machining plan [5], [6]. Starting from the complexity of the planning the machining process and impossibility of one centralized expert system resolving such a substantial problem, eminent authors [7] had proposed a distributed environment for developing the CAPP system. Their system, CoCAPP has characteristic multi-agent architecture and is completely autonomous, meaning that it is developed independently from existing CAD/CAM systems. The CoCAPP system is projected for a simple integration with CAD/CAM systems; it's flexible to accept new technologies and methods, it can be distributed in multiple different machining centers, it's operative, modular and easily extendible. This system is one of the best known CAPP systems based on multi-agent architecture developed up till now, in the area of designing production technologies in the field of manufacturing by cutting [4]. CoCAPP as well as all other modern CAPP systems that were discussed, is characterized by a fixed multi-agent architecture where each agent has clearly defined expert knowledge for performing a specific activity within the planning process.

This paper shows the research results [8] for the development of a multi-agent model of designing CNC technologies, realized in flexible and intelligent technological systems [1].

3. BASES OF DESIGNING METAL MACHINING PROCESS BY CUTTING, BASED ON INTELLIGENT SOFTWARE AGENTS

Modern CAD, CAM and CAPP systems are based on the Machining feature concept, on which is also based the production strategy for planning clamping, designing the cutting process technology, choosing the cutting tool, designing clamping accessories. A machining feature is a technological form of processing, a parameterized geometrical body to which, apart from natural geometrical attributes have also been assigned the attributes of position, orientation, geometrical tolerance, material qualities, quality of the contact surface between the parameterized geometrical body and the rest of the machining part, as well as references to other technological types of machining within the work piece model (Figure 1).

Engineering the machining plan in technological processes of manufacturing metal by cutting, has some very significant aspects of complexity [9], which must be resolved in a multi-agent CAPP system.

---

*Corresponding author: Faculty of Mechanical and Civil Engineering in Kraljevo, lukic.lj@mfkv.kg.ac.rs

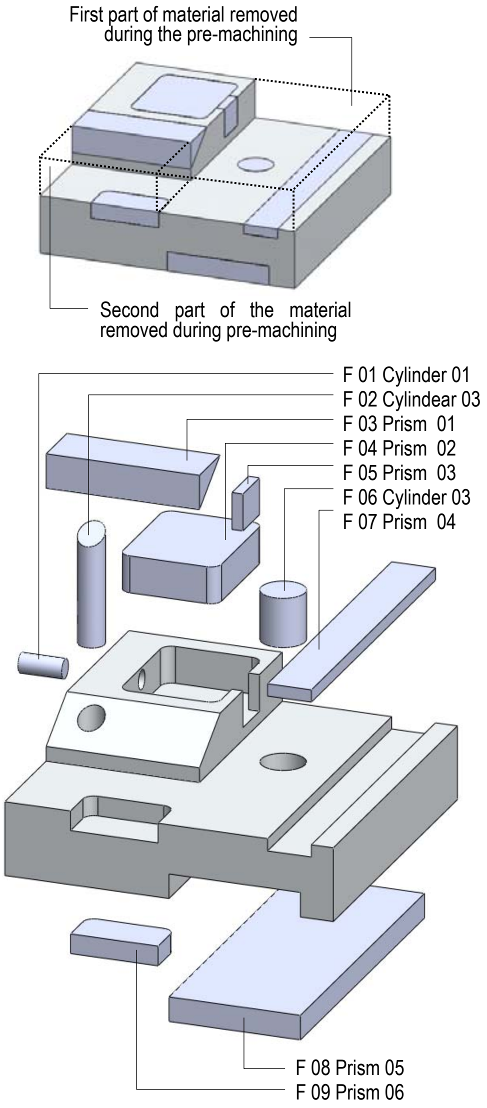


Figure 1: Machining feature and transformation of the raw material throughout the machining process

Combinatory complexity – Generating a processing plan requires the best choice in every step of the decision-making process, which can always be performed in a number of different ways, which for instance leads to a large number of variants when choosing the cutting tool for one and the same technological operation when a machining feature is being processed (Figure 2).

