Int J Adv Manuf Technol (2011) 53:385–398
DOI 10.1007/s00170-010-2822-0

ORIGINAL ARTICLE

Information requirements analysis for holonic manufacturing systems in a virtual environment

Hamed Farahani Manesh · Dirk Schaefer · Majid Hashemipour

Received: 16 February 2010 / Accepted: 28 June 2010 / Published online: 13 July 2010

Abstract The design and development of holonic manufacturing systems requires careful, and sometimes risky, decision making to ensure that they will successfully satisfy the demands of an ever-changing market. In this paper, the authors propose a methodology for a holonic manufacturing systems requirement analysis that is based on a virtual reality approach and aimed at assisting designers of such systems along the entire systems design and development process. Exploiting virtual reality helps the user collect valid information quickly and in a correct form by putting the user and the information support elements in direct relation with the operation of the system in a more realistic environment. A prototype software system tool is designed to realise the features outlined in each phase of the methodology. A virtual manufacturing environment for matching the physical and the information model domains is utilised to delineate the information system requirements of holonic manufacturing systems implementation. A set of rules and a knowledge base is appended to the virtual environment to remove any inconsistency that could arise between the material and the information flows during the requirement analysis.

Keywords Information requirements analysis · Holonic manufacturing systems · Virtual reality · Virtual factory

H. Farahani Manesh (✉) · M. Hashemipour
Department of Mechanical Engineering,
Eastern Mediterranean University,
Gazimagusa, Via Mersin 10, Turkey
e-mail: hamed.farahani@emu.edu.tr

D. Schaefer
The George W. Woodruff School of Mechanical Engineering,
Georgia Institute of Technology,
210 Technology Circle,
Savannah, GA 31407, USA

1 Introduction

In recent years the manufacturing industry has been strongly impacted by globalisation, which has resulted in increased global competition. This manufacturing competition has acted as a driving force for employing the latest manufacturing systems that have better capabilities to help enterprises increase their agility. Examples of such manufacturing systems include today's so-called flexible manufacturing systems as well as many predecessors such as computer integrated manufacturing implementations dating back to the 1980s. A major disadvantage of many traditional manufacturing systems was that they suffered from limitations in terms of scalability, robustness and re-configurability due to their traditionally centralised control architectures [1].

In order to better address and meet the requirements of agile manufacturing, research in the area of manufacturing systems control has shifted from traditional centralised approaches to the development of distributed control architectures that range from hierarchical structures to non-hierarchical or 'heterarchical' structures [1]. In this regard, the concept of the so-called holonic manufacturing system (HMS) has been suggested as an intelligent manufacturing paradigm aimed at organising and controlling manufacturing-related activities with greater agility, scalability and fault tolerance [2, 3]. In industrial production environments, the main advantages of holonic manufacturing systems over traditional ones are the distributed computational operations as opposed to a centralised control algorithm, distributed decision making capabilities, as well as an autonomous cooperation between intelligent system entities. This is a key characteristic that does not exist in traditional manufacturing control systems [4].

Over the past decade a number of mainly empirical activities have been going on in the arena of HMS development [5]. However, there still is no standard approach for companies to develop their holonic manufacturing systems as they often have unique and proprietary operational strategies and practices [5]. This fact implies the key importance of a company specific requirement analysis, which is precise and accurate, as the first stage of any HMS development process. Simply put, a requirement analysis (RA) is the process of the discovery, refinement, model building and specification of the information within the problem domain. The importance of requirements analysis has been emphasised by a number of authors as an essential step leading to the successful implementation of manufacturing systems [5–8]. It is stated that acquiring valid specification of system requirements has an increased influence on productivity output of operations and manufacturing performance [7, 8].

The emphasis on the holonic manufacturing system requirements analysis has led to the development of a number of methodologies and methods based on software engineering principles. More recent literature refers to ANEMONA which employs the multi-agent system methodology for HMS modelling requirements [9].

From the research side, several information modelling methods have been utilised for HMS information modelling. Blanc et al. [10] exploited unified modelling language (UML) for modelling static and dynamic behaviour of the system. Moreover, Giret and Botti [5] extended INGENIAS models (a multi-agent systems (MAS) methodology for complex systems) to model and analyse the HMS requirements.

Structured analysis diagrams have been the principal tool used in the aforementioned methods for requirements modelling and analysing. In addition, these methods do not provide a means to validate the models, which is a desirable way to identify errors and problems in the early stage of system modelling [11].

Wand and Weber [12] have indicated that the high-quality representation, both static and dynamic phenomena of the system, can facilitate early detection and correction of the system development errors. According to Gerstenfeld and Roberts [13], 70% of small- and medium-sized enterprises used some form of process flow charting in support of the design process, and only 20% used a system design suggested in the literature or by the consulting firms. It has been stated that extensive and heavy documentation, lack of a communication between the user and the system analyst and loss of interest by management has been major drawbacks in preventing an effective realisation of the system [14].

Simulation and animation have emerged as effective means for relieving such problems by helping the system developer in better understanding and verifying the behaviour of a desired system. Model animation is not always user-friendly. The standard technique of model animation is based on highlighting the current state of the model, which in turn is represented by the notations of the supported modelling languages. Nontechnical users, who are not familiar with systems engineering modelling languages, might experience difficulty in reading and interpreting such animations. Surveys show that the simulation capabilities of commercial tools have similar characteristics [15, 16]. They support users in understanding the process characteristics but often suffer from limitations in the detailed analysis necessary for accurate system adoption and in capturing visual details of implementation. They simply fall short in examining requirements that are specific to the equipment and physical configuration.

