CODEN STJSAO
ZX470/1503
ISSN 0562-1887
UDK 004.822:004.94:658.51.012:519.172

Design Methodology for Discrete Event Simulation Solutions in Manufacturing Environment

Milan PAVLOVIĆ*[1], *Slavko ARSOVSKI*[2], *Zora ARSOVSKI*[3], *Zoran MIROVIĆ*[2] and *Miodrag LAZIĆ

1) Fakultet tehničkih nauka u Zrenjaninu, Univerzitet u Novom Sadu (Faculty of Technical Sciences in Zrenjanin University of Novi Sad), Đure Đakovića bb, 23000 Zrenjanin, **Republic of Serbia**

2) Mašinski fakultet Univerzitet u Kragujevcu (Faculty of Mechanical Engineering University of Kragujevac) Sestre Janjić 6, 34000 Kragujevac, **Republic of Serbia**

3) Ekonomski fakultet u Kragujevcu, Univezitet u Kragujevcu (Faculty of Economic, University of Kragujevac), Đure Pucara Starog 3, 34000 Kragujevac, **Republic of Serbia**

pmilan@sbb.rs

Preliminary note

The paper considers Discrete Event Simulation design methodology. Primary objective is proposal of comprehensive, scientifically established design methodology to structure, guide, and improve manufacturing processes modeling efforts. Proposed approach replicate structure and behavior of original system at desired level of abstraction, incorporates specific academic knowledge in simulation solution and results in object oriented application architecture completely consistent with production planning and scheduling ontology. The paper begins pointing out importance of modeling and equality between knowledge and models from viewpoint of modern cybernetic science and referenced cybernetics theory relocates in manufacturing systems and discrete event simulation domain. Paper then proceeds to discuss a structure and main steps of proposed methodology with a more detailed discussion of simulation based production planning and scheduling. Finally, an example of successfully designed simulation solution is given. Proposed approach gives verbal and mathematical problem description, builds ontology of problem domain, uses Extended Petri Nets, event graphs and activity cycle diagrams as modeling tools, in order to obtain faithful model which easily can be replicated in object oriented class and object hierarchy. Methodology highlights inevitability of knowledge transfer between business processes, software development and academic research experts. Integration and overlapping of mentioned fields of knowledge result in object oriented application architecture fully consistent with ontology derived from conceptual phase of design methodology. Proposed methodology enables convergence of comprehensive but static MIS knowledge in dynamic simulation model in order to fully utilize its prediction power for effective integration of strategic and tactical decision making.

Metodologija dizajniranja simulacijskih sustava temeljenih na metodi diskretnih događaja u proizvodnom ambijentu

Prethodno priopćenje

U radu se razmatra metodologija dizajniranja računalnih simulacijskih sustava temeljenih na metodi diskretnih događaja. Primarni cilj je prijedlog sveobuhvatne, znanstveno utemeljene metodologije dizajniranja koja formulira, usmjerava te poboljšava nastojanja modeliranja proizvodnih procesa. Predloženi pristup replicira strukturu i ponašanje originalnih sustava na željenoj razini apstrakcije, inkorporira specifična akademska znanja u simulacijski sustav te rezultira u objektno orijentiranoj arhitekturi aplikacije koja je u potpunosti konzistentna sa ontologijom planiranja i terminiranja proizvodnje. Rad započinje naglašavanjem značenja modeliranja stavljajući znak jednakosti između znanja i modela iz perspektive moderne kibernetičke znanosti te pomenutu kibernetičku teoriju relocira u domene proizvodnih sustava i simulacije diskretnih događaja. U radu se potom daju osnovni podaci o strukturi i glavnim koracima predložene metodologije uz detaljnije objašnjenje planiranja i terminiranja proizvodnje uz pomoć simulacije. Na kraju se opisuje primjer uspješno dizajniranog simulacijskog sustava. Predloženi pristup daje verbalni i matematički opis problema, gradi ontologiju domene problema, koristi proširene Petrijeve mreže, graf događaje i cikluse dijagrama aktivnosti kao alate modeliranja, sa svrhom dobivanja vjernog modela koji se lako može replicirati u objektno orijentiranu hijerarhiju klasa i objekata. Metodologija ističe neizbježnost transfera znanja između stručnjaka za poslovne procese, te eksperata iz oblasti razvoja softvera i akademskih istraživanja. Integracija i preklapanje navedenih polja znanja rezultira u objektno orijentiranoj arhitekturi aplikacije koja je potpuno u skladu s ontologijom koja slijedi iz konceptualne faze metodologije dizajniranja. Predložena metodologija omogućava konvergenciju sveobuhvatnog ali statičkog znanja sadržanog u informacijskom sustavu upravljanja u dinamički simulacijski model, sa ciljem potpunijeg iskorišćenja mogućnosti predviđanja te učinkovito integriranje strateškog i taktičkog odlučivanja.

Keywords
Discrete Event Simulation Methodology
Manufacturing planning and scheduling
Petri Nets

Ključne riječi
Metodologija dizajniranja simulacije diskretnih događanja
Petrijeve mreže
Planiranje i terminiranje proizvodnje

Received (primljeno): 2009-11-30
Accepted (prihvaćeno): 2010-01-25

1. Introduction

Discrete Event Simulation (DES) software solution is a dynamic model of an actual dynamic system for the purpose either of understanding the behavior of the system or of evaluating various strategies (within the restrictions forced by set of criteria) for the control of the system operation. The strength of DES solutions is their ability to mimic the dynamics of complex structure and behavior of real system. Many models, including advanced optimization models, cannot take into account the dynamics of a real system. It is the capability to mimic the dynamics of the real system that gives Discrete Event Simulation its unique way to analyze obtained results.

Simulation software systems are dynamic models of complex reality and this fact result in complexity of all stages of software development. The paper suggests methodological design approach for Discrete Event Simulation software solution. Proposed methodology is applied in environment of operational planning, scheduling, and control of manufacturing systems.

The approach proposed in the paper gives verbal and mathematical problem description, builds ontology of problem domain, uses Extended Petri Nets and event graphs as activity cycle diagrams [1, 5-6] as modeling tools, in order to obtain faithful model which easily can be replicated in object oriented class and object hierarchy [7, 10].

The paper also highlights unavoidability of knowledge transfer between four expert groups from different knowledge domains. First group are business processes management experts, second software development or Information Technology (IT) experts, third from academic community research and user experts as fourth. Synthesis of mentioned fields of knowledge in proposed approach result in effective and functional simulation model.

Structure of the paper is as follows. The paper begins referencing the fundamental cybernetic concepts pointing out its strength and universality and equating cybernetic concepts with corresponding concepts in computer integrated manufacturing environment.

Concepts like control, controller, controlled system, control actions, system dynamics, perception, control model, representation and observer have its counterparts in the manufacturing environment (Table 1). Thus management, manager, manufacturing process, management actions, behavioral aspects of the manufacturing process, information flow between Manufacturing Execution System and Management Information System, Management Information System integrated with Discrete Event Simulation solution acquire mentioned strength and universality of fundamental cybernetic concepts definitions.

Table 1. Cybernetics and Manufacturing concepts

Tabela 1. Kibernetički i proizvodni koncepti

Concepts /	**Koncept**
Cybernetics / Kibernetika	**Manufacturing / Proizvodnja**
Control / Upravljanje	Management / Menadžment
Controller, Observer / Upravljač, promatrač	Manager / Menadžer
Controlled system / Sustav kojim se upravlja	Manufacturing process / Proizvodni proces
Control actions / Upravljačke akcije	Management actions / Akcije menadžmenta
System Dynamics / Dinamika sustava	Behavioral aspects of the manufacturing process / Aspekt ponašanja proizvodnih procesa
Perception / Percepcija	Information flow between MES and MIS / Tok informacija između informacijskog sustava proizvodnje i upravljačkog informacijskog sustava
Control model / Model upravljanja	ICT solutions / Informatička rješenja
Representation / Reprezentacija	Information on ICT solution presentation layer / Informacije na sloju prezentacije informatičkih rješenja

In the paper, as cybernetic theory suggests, knowledge is understood as consisting of models. Knowledge is the instrument of survival, adaptation and growth of a system and this is achieved bay anticipation of possible perturbations or effects of unavoidable disturbances. So true knowledge is consists of models as generators of predictions and only criterion for true knowledge is the prediction capability it gives. In this context, the paper highlights modeling power of Discrete Event Simulation software and its capability for prediction as well as multidimensional representation of knowledge.