Technological complexity – Identifying the machining feature requires technological know-how for selecting cutting tools, defining the cutting geometry of the tool blade, defining process parameters, planning clamping, designing accessories for clamping and choosing the optimization strategy for global surveillance of all technological indicators of the machining processes in IMS (Figure 3, Figure 4).

Logical complexity – Choice of machining parameters in one step of independent decision making can have an indirect influence on the decisions in the next steps, for instance choice of cutting depth and feed has a direct impact on the intensity of the cutting forces that can be of a satisfactory level from the aspect of dynamic stability of the machining centers, but can be of high intensity for the clamping accessories (Figure 5).

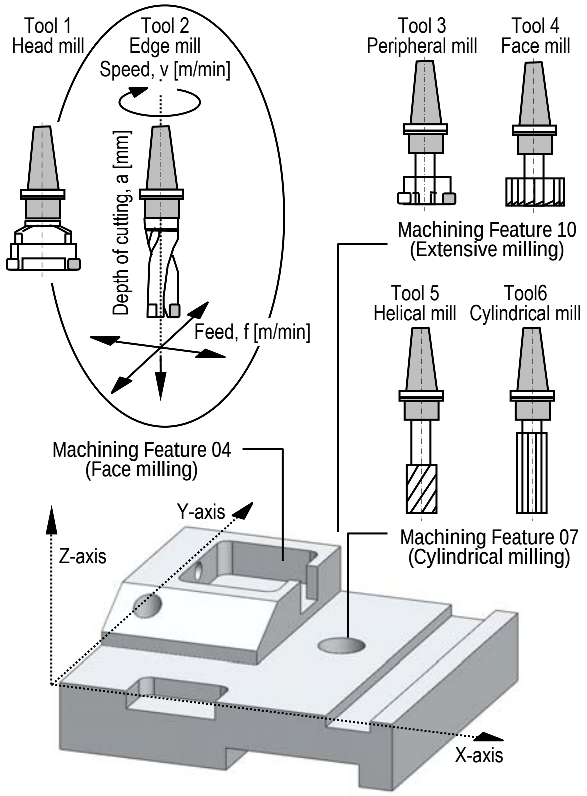


Figure 2: Combinatory complexity in engineering the technological process

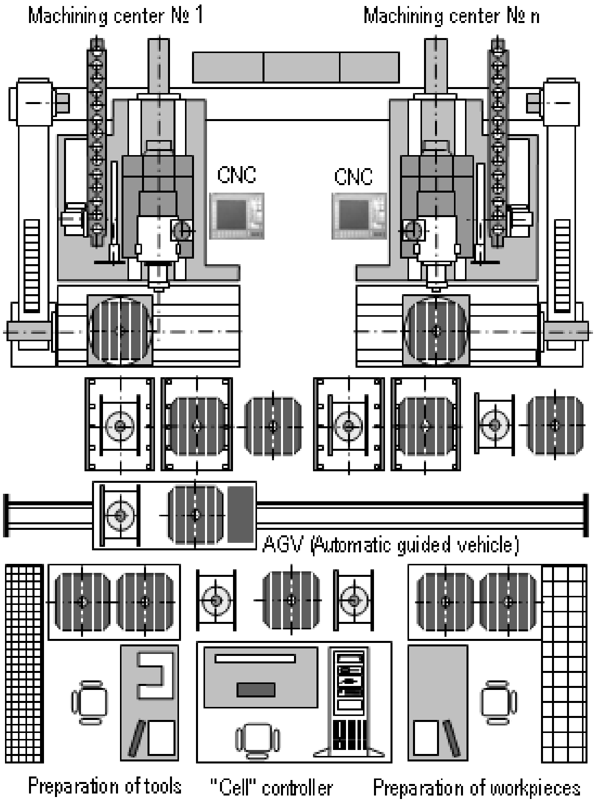


Figure 3: IMS – Inteligent manufacturing system

Social complexity – Communication and coordination requirements are greater from the CAPP system for the distributed organization of modern production, because the sources of technological know-how are even more dispersed between diverse organizations and countries with linguistic, cultural, administrative and other systems differences.

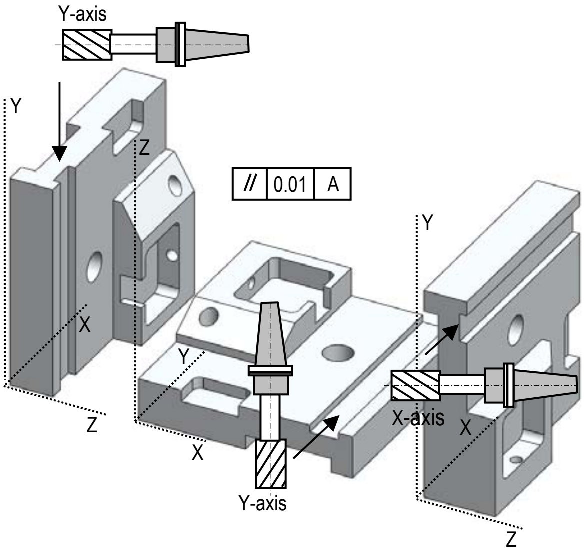


Figure 4: Technological complexity in engineering the machining plan

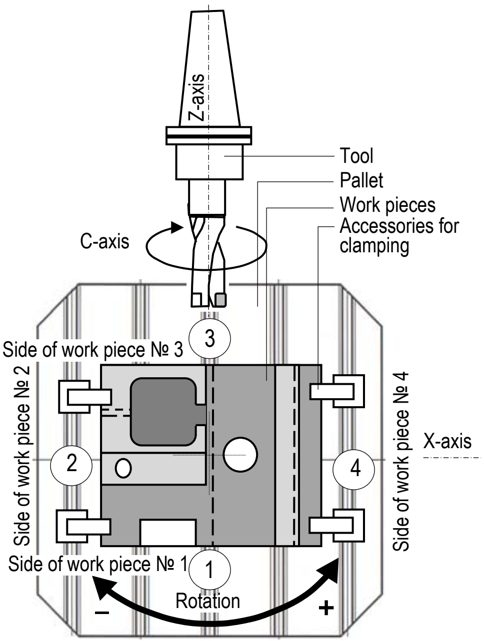


Figure 5: Logical complexity in designing the clamping and order of technological operations

The empirical nature of knowledge on phenomenological occurrences in metal machining processes – Knowledge of the technologies for metal cutting processes are based on empirical and laboratory research in combination with the analytical methods of testing machining parameters, expanding the available know-how base to a wider number of users with the assistance of Internet technologies.

Difficult formalizing of reasoning – engineer, expert immediately discusses the real options when engineering the machining plan, however, intuitive, qualitative and approximate reasoning of a human is difficult to formally express and efficiently emulate with an automated computer system.

4. MULTI-AGENT TEAM FOR ENGINEERING A MACHINING PLAN IN CUTTING PROCESSES

A multi-agent team for engineering a machining plan performs five abstract agent roles that are defined as elements of the metamodel for cooperative planning [1]. These are: the organizer, engineer, evaluator, strategist and interagent (Figure 6).

The organizer gives instructions to individual agent groups or independent agents, decides on activities for making plan elements, selects evaluation methods and plan optimization strategy and it also approves the machining plan.

The designer engineers elements of the machining plan based on technological competence and its specialization in certain phases of designing the technological process.

The evaluator calculates the values of performance indicators of the machining plan based on metrics for plan evaluation (calculates the total machining time for clamping in order to optimally plan the transport process and use the machining centers.

A strategist uses evaluation results to determine and suggest how to optimize the current version of the machining plan.