In the next section the paper states that integration and overlapping of mentioned specialized fields of knowledge result in multidimensional conceptual framework. The conceptual framework is a base for domain ontology building. The domain ontology is a means of bridging the gap between domain analysis and application system construction. On this basis, concepts defined in the domain ontology can be related to software objects descriptions, acting both as mechanisms for pointing to relevant software components and as specifications of overall configuration requirements.

The next section gives verbal and mathematical description of production planning and scheduling problem, builds ontology of problem domain [3-4, 12] uses Extended Petri Nets, event graphs and activity cycle diagrams as modeling tools, in order to obtain faithful model which easily can be replicated in object oriented class and object hierarchy.

Integration and overlapping of different fields of knowledge performed at beginning of the simulation study [13], and followed with proposed methodology steps, result in object oriented application architecture fully consistent with ontology derived from conceptual phase of design methodology. Proposed methodology enables convergence of comprehensive but static Manufacturing Execution System knowledge in dynamic simulation model in order to fully utilize its prediction power for effective integration of strategic and tactical decision making.

The paper continues describing every step of proposed methodology giving in the last section an example of successfully designed simulation solution

2. Theoretical background

2.1. Cybernetic conceptual framework

Cybernetics studies organization, communication and control in complex systems by focusing on circular (feedback) mechanisms. It grew out of Shannon's information theory, which was designed to optimize the transmission of information through communication channels, and the feedback concept used in engineering control systems. Emphasis of modern, second-order cybernetics is on how observers construct control models of the systems with which they interact to maintain, adapt, and self-organize.

Such circularity or self-reference makes it possible to make precise, scientific models of *purposeful* activity, that is, behavior that is oriented towards a goal or preferred condition.

The fundamental concepts of cybernetics have proven to be very powerful in a variety of science disciplines. Many of contemporary approaches in management and computer science, in the context of the paper, have their roots or can be explained by using cybernetics concepts. As an example, some concepts relating to cybernetic systems like *complexity, mutuality, complementarity* and *evolvability* are in full accordance with manufacturing system definition:

Manufacturing (cybernetic) systems are complex structures, with many heterogeneous interacting components (*complexity*). These many components interact in parallel, cooperatively, and in real time, creating multiple simultaneous interactions among subsystems (*mutuality*). These many simultaneous modes of interaction lead to subsystems which participate in multiple processes and structures, yielding any single dimension of description incomplete, and requiring multiple complementary, irreducible levels of analysis (*complementarity*). Manufacturing (cybernetic) systems tend to evolve and grow in an opportunistic manner, rather than be designed and planned in an optimal manner (*evolvability*).

Management can be defined as process of administering an enterprise including: development of corporate strategy and long range planning (strategic), regulation, coordination and control of functional processes as production, marketing, research and development etc. (functional) and supervision of operations being performed (operational). The important aspect to effective management is the adequate flow of information between and within strategic, functional and operational levels in order to allow for timely and appropriate decisions to be made. Management information system constitutes a technological solution to information flow problems. Another important aspect is the form of control exercised through establishing objectives providing working schemes for functional processes (production, etc).

2.2. Control

The concept of control is the cornerstone of cybernetics and can be defined as operation mode of a manufacturing (cybernetic or *control) system* which includes two subsystems: controlling (management), and controlled (manufacturing process). They interact, but there is a difference between the management action on manufacturing process, and the action of manufacturing or controlled process on management. The management may change the state of the process using control actions. The action of controlled process on control system is formation of a *perception* of system state space. This understanding of control, in the context of the paper, is presented in Figure 1. Controlled system is described by using variables directly *affected* by the control actions and variables *observed* by the controller in perception. The relation between observed variables and the affected variables is determined by the intrinsic dynamics of the system. Integral parts of the control representation are effects of uncontrollable disturbances on the observed variables.

Controller is an *agent* which is responsible for its actions and also for a *representation* of the controlled system. Re*presentation* is an object whose states controller recognize with perceptions. The relation between representation and agent is described as a flow of *information*. The actions of the agent depend on the flow of *information*. Thus the action of controlled system on controller (perception) is limited in its effect, by changing only its own representation. The action of controller on controlled system is directed through the representation: its effect on controlled system cannot be greater than allowed by the changing state of the representation.

Goal is one more important object which influences the controller as depicted in Figure 1. The agent compares the current representation with the goal and takes control actions which tend to minimize the difference between them. This is known as *purposeful behavior* or *behavior* that is oriented towards a goal. Even though the relation of control is asymmetric, it includes a closed loop. Looked from the controller, the loop starts with its action and is followed by a perception, which is an action in the opposite direction: from the controlled to the controller. This aspect of control relation is known as feedback.

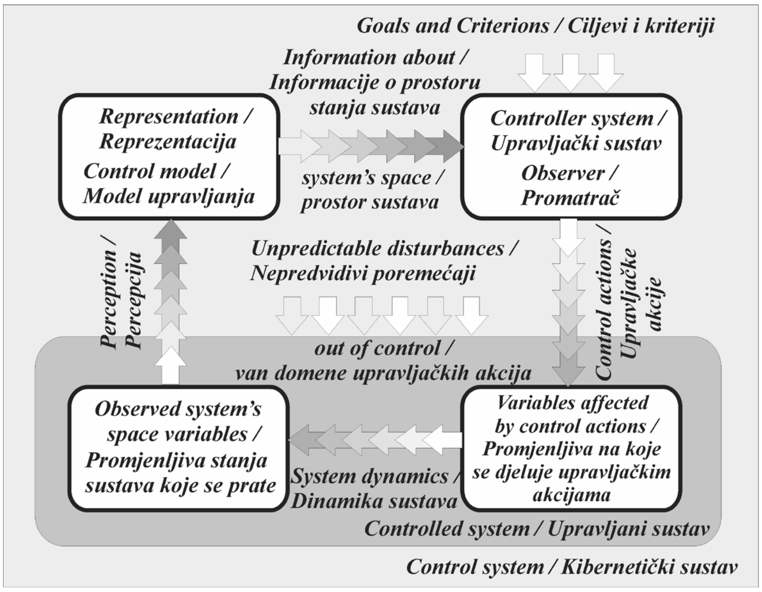


Figure 1. Basic cybernetic control scheme

Slika 1. Osnovna kibernetička shema upravljanja

In cybernetic scheme of control, controller is the same as observer. In modern cybernetic theory observer is defined as a system which, through recursive interactions with its own linguistic states, may always linguistically interact with its own states as if with representations of its interactions. Knowledge of reality is dependent upon the perceptions passing through control model and information flow to the observer.

2.3. Control model

In the context of the simulation of manufacturing systems a simulation model is software system which shares crucial properties of modeled, original system and determines how key properties of the original will change over time. A model is typically used in systems analysis to predict the consequences of a course of action.

Models are not static reflections of the modeled environment, but dynamic constructions achieved through trial-and-error by the individual. What models represent is not only the structure of the environment but also its behavior, insofar as it has an influence on the system. Models function as recursive generators of predictions about the world and itself. As a consequence of complexity and multidimensional, hierarchical nature of real systems there is no absolutely accurate model of reality: there are many different models, any of which may be adequate for solving particular problems, but no model is capable to solve all problems.