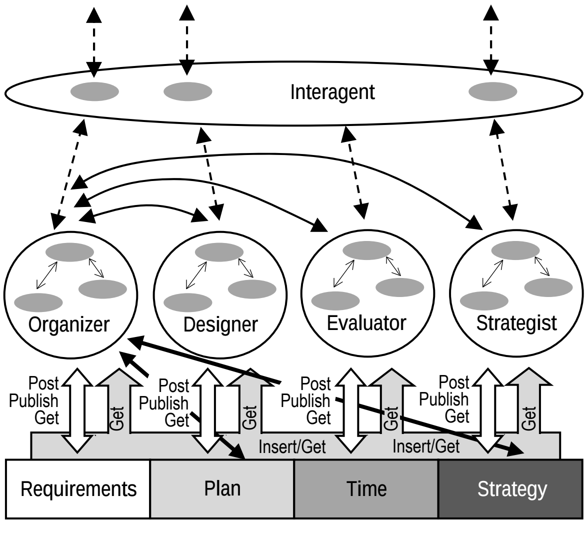


Figure 6: General architecture of a multi-agent CAPP system for designing the machining plan in cutting processes

The interagent is an active, intelligent interface according to some other segment of the production process (specialized roles like the CAD interagent or the interagent distributor are necessary to provide the dynamic interaction of the plan team with other elements of production [15].

Formally the CAPP plan multi-agent team is made up of four people

$$T = (Roles, Kom, A, \alpha)$$

where:

- *Roles*, a set of agent role instances,
- *Kom* = (*E, Prot*), communication model comprised of environment *E* and a set of protocols *Prot* with which define the mutual interactions of agent role instances and environment instances,
- *A*, is a set of agents, members of the multi-agent team and
- α: *Role* $\rightarrow$ *A,* replication, role assignment.

The main feature of rational agents is their mental state. The features of the mental state of an agent, member of the plan team for the CAPP system, reflect domain-specific knowledge and assigned role within the plan team [10]. When describing the mental states and mutual interactions of a multi-agent plan team of the CAPP system, we use the paradigm Belief-Desire-Intention [11], [4]. This entails:

- Combined beliefs,
- Combined wishes and intentions,
- Personal beliefs,
- Personal intentions,
- Relationship between combined and personal beliefs-wishes-intentions.

Defining the mental state of individual agents and agent groups is done with primitive operators (Figure 7).

(Bel i φ)	agent i believe that it is φ
(Goal i φ)	agent i has goal φ
($\tau = \tau'$)	term τ is equal to term τ'
($i \in g$)	agent i isa member of agent group g
(Agts a g)	agent group g needs to perform a series of actions a
A φ	φ applies on all paths of temporal stub A
(Happens a)	the next action that will happen is a

(M-Bel g φ**)**	agent group g believes in φ
(M-Goal g φ**)**	agent group g has goal φ
(J-Commit g φ ψ χ c**)**	agent group g has a combined commitment to goal φ in relation to motivation ψ, with preconditional χ, and custom c

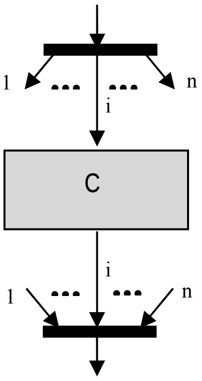
Figure 7: Primitive operators of modal logic for defining the mental state of the agent and group of agents

In order to develop the agent team model, we use multi-modal logic [11]. The environment model of a multi–agent CAPP system is presented with a time tree where traveling through the tree represents the potential histories, states and environments.