2.4. Knowledge and control models

Also, in modern, second order cybernetic theory knowledge is understood as consisting of models that allow the survival, adaptation and growth of a cybernetic system in its environment, by anticipation of possible perturbations or effects of unavoidable disturbances (Figure 1.).

A cybernetic system makes predictions by its knowledge i.e. models in order to achieve certain goal, ultimately – survival, adaptation and growth. So true knowledge is an instrument of survival, adaptation and growth and true knowledge consists of models – generators of predictions. There is no any criterion of true knowledge other than the prediction power it gives.

2.5. Simulation as dynamic control model

At this point, the paper highlights correlation between cybernetics concepts and corresponding conceptual framework of modern computer integrated manufacturing system in the context of the paper.

Controller is equated with manufacturing management, controlled system with managed manufacturing processes, goal is defined by production planning and scheduling policy, control actions is information flow between Management Information System and Manufacturing Execution System, system dynamics caused by control actions is functional aspect of manufacturing system and perception is correlated with information flow between Manufacturing Execution System and Management Information System.

Control model as central object of cybernetic control scheme, as well as of this paper, is equated with Management Information System integrated with Discrete Event Simulation system. Management Information System delivers information about existing status of manufacturing system environment and Discrete Event Simulation system is generator of predictions and anticipations of effects caused by unavoidable disturbances or supply, operational and demand exceptions. Primary purpose of Discrete Event

Simulation system is generation of predictions caused by possible management decisions.

In proposed conceptual framework, management is an observer which through recursive linguistic interactions with Discrete Event Simulation system acquires knowledge essential for timely and correct management decisions to be made.

Cited cybernetic theory states that control of any real system is not possible without corresponding control model. This is evident in cybernetic scheme of control (Figure 1.) where control model (identical with representation) transforms action of controlled system (perception) to flow of information. Additionally, control model must be, as much as possible, replication of the real system because effectiveness of control depends on similarity between modeled system and corresponding model. This is evident from statement that observer's knowledge about consequences of its control actions is fully dependent upon control model (representation of perception).

For the aims of this paper cited general definitions could be interpreted as follows:

Prediction power of control model is crucial for timely and appropriate management decisions. The paper proposes integration of Management Information System and Discrete Event Simulation system as control model.

Management Information System:

- is basic control model of structure and behavior of the manufacturing system,
- incorporates all knowledge about the manufacturing system necessary for system function and control in line with business goals,
- is a static model of dynamic and complex structure and behavior of business system without power of prediction and
- represents manufacturing system on the level of abstraction that is not adequate for decision making on the operational and tactical level.

Discrete Event Simulation system:

- is basic control model of structure and behavior of the manufacturing process,
- incorporates all knowledge about the manufacturing process necessary for process function and control in line with tactical and operational goals,
- is a dynamic model of dynamic and complex structure and behavior of business process with power of prediction and
- represents manufacturing process on the level of abstraction that is adequate for decision making on the operational and tactical level.

1. Any single dimension of description of complex and dynamic manufacturing system is incomplete. The paper, in the conceptual phase of DES methodology, proposes integration of specialized fields of knowledge as well as includes all potential users of simulation software. The goal is multidimensional description (conceptual framework, problem domain ontology) of system's structure and behavioral states. The multidimensional description leads to multidimensional representation what makes possible observer's linguistic interaction with its own linguistic states.
2. The multidimensional representation, equated in the paper with DES software solution or a control model, must be, as much as possible, replication of the real system because effectiveness of control depends on similarity between modeled system and corresponding model. The proposed methodology makes an attempt to achieve this goal using Extended Petri Nets and event graphs as activity cycle diagrams as modeling tools. Developed ontology, used together with mentioned modeling tools is used as a base for an object oriented analysis software development phase. Obtained data easily can be replicated in object oriented, class and object hierarchy in software design phase.
3. From the software engineering viewpoint DES software solution is a data centric application with fully object oriented architecture. Currently simulation applications are written on top of relational databases of Management Information System. This disturbs consistency between the conceptual level of abstraction, obtained in conceptual phase, and building blocks of fully object oriented application architecture. Proposed methodology overcomes this impedance mismatch problem between object oriented and relational models using recent technologies. In this case, DES application is relatively independent and, in the same time, fully integrated with MIS databases.

3. Applied methodology and review of concrete solution

A software development methodology is a framework that is used to structure, plan, and control the process of developing software systems. Each of the available methodologies is best suited to specific kinds of projects, based on various technical, organizational, project and team considerations inside software development company. All companies use some kind of project management strategy, which give guidelines for how the project should be carried out.

The simulation software development process can be divided in the following phases: functional, conceptual, design, realization, experimental, implementation, and review and learning phase.

This paper considers a practical aspect of software development methodology with an emphasis on the object oriented analysis as well as object oriented design phase of DES simulation solutions. The analysis phase consists of functional and conceptual phases while design phase deal with models of data and business logic layers of simulation application architecture.

Proposed methodology is applied in environment of operational planning, scheduling, and control of manufacturing systems. While planning is concerned with the long-range determination of what needs to be manufactured, typically over a relatively long time period, scheduling is the task of deciding how that manufacturing is to be accomplished, usually over a relatively short time period.

The operation scheduling in a computer integrated environment belongs to the class of problems that are too complex for mathematical formulation and whose optimal solutions are feasible by traditional operational research techniques only for small-scale or idealized scheduling problems. Solutions require knowledge-based systems that combine simulation techniques with those of current expert systems.

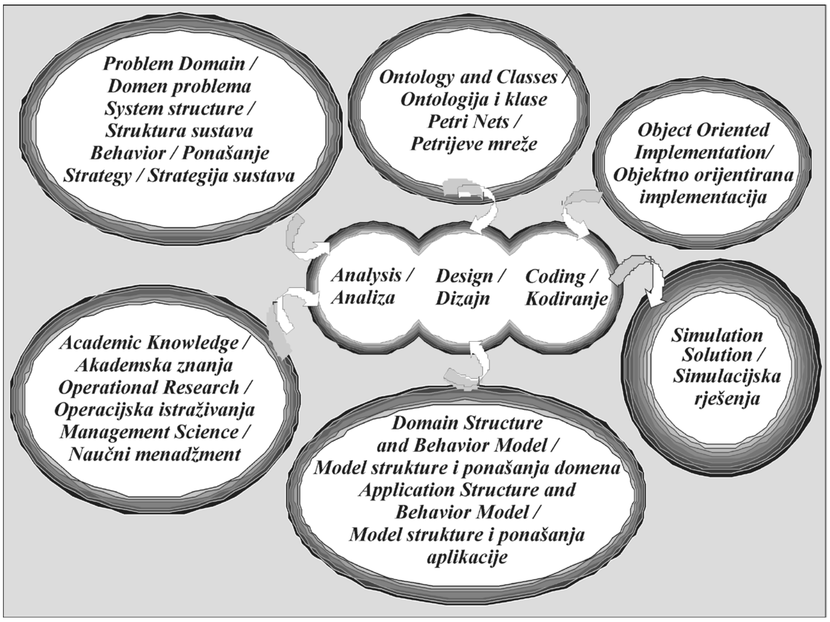


Figure 3. Integrating knowledge for multidimensional representation

Slika 3. Integriranje znanja u cilju multidimenzionalne reprezentacije

3.1. The functional and conceptual phases

In the functional phase all participants in DES simulation study document information requirements and create functional baseline for project realization. The participants in a simulation study are information technology, business processes management and operational research experts as well as future DES users experts. Figure 3. presents allocation of different group of participants in DES simulation study in time.

Since a simulation study always crosses over functional domain borders the user side team members come from several domains. According to information flow between different functional areas of a manufacturing system presented in Figure 4., potential permanent or temporary team candidates are: capacity planners, schedulers, shop floor managers, marketing and sales managers and engineering function managers [15]. The user domain experts are the suppliers of the information required to build, verify and validate the simulation model. Information collection by group meetings and information collection by interviews is a base for future conceptual framework.

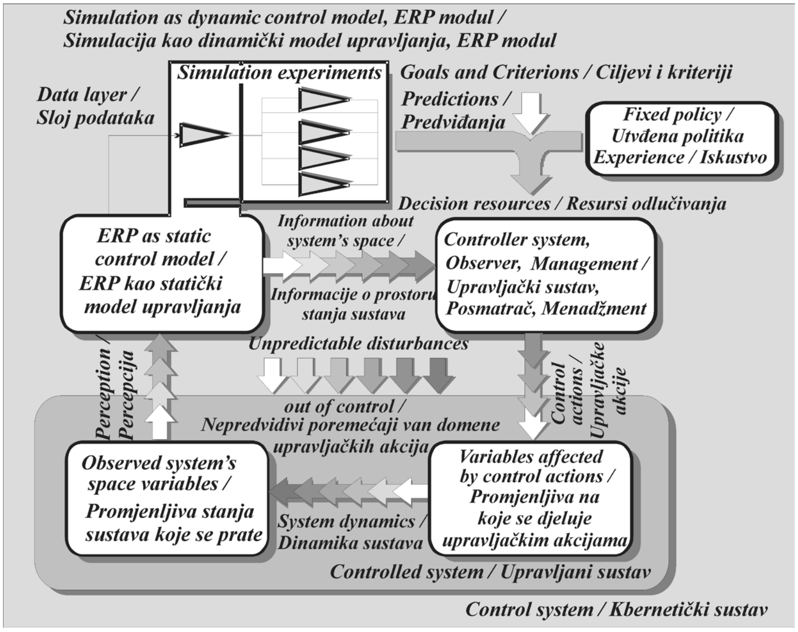


Figure 2. Proposed control scheme

Slika 2. Predložena shema upravljanja

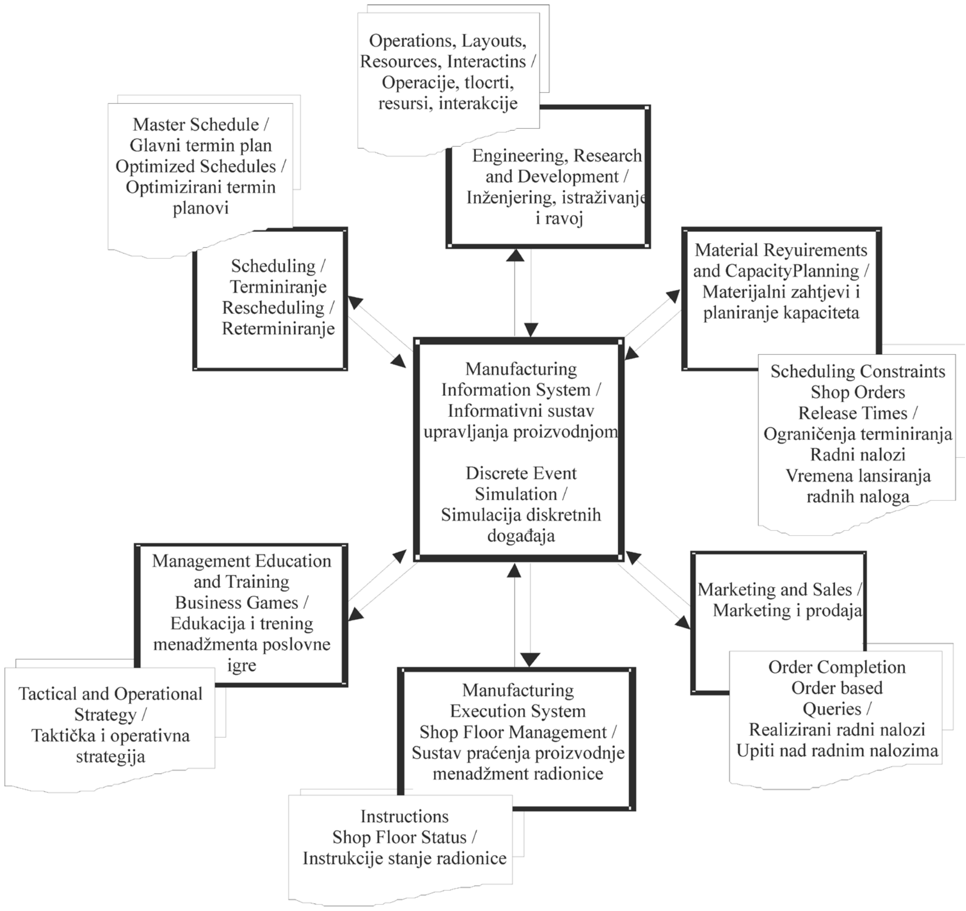


Figure 4. Identification of future DES users

Slika 4. Identifikacija budućih korisnika simulacijskog sustava

At this point, the paper highlights inevitability of knowledge transfer between mentioned knowledge domains (Figure 5.). Integration and overlapping of mentioned specialized fields of knowledge result in multidimensional conceptual framework as a base for domain ontology building. The paper proposes the use of domain ontology as a means of bridging the gap between domain analysis and application system construction. On this basis, concepts defined in the domain ontology can be related to software objects descriptions, acting both as mechanisms for pointing to relevant software components and as specifications of overall configuration requirements.

The paper at this point gives short description of concepts belonging to production scheduling domain. Production scheduling is defined as a process of realistically synchronizing the use of resources by activities to satisfy orders over time. An order is an input

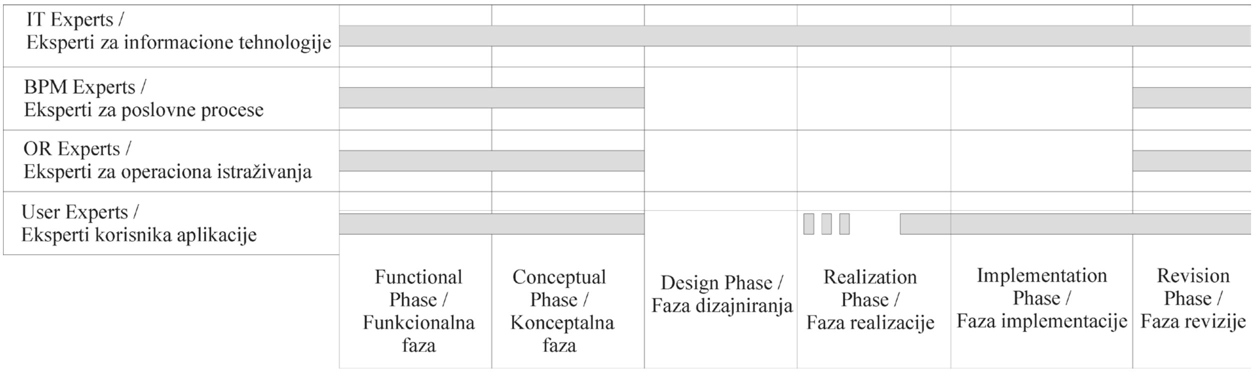


Figure 5. Expert teams' allocation

Slika 5. Alokacija ekspertskih timova request for one or more products, which designate the goods or services required. Satisfaction of orders centers on the execution of activities. An activity is a process that uses resources to produce goods or provide services. The use of resources and the implementation of activities are restricted by a set of constraints. These five elements of basic ontology – order, activity, resource, product and constraint – together with the inter-relationships define an abstract model of a production scheduling domain. In the following text, we analyze the basic ontology concepts in more detail.

An *order* is a specific request for goods and/or services, or more generally products, which the system being modeled is able to produce. Orders specify the input goals that drive the system, along with any constraints that should be taken into consideration when achieving them. An order has several defining properties: *Order_code, Product_code, Quantity, Due_date, Priority, Order_status*.