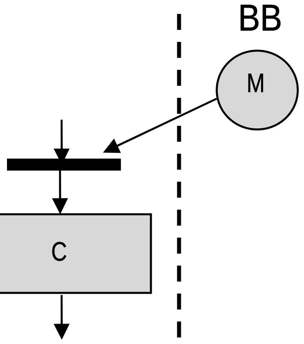
A tree has its own start, the moment when the plan organizer (OP) puts the task of planning into the environment, and the end is defined by accepting a version of the CAPP system plan. Tree nodes are the environment states. Branches, representing branching over time, are marked with primitive actions that transform one state into another. Each of these primitive actions is associated with an agent or agent team [12], [13].

5. PROCEDURE FOR DEFINING THE WORK ALGORYTHM OF A MULTI_AGENT TEAM

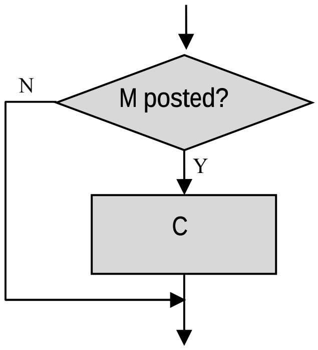
To define the algorithm the necessary lingual elements are defined that together with graphic symbols make up an extension of standard UML activity diagrams when designing the machining plan (Figure 8).



foreach < *iterator* > **do** < *command* >



when < *model component* > **posted** < *command* >



if < *model component* > **posted**

Figure 8: UML activity diagram element and pseudo code language

Apart from standard lingual elements, like ***while*** and ***for*** loops, in this pseudo code are introduced commands used for external synchronization with information being received from the blackboard. In that way, a command with syntax “**when** *m* **posted** *C*” stops the work of the agent while component *m* is not posted on the blackboard, and then continues executing *C*. The agent subscribed to component *m*, may, once having executed the ***get*** instruction, take the information from the component. Command “**when** *m* **published** *C*” has a similar meaning to command “**when** *m* **posted** *C*” except that it’s not necessary for the agent to execute command ***get*** because the value of the component is in variable *m*.

The versions of these commands that do not block executions are conditional commands "**if** *m* **posted** C_1 [**else** C_2]" and "**if** *m* **published** C_1 [**else** C_2]", where the part in square brackets is a possible option. With these commands the agent executes expression C_1 if *m* was either posted or published on the blackboard and if not, and there is no **else** option, it prolongs its normal execution flow. The existence of option **else** conditions first the execution of C_2 then the continuation of the normal flow. Another important command "**foreach** $i \in S$ **do** C^m", is actually a loop with multiple concurrent multithreaded loops that usually end on the same processor.

6. DESIGNONG THE PROCESSING PLAN

When generating the initial version of the processing plan all the agents to which have been assigned specialist roles of engineers and which make up the project sub-team have been directly included, and they are:

- Designer of the Machining Feature Model (DMFM),
- Designer for Clamping (DC),
- Designer of Clamping Accessories (DCA),
- Designer of Machining Types Model (DMMT),
- Designer of the Machining Processes Method Model (DMMPM),
- Designer for the Selection of Cutting Tools (DSCT) and
- Designer for Tool Settings (DTS).

The basic characteristics of the generative way of designing the machining plan by a multi-agent plan team from the viewpoint of decision-making activities can be summarized as follows:

1. Decisions when recognizing the machining feature.
2. Decision when choosing machining process types.
3. Decisions when designing clamping.

The task of planning the machining process can be presented as an assortment of four elements

$$T = (A, C, E, R),$$

where A is the assortment, C is the configuration of the manufacturing system, *E* the evaluation metrics and R requests. Decision making problems must be observed simultaneously in order to design an optimal solution for the machining plan for the posed technological task in conditions when the planning process becomes a combinatory problem [14], [15].

The Machining feature model (MFM) of the work piece is an integration element of all activities that preclude the production process, such as generating the work piece plan, generating the project of the clamping tools and tool plan.

In order to generate the work piece performance design (DPW) the agent uses project models from the combined environment to which he gets by perceiving the common environment.. By using personal beliefs and know-how of the rules for determining material accessories for the machining process. DMFM defines its wishes and intentions to find a favorable solution for projects of the raw material, by determining the processing accessories and generating the machining feature model (MFM), based on recognizing the elementary features that through the clamping process must be turned into chips of metal by respecting the STEP standard. It uses machining parameters to define wishes for determining geometrical shapes and material volume that needs to be removed from the raw material in order to get the shape defined by the project. It expresses intentions by decomposing the total volume into the machining feature, such as the through-hole, blind hole, ring, notches of different shapes, stepped slits etc. that can be processed with operations for removing metal chippings – i.e. drilling, chipping or milling.

Since the solution can come in several variations, MFM, DMFM publishes one of the solutions for any project model from the assortment into the space of combined beliefs while it places alternative solutions into the area of personal beliefs. Recognizing the shape and generating MFM is not a single step, but in the optimization process can be repeated, which is why alternative solutions are kept.

While DMFM is looking for a favorable solution for MFM_i, the DCA updates personal beliefs on the information about MPW_i that will enable him to achieve its purpose of determining the appropriate surfaces of the raw material for locating and clamping the work piece, i.e. the reference surfaces necessary when placing the work piece onto the work table of the machine. From multiple possible solutions, the DCA opts for one adequate solution which it posts on the BB, thereby updating the combined beliefs of the plan team. After having found favorable solutions for MFM_i, DMFM selects one and posts it on the BB, while it stores other solutions into its personal beliefs for potential future use. The machining type model (MMT) is created by copying individual machining features and their relations from MFM into one or more technological operations based on the type of the machining feature (hole, slot and so on), dimensions, tolerances and quality of the processed surface on the work piece that borders with the technological shape. Thus, for example, for the technological shape of a hole the size of 20 mm without tolerance or quality of the processed surface, the type of process used will be drilling, while in the case of a defined tolerance and quality of the processed surface center drilling-drilling-enlarging or central drilling- drilling-reaming can be used. For each technological shape, DMMT first determines the type of process (chipping, drilling and/or milling) and type of the technological operation inside the processing type, such as for instance vertical chipping or peripheral chipping (Figure 8). For this purpose the agent uses a knowledge database formed based on the model of the technological work piece processing by cutting. By using the information from an updated storage of personal beliefs about the technological shape concerned, DMMT creates wishes containing the possible types of technological operations. When forming wishes it uses know-how about the available technological system providing the feasibility of the technological operation.

To design a processing plan is required the cooperative work of several agents: organizer-strategist (OS), designer of the machining feature model (DMFM), designer of clamping (DC) and designer of model of machining types (DMMT) which work together through a blackboard (BB). Agent DMMPM has activities concerning the selection of the cutting tool (characterized by its geometry, tool material), cutting speed and feed when cutting during a given type of technological operation, work piece material and tool type.

```
agent Designer of Model of Machining Types {
    when DT posted {
        get $n$ =BB.size of product range from $\mathcal{A}$;
        get BB.machining constraints from $C$ ;
        get BB.team goals from $\mathcal{R}$;
        make design criteria for machining;
    }
    foreach $i \in \{1 \ldots n\}$ do {
        when $MTO_i$ posted
            get $F_i$ =BB.a set of index feature ($MFM_i$);
        for $f \in F_i$ do {
            get BB.descriptionof the feature(f);
            make MMT(f);
        }
        make $MMT_i$ ;
        post BB.$MMT_i$ ;
    }
}
```

Figure 9. An example of the agent algorithm pseudo code designer of machining types model (DMMT)

Numerous alternative possibilities for choosing the parameters on the level of individual operations generate a very large space for potentially satisfactory solutions, but in the agent team there must be such a decision making order to avoid a detailed elaboration of inadequate solutions. It's very important to notice as early as possible for example the incompatibility of the suggested type of technological operation and solution for clamping accessories so as not to overstep permitted forces, in order to repeat the selection of adequate machining parameters.