A *product* is a good realized through the execution of some set of activities. An order for a product is considered satisfied when all of these activities have been completed. The product definition includes the following properties: *Product_code, Routing_code.*

The *routing* indicates the specific set of activities which must be implemented so that a product is produced, as well as their sequence. The routing definition includes the following properties: *Routing_code, Activity_code, Activity_range.*

An *activity* represents a process that can be executed over a certain time interval. An activity requires resources to be executed and its execution both depends on and affects the current state of these resources. The basic decision variables associated with an activity are: *Activity_code, Product_code, Start_time, End_time, Workcenter_code, Activity_range, Activity_status.*

A *resource* is an entity that supports or enables the execution of activities. Resources are generally in finite supply and their availability constrains when and how activities can be implemented. In a manufacturing system, a resource can be a machine (or a work center in general) and/or human resource needed for the proper execution of an activity.

In general, we can distinguish two broad classes of resources from the standpoint of availability:

- Capacitated-resources, whose availability is characterized in terms of the amount of capacity that is available,
- Discrete-state-resources, whose availability is a function of some discrete set of possible state values.

It is stressed that the resources, together with their properties (*machine_capacity, machine_code, technician_code, etc.*) are embodied in the other concepts of basic ontology.

A *work center* is an area in a business in which productive resources are organized to implement a specific work. A work center may be a single machine or a group of similar machines. These work centers can be organized according to functions in a job-shop configuration, or by product in a flow-shop (assembly line). The work center definition includes the following properties: *Workcenter_code, Activity_code, Average_setup_time, Average_capacity.*

A *machine* is considered to execute only one activity every time and it is specified by the scheduling system. The properties of a machine are the following: *Machine_code, Workcenter_code, Machine_capacity, Machine_availability, Machine_flag.*

An activity can be executed by using machines and human resources (personnel) as well. The *human resources* concern both the necessary setup of a machine and the whole execution of an activity. The properties of personnel are the following: *Technician_code, Product_code, Activity_code, Machine_code, Setup_input, Activity_input.*

A *constraint* restricts the set of values that can be assigned to a variable. In the production scheduling domain, constraints restrict the assignment of start and end-times and the allocation of resources to activities. From this perspective, we can identify two basic types: Hard and Soft. Alternatively, soft-constraints are considered to be relaxable, if needed so. For example, due-date constraints are treated as relaxable constraints in many scheduling systems.

Possible restrictions and constraints which may appear in a production scheduling system are the following:

- Release dates. A specific “job” may not start its processing before its release date.
- Preemptions. Preemptions imply that it is not necessary to keep a “job” on a machine until completion. The scheduler may interrupt the processing of a “job” (preempt) at any time and put a different “job” on the machine. The amount of processing a preempted “job” has already received is not lost. When a preempted “job” is put back on the machine (or on another machine), it only needs the machine for its remaining processing time.
- Precedence constraints. Precedence constraints may appear in single machine or in parallel machine environments, requiring that one or more “jobs” may have to be completed before another “job” is allowed to start its processing.
- Permutation. A constraint that may appear in the flow-shop environment is that the queues in front of each machine operate according to the FIFO

discipline. This implies that the order in which the "jobs" go through the first machine is maintained throughout the system.

- Blocking. Blocking is a phenomenon that may occur in flow-shops. If a flow-shop has a limited buffer in between two successive machines, the upstream machine is not allowed to release a completed "job".
- Recirculation. Recirculation may occur in job-shops, when a job may visit a machine more than once.

Conceptual framework or problem domain ontology created in accordance with proposed methodology results in multidimensional description of system's structure and behavior. The next step is design of application architecture of DES solution that makes possible observer's linguistic interaction with its own linguistic states.

3.2. Design phase

The proposed approach uses Extended Petri Nets (EPN) like graphical and mathematical modeling tools for dynamic modeling of obtained conceptual framework of manufacturing system's structure and behavior. This formal, graphical, executable technique is especially convenient for the specification and analysis of concurrent, discrete event dynamic systems.

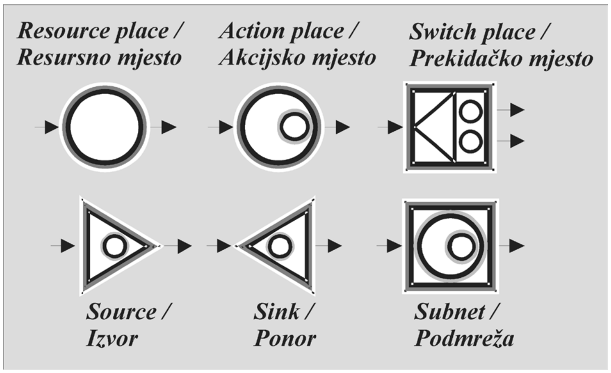


Figure 6. Extended Petri Net Places

Slika 6. Mjesta u proširenoj Petrijevoj mreži

A place denoted by a circle represents a condition such as input data, input signal, resource, condition, or buffer. A transition denoted by a solid bar represents an event such as a computation step, task, or activity. Arcs are utilized to connect places and transitions in a Petri Net. Arcs are directed (depicted by arrows) either drawn from a place to a transition or from a transition to a place. Arcs in a Petri Net have multiplicity. The fourth element called the token and denoted by a solid circle provides the dynamic simulation capabilities to Petri Nets. Tokens are initialized at a place and a place may contain zero or more tokens. With the use of tokens the modeler can provide the necessary dynamic links between the places (conditions) and transitions (tasks or events) in a Petri Net. The concept of transition "firing" allows a Petri Net to simulate the dynamic behavior of a system. In their original form transition firing in Petri Nets was instantaneous, but, time is incorporated into Extended Petri Nets. This results in a timed transition that will have the ability to model tasks or activities.

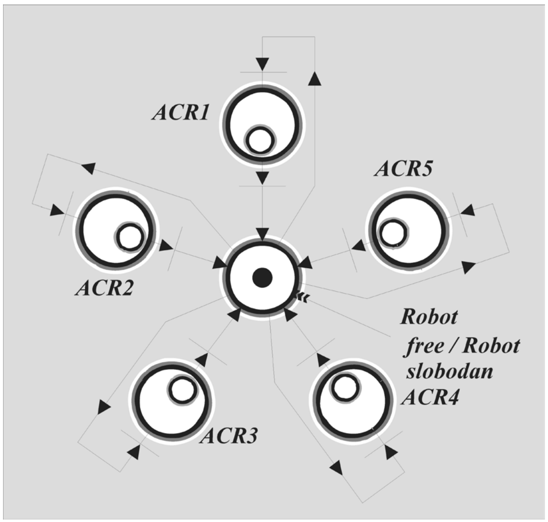


Figure 7. Robot activity cycle

Slika 7. Ciklus aktivnosti robota

The proposed methodology considers the operation of the manufacturing system as a process that is decomposed into operations with specified precedence relations. For each operation the required resources are identified and based on them, the overall system is decomposed into a set of finite subsystems. The operation of each subsystem is modeled as an event-graph representing a single resource activity cycle.

Obtained EPN model describes concurrent, discrete-event system's dynamic, recording in time scale all events happened in the system. This EPN model is crucial part of baseline required for design of the business logic layer.

At this point, IT experts possess complete picture about system being modeled, and initiate software design phase. Developed conceptual framework or domain ontology, used together with mentioned modeling tools is used as a base for an object oriented analysis software development phase. Obtained data easily can be replicated in object oriented, class and object hierarchy of application architecture in software design phase.

Application architecture consists of presentation, business logic and data layers. Presentation layer, assembled of graphical user interface objects is fully consistent with business logic of application. The user interacting with application through user friendly interfaces realizes everyday's business tasks in terms of his own linguistics concepts.

The structure and behavior of the business logic layer is the same as the structure and behavior of modeled system as a consequence of class and object hierarchy realized through steps of proposed methodology. Every object from manufacturing environment has its own counterpart in software environment with both static and dynamic properties modeled. Every communication between objects from manufacturing environment is modeled by functionality of corresponding objects inside object hierarchy of business logic layer. The requirement that Discrete Event Simulation software solution or a control model must be, as much as possible replication of the real system is fulfilled.

Data layer is fully object oriented replication of conceptual data model obtained in analysis phase of software design. This layer communicates with fully object oriented business logic layer and relational database model. This disturbs consistency between the conceptual level of abstraction, obtained in conceptual phase, and business logic and data logic layers of fully object oriented application architecture. Proposed methodology overcomes this impedance mismatch problem between object oriented and relational models using tools for mapping related data in data layer objects with corresponding data in database. In this case, data model, or, Discrete Event Simulation application is relatively independent and, in the same time, fully integrated with Manufacturing Information System databases.

3.3. Review of concrete solutions

The proposed approach is applied in concrete environments of a radiator production manufacturing system and a flexible manufacturing system. The structure of the first system consists of three radiator production lines, five welding machines, six control points, a crane, workers and transport vehicles (Figure 8.). The structure of the second system consists of four work centers, a robot, work in process buffers and, input and output work center buffers and personnel (Figure 9.).

Realized simulation tools have a number of unique characteristics such as interactive Gantt chart display, specialized reports, integration with external data sources, specialized scheduling rules, concurrent graphical animation etc. The quality of the generated schedule is largely determined by the scheduling rules that are specified for selecting resources and operations.

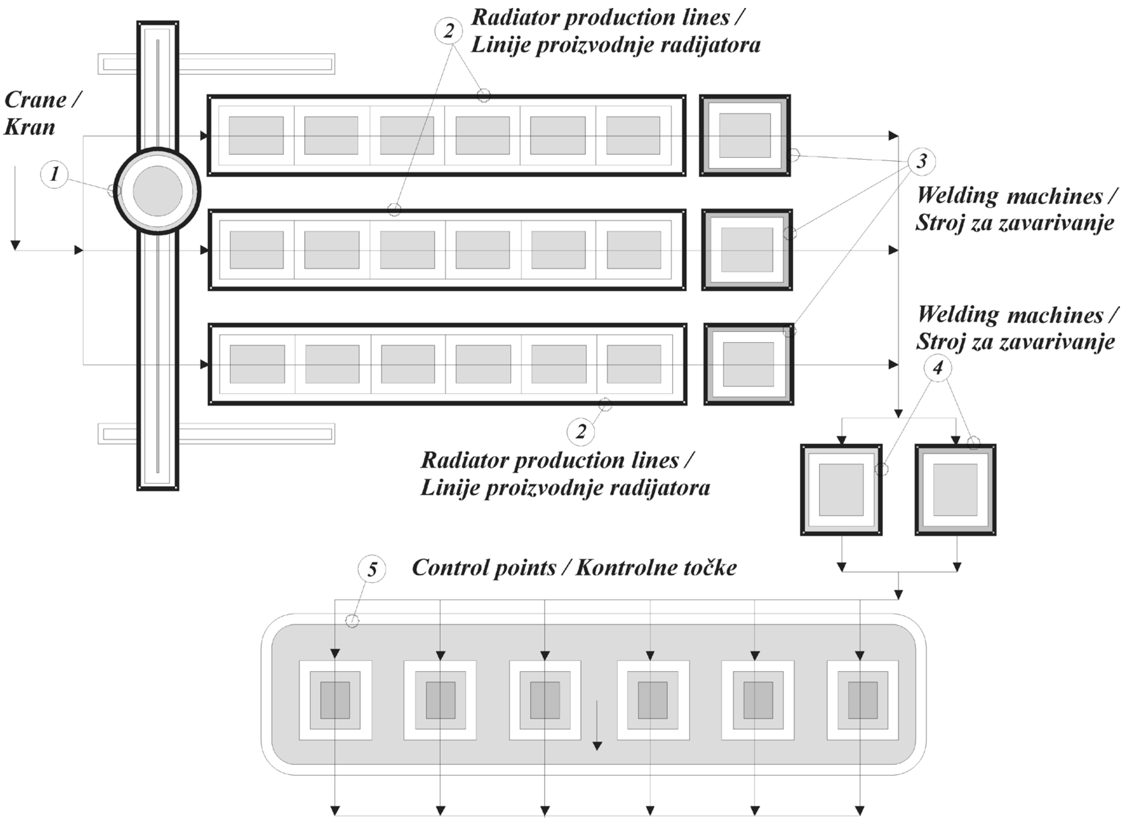


Figure 8. Radiator production layout

Slika 8. Tlocrt radionice za proizvodnju radijatora

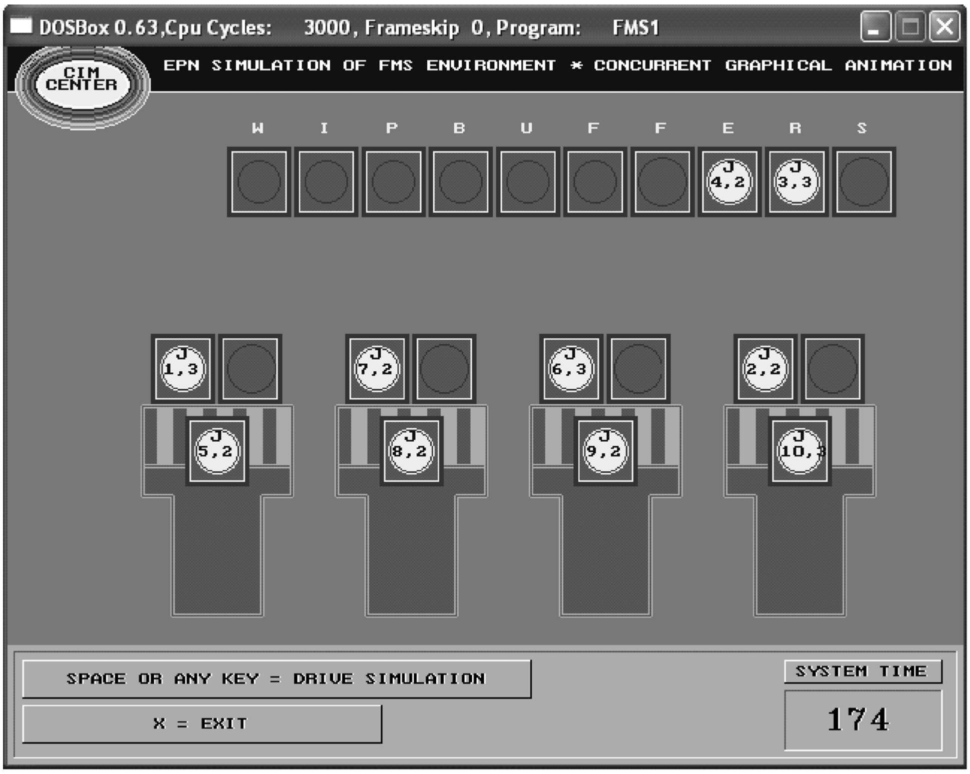


Figure 9. Flexible Manufacturing System layout

Slika 9. Tlocrt fleksibilnog proizvodnog sustava

A complete set of rules must be incorporated into the simulation tool to support a specific range of given manufacturing identity.

Realized simulation software faithfully mimic structure and behavior of modeled systems registering all relevant data generated by simulation experiments. Simulation model gathers in time scale all relevant events for every job order and utilization of machines and other modeled resources in the system. User obtains input, throughput and output time for every job order and calculates stock levels for material and finished product inventory. Simulation experiment covers complete logistic chain connecting system communication with supplier and customer by optimized production schedule in accordance with determined goals and criterions.

Figure 10. represents Extended Petri Net model of structure and behavior of radiator production shop floor. Subnets in EPN model replicate both simulation software modules and activity cycle diagrams of modeled system resources.

Figure 11. and Figure 12. present Gantt charts obtained after first and second simulation experiments for first modeled system. Obtained schedule is optimized and throughput time is decreased for 18 percents for first system and 11 percents for second (Figure 13., left, right). Figure 14. represents increased FMS work centers utilization after schedule optimization.