7. CONCLUSIONS

The further course of research refers to modeling the mental state of individual agents and agent teams, connected to the CAPP domain and its specifics. Of particular interest is research on the development of suitable learning methods that would rely on previous agent experiences and reasoning methods of agents-experts, to quickly get to the desired goal. This entails the modeling of intuitive reasoning and deduction based on the experience of particular significance for planning metal processing by cutting. From a research point of view, we need to formally specify the mental states of agents by using multi-modal logic and develop learning modules alongside the existing knowledge databases for cutting processes.

One course for future research refers to multi-agent methodology in particular. When it comes to the environment in which the agents exist, two approaches are possible for their modeling. In one approach, the environment is modeled as a passive component, while in the other it is modeled as an active component of the multi-agent team. This vagueness comes from a lack of efficient mechanisms for software modeling of the environment, which also represents a research challenge. Apart from researching the necessary mechanisms for an efficient distribution of the environment, we need to bear in mind the mobility of software agents across the internet.

REFERENCES

[1] D.Polajnar, “Multi-agent model designing of CNC technologies in intelligent manufacturing systems”, PhD Dissertation, University of Kragujevac, Faculty of Mechanical Engineering in Kraljevo (2010).

[2] D.Polajnar, Lj. Lukic and, V. Solaja, “An expert system for the design of cutting processes in FMS”, Proc. of Factories of the future CARS & FOF’94 conf., Ottawa, Canada, August 21-24 (1994).

[3] D.Polajnar, J. Polajnar, Lj. Lukic and M. Djapic, “Complexity Challenges in CAPP Systems and Promises of Multi-Agent Technology”, Proceedings The Sixth Triennial International Conference "Heavy Machinery HM 2008", Kraljevo, 24-29 June 2008, pp.F.19-P.24.

[4] M.Wooldrige, An Introduction to MultiAgent Systems, Wiley (2009).

[5] Lj. Lukic, “Flexible Manufacturing Systems – Structure, construction, control and technology”, Monographs, University of Kragujevac, Faculty of Mechanical Engineering of Kraljevo (2008).

[6] W.J.Zhang and S.Q.Xie, “Agent technology for collaborative process planning: a review”, The International Journal of Advanced Manufacturing, 3(2007), pp.41-71.

[7] F.L.Zhao, S.K.Tso and P.S.Y.Wu, “A cooperative agent modeling approach for process planning”, Computers in Industry, 41(2000), pp.83-97.

[8] Lj.Lukic, D.Polajnar and V.Solaja, “A Yugoslav approach to decision support for optimization of FMS technologies”, In Annals of the CIRP, vol. 40 (1991)1, pp. 99–102.

[9] V.Solaja, M.Dimitric and Lj.Lukic, “On the Two Cases of Yugoslav Attempts in Adaptive Control”, Robotics & Computer Integrated Manufacturing 4(1988)1/2, pp.241-244.

[10] D.Polajnar, J.Polajnar and Lj.Lukic, “Metamodel Abstractions of Agent Roles in Cooperative Process Planning”, Proc. IEEE SMC Int. Conf. on Distributed Human-Machine Systems, (2008), pp. 77-82.

[11] A.Rao and M.Georgeff, “BDI-agents: from theory to practice”, Proceedings of the First International Conference on Multiagent Systems, San Francisco (1995), pp.185-204.

[12] F.Zambonelli, N.Jennings and M.Wooldridge, “Developing multiagent systems: The methodology”, ACM Transactions Software Engineering Methodology, 12 (2003)3, pp 317-370.

[13] C.Carrascosa, J.Bajo, V.Julian, J.Corchado and V.Botti, “Hybrid multi-agent architecture as a real-time problem-solving model”, Expert Systems with Applications, 34 (2008)1, pp 2-17.

[14] P. Dutta, N.Jennings and L.Moreau, “Cooperative information sharing to improve distributed learning in multi-agent systems”, Journal of Artificial Inteligence Research 24 (2005), pp. 407–463.