4. Conclusion

One of the basic management problems today is how to manage a business process and at cost effective manner adapt to unavoidable change dictated by environment. The most universal challenge in this environment is handling supply, operational and demand exceptions what leads to the apparently impossible mission of planning for the unplanned. The speed at which business system identify these exceptions and react in order to reduce its negative impact on overall system performance primarily depends on responsiveness of Management Information System and its capability of handling the above three factors [16]. The contemporary solution is integration of strategic and tactical decision making and, on production level, developing the capability for synchronization, modification and fine-tuning of production plans and schedules as quick as possible. Instrument of mentioned integration is Discrete Event Simulation scheduling solution integrated with Management Information System. Simulation's scheduling role in such a system is key due in large part to its ability to faithfully replicate the real production ambiance and quickly react on unpredictable exceptions in the field.

This causes problem in designing corresponding simulation software models which will replicate complexity, enable evolution control of modeled system as well as synchronization of its own growth with named evolution? The simulation model designed by

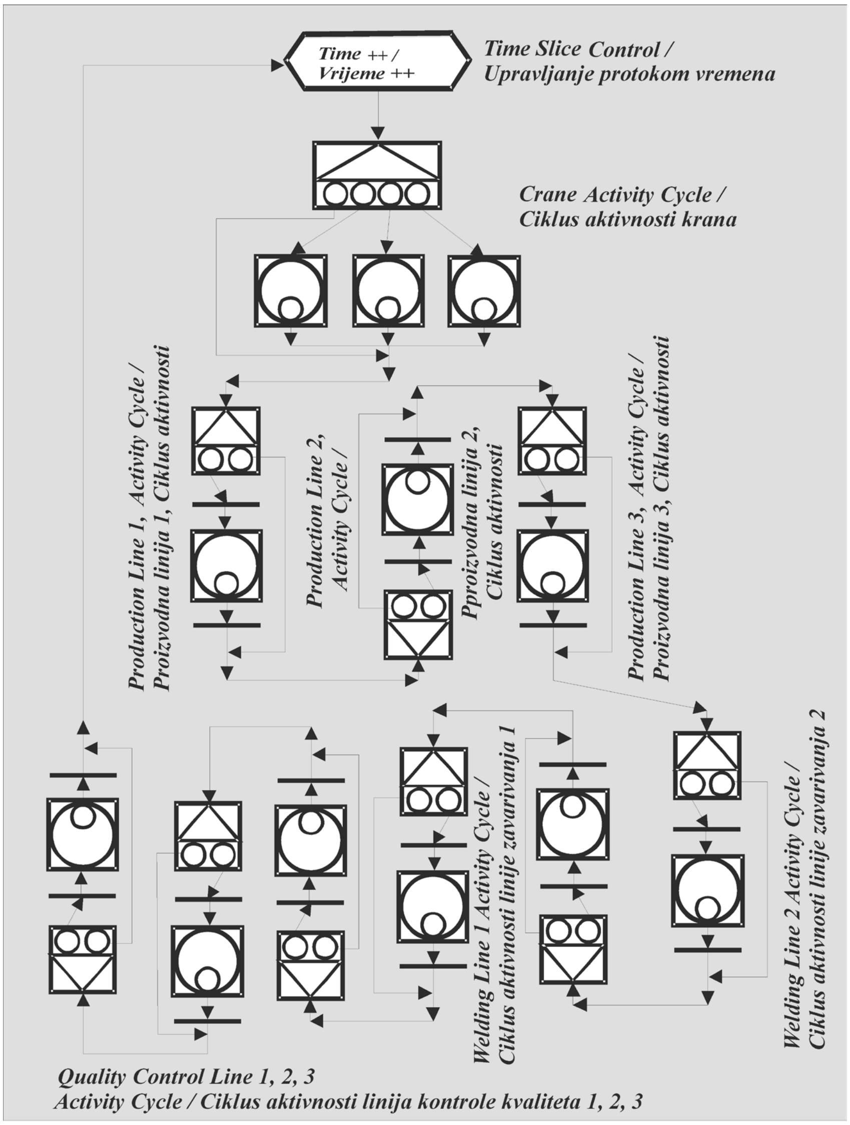


Figure 10. Extended Petri Net model of radiator production shop floor

Slika 10. Model proširenih Petrijevih mreža radionice za proizvodnju radijatora

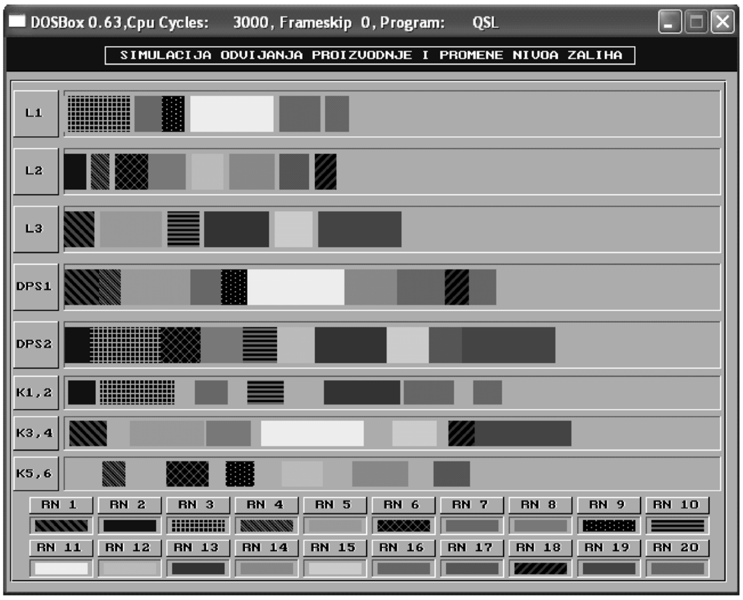
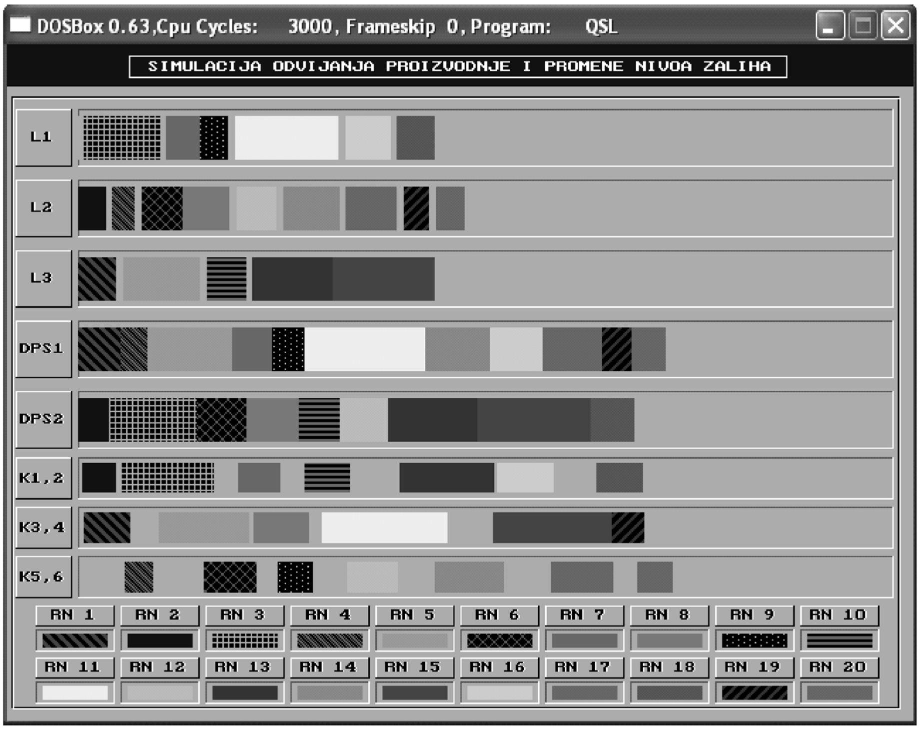


Figure 11. FMS Gantt chart obtained after first simulation experiment

Slika 11. Gantt dijagram nakon prvog simulacijskog eksperimenta za fleksibilni proizvodni sustav



DOSBox 0.63,Cpu Cycles: 3000, Frameskip 0, Program: QSL

SIMULACIJA ODVIJANJA PROIZVODNJE I PROMENE NIVOA ZALIHA

Figure 12. FMS Gantt chart obtained after second simulation experiment

Figure 12. Gantt dijagram nakon drugog simulacijskog eksperimenta za fleksibilni proizvodni sustav

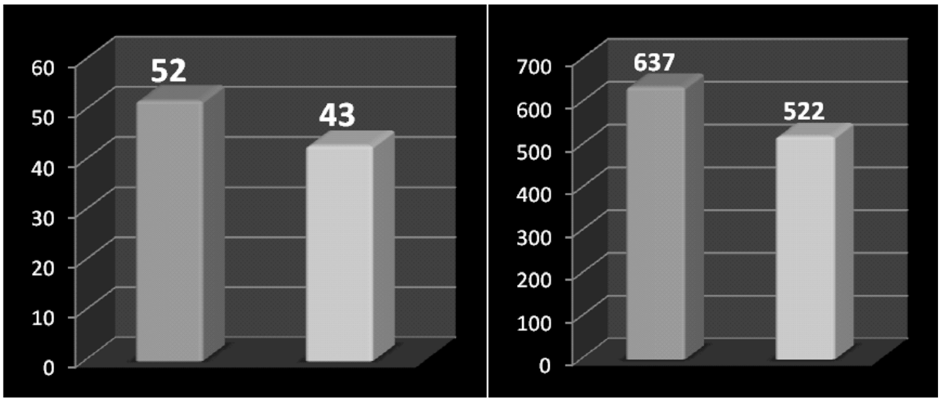


Figure 13. Reduced throughput times for modeled system

Slika 13. Reducirano vrijeme protoka dijelova za modelirane sisteme

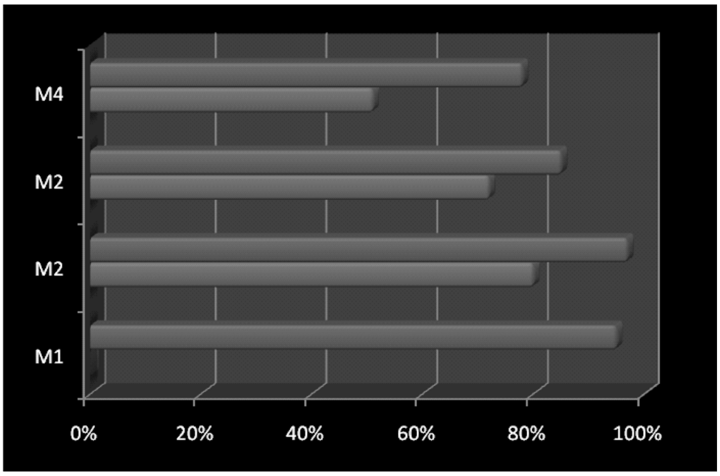


Figure 14. Increased work centers utilization

Slika 14. Uvjećanje iskoristivosti radnih centara methodology proposed in the paper attempts to give an answer to this challenge: integration of strategic and tactical decision making, developing the capability for plans and schedules reconfiguration and synchronization in a very short cycle as well as handling supply, operational and demand exceptions.

This approach gives verbal and mathematical problem description, builds ontology of problem domain, uses Extended Petri Nets and event graphs as activity cycle diagrams as modeling tools in order to obtain faithful model which easily can be replicated in object oriented class and object hierarchy. Proposed approach also gives emphasis to unavoidability of knowledge transfer between business processes experts, software development experts, academic community research groups as well as future DES users. Integration and overlapping of mentioned fields of knowledge result in object oriented application architecture fully consistent with ontology derived from conceptual phase of design methodology. Proposed methodology enables convergence of comprehensive but static MIS knowledge in dynamic simulation model in order to fully utilize its prediction power for effective integration of strategic and tactical decision making.

REFERENCES

[1] BANKS, J.; CARSON, J.; NELSON, B.; NICOL, D.; *Discrete-Event System Simulation,* Prentice Hall, 3rd ed., 2001

[2] MUSSELMAN, K.J.; UZSOY, R.; *Advanced planning and scheduling for manufacturing*, Handbook of Industrial Engineering (3rd ed.). 2001, New York.

[3] SMITH, S. F.; BECKER, M. A.; *An Ontology for Constructing Scheduling Systems*, Proceeding of AAAI-97, Spring Symposium on Ontological Engineering, 1997.

[4] METAXIOTIS, K.; PSARRAS, J.; ASKOUNIS, D.; *Building ontologies for production scheduling systems: towards a unified methodology***,** Information Management & Computer Security, Volume 9, Number 1, 2001 pp. 44-51

[5] VALAVANIS, K.*; On the Hierarchical Modeling Analysis and Simulation of Flexible Manufacturing Systems with Extended Petri Nets*, IEE Transactions on Systems, Man and Cybernetics, N^o 2,1990.

[6] CARRIE, A.; *Simulation of Manufacturing Systems*, John Wiley & Sons, 1989

[7] ANGLANI, A.; GRIECO, A.; PACELLA, M.; AND TOLIO, T.; *Object oriented modeling and simulation of flexible manufacturing systems: A rule-based procedure*, Simulation Modeling Practice and Theory 10, 2002, pp. 209-234

[8] ARSOVSKI S.; ARSOVSKI Z.; MIROVIC Z.; *The Integrating Role of Simulation in Modern Manufacturing Planning and Scheduling,* Strojniski vestnik - Journal of Mechanical Engineering 55 (2009) ‘98, 33-44,

[9] ARSOVSKI Z.; ARSOVSKI S.; MIROVIC Z.; *Integration of Just In Time and MRPII concepts by simulation software,* INFOFEST ‘98, Budva, 1998,

[10] PARK, H.; KIM, Y.; PARK, S.; *An object oriented production planning system development in an ERP environment*, Computers in Industrial Engineering, Vol. 35 No.1-2, 1999, pp.157-60.

[11] KUNNATHUR, A.; SUNDARARAGHAVAN, P.S.; SAMPATH, S; *Dynamic rescheduling using a simulation-based expert system,* Journal of Manufacturing Technology Management Volume 15 Number 2, 2004, pp. 199-212.

[12] JEONG, K.; *A conceptual frame for development of optimized simulation-based scheduling systems,* Expert Systems with Applications, Vol. 18 No.4, 2000, pp.299-306.

[13] ROSEMANN, M.; *Structuring and modeling knowledge in the context of enterprise resource planning*, 4th Pacific Asia Conference on Information Systems – PACIS, Hong-Kong, 2000.

[14] PINEDO, M.; SAMROENGRAJA, R.; YEN, P.C.; *Design issues with regard to scheduling systems in manufacturing*, Control and Dynamic Systems, Vol. 60, 1994, pp.203-38.

[15] MAHDAVINEJAD, R.; *A New Approach to Job Shop-Scheduling Problem*, International Journal for Quality Research, Volume 1, Number 1, 2007.

[16] ARSOVSKI, Z.; ARSOVSKI, S.; *The Quality Metrics of Information Systems,* International Journal for Quality Research, Volume 2, Number 2, 2008.

[17] ARSOVSKI, Z.; PAVLOVIC A.; ARSOVSKI A.; MIROVIC Z: *Improving the Quality of Maintenance Processes by Using of Information technology,* Strojniski vestnik – Journal of Mechanical Engineering, Vol. 55, No. 11, 2009.