Virtual reality (VR) can be considered a solution for such problems, as the user and the information support elements are put in direct relation with the operation of the system in a realistic environment. VR, a synthetic environment providing a sense of reality and an impression of 'being there', has been increasingly employed in various applications of design and manufacturing. Such applications include computer-aided design (CAD), tele-robotics, assembly planning, manufacturing system visualisation and simulation [17, 18].

Over the last decade VR has also been utilised for modelling specific types of manufacturing environments, which has led to what today is known as virtual factory. A virtual factory can be defined as a computer-based simulation of a real manufacturing system that provides users with valid results on manufacturing system-related operations [19]. The virtual factory approach is widely accepted as a useful tool for facilitating user's access and understanding of manufacturing processes and operations [18]. The use of VR for simulating manufacturing systems gives engineers and/or trainers an opportunity to play a pro-active role in identifying flaws and optimising aspects of manufacturing processes and activities [20]. In addition, simulation and visualisation environments are often advocated as a cost-effective means of improving manufacturing productivity by defining and validating the most appropriate configuration and architecture for any given manufacturing system without really implementing it into the current manufacturing operations [21, 22].

Several commercial software packages have been developed for VR in manufacturing applications (e.g. VisFactoryTMDELMIATM). The VisFactory tool provided by Engineering Animation Inc. consists of modules for designing, analysing, comparing and improving factory layouts with respect to material flow. Through the VisFactory tool, alternative facility designs are generated by entering relevant data and related information such as flow distances, handling cost and adjacency scores. The DELMIA package, released by Dassault Systems, provides authoring applications that include platforms to develop and create a virtual manufacturing environment to address process planning, cost estimation, factory layout, ergonomics, robotics, machining, inspection, factory simulation and production management. The overall goal is to support the roadmap towards the data integrated 'digital manufacturing enterprise'. However, packages such as DELMIA and VisFactory, which support VR-based simulations, do not fully cover applications regarding the information modelling for requirement analysis.

In this context, information modelling tools enhanced with VR, which is the main scope of the current paper, are potentially useful for supporting the user in planning and visualising the manufacturing system. The integration of VR into information modelling helps to improve information requirements analysis for HMS implementations by avoiding some of the shortcomings of the commercial modelling and simulation tools as stated above.

In this paper, a VR-based modelling methodology for the requirement analysis stage of HMS development is proposed. A desktop VR system allows for users and information support elements to be in direct relation with the operation of the system in a more realistic environment. The prospective user not only gets a verbal or diagrammatic description but also gains the capability to practice familiar tasks in a carefully imitated operating environment. Using VR in RA can be considered an interactive simulation environment conveying the data in a form the user is accustomed to treating; while it is responding to the specific habits of the user. This methodology makes use of VR for requirements analysis in order to answer the following questions:

1. How to define an innovative method to understand complex systems such as HMS?
2. How to enhance the understanding of the system and communication consistency between system analyst and user?

For realising the features outlined in each phase of the methodology, a VR-HMRA prototype tool is designed and developed as a software system and an illustrative example using a firm in the die-casting industry is presented. A significant aspect of the research presented here is to use innovative methods for HMS development by introducing VR in order to reinforce the conventional methods in requirement analysis. Another aim is to maximise the degree of agility and produce a precise and accurate requirement analysis for each phase of the development process that may finally lead to an increased performance of the plant.

2 Overall methodology

The proposed methodology emphasises the issue of information integration as the key factor in the design and implementation of desired HMS goals. A typical manufacturing system is conceptualised as composed of two complementary domains: the physical system domain and the information system domain. The physical system is concerned with the physical structure and flow of materials within the enterprise. The information system covers the functions ranging from the highest level of strategic planning of the business activities to the lowest level of the manufacturing cell level. However, in this paper our focus is on the manufacturing-related tasks alone. As shown in Fig. 1, the proposed methodology comprises the following phases: (I) conceptualization phase, (II) implementation phase and (III) analysis phase, including the specifying of the HMS information requirements. The conceptualisation phase is intended to construct the initial models. In the implementation phase, the to-be modelling is constructed. Finally, in the last phase, HMS information requirements will be analysed and specified.

A potential surrogate approach for the development of holonic manufacturing systems is that of multi-agent systems. Multi-agent systems are well understood and provides clear and unambiguous analysis and design guidelines for HMS modelling [5]. The MAS paradigm possesses special features for modelling intelligent systems, including autonomy, rationality, reactivity, pro-activeness, adaptability, mobility and benevolence, which enables the realisation of complex systems with intelligent behaviour. Due to similarities between holonic and agent-based concepts and the fact that complete multi-agent system methodologies are readily available, the development and implementation of HMSs is often done based on MAS technology [5].

Many researchers have pointed out that holons and agents are very similar concepts [23]. However, some extensions must be considered in the MAS methodology to model the HMS requirements in an appropriate way. These are: "holon recursive structure, system abstraction levels and a mixed top-down and bottom-up approach in design and analysis process" [23]. Moreover, the availability of complete MAS methodologies makes such technology suitable for HMS modelling and implementation, in such a way that, the holons and their interactions are modelled and simulated throughout the agents [5, 10, 24]. Motivated by above consideration, MAS approach is adopted for modelling HMS for both physical and informational system domain in the presented methodology. Consequently, henceforth in this document the term agent will be referred to as holon.

2.1 Information system domain development

The prerequisite for performing a thorough system requirement specification is the development of a model of the system. The model is a description of manufacturing-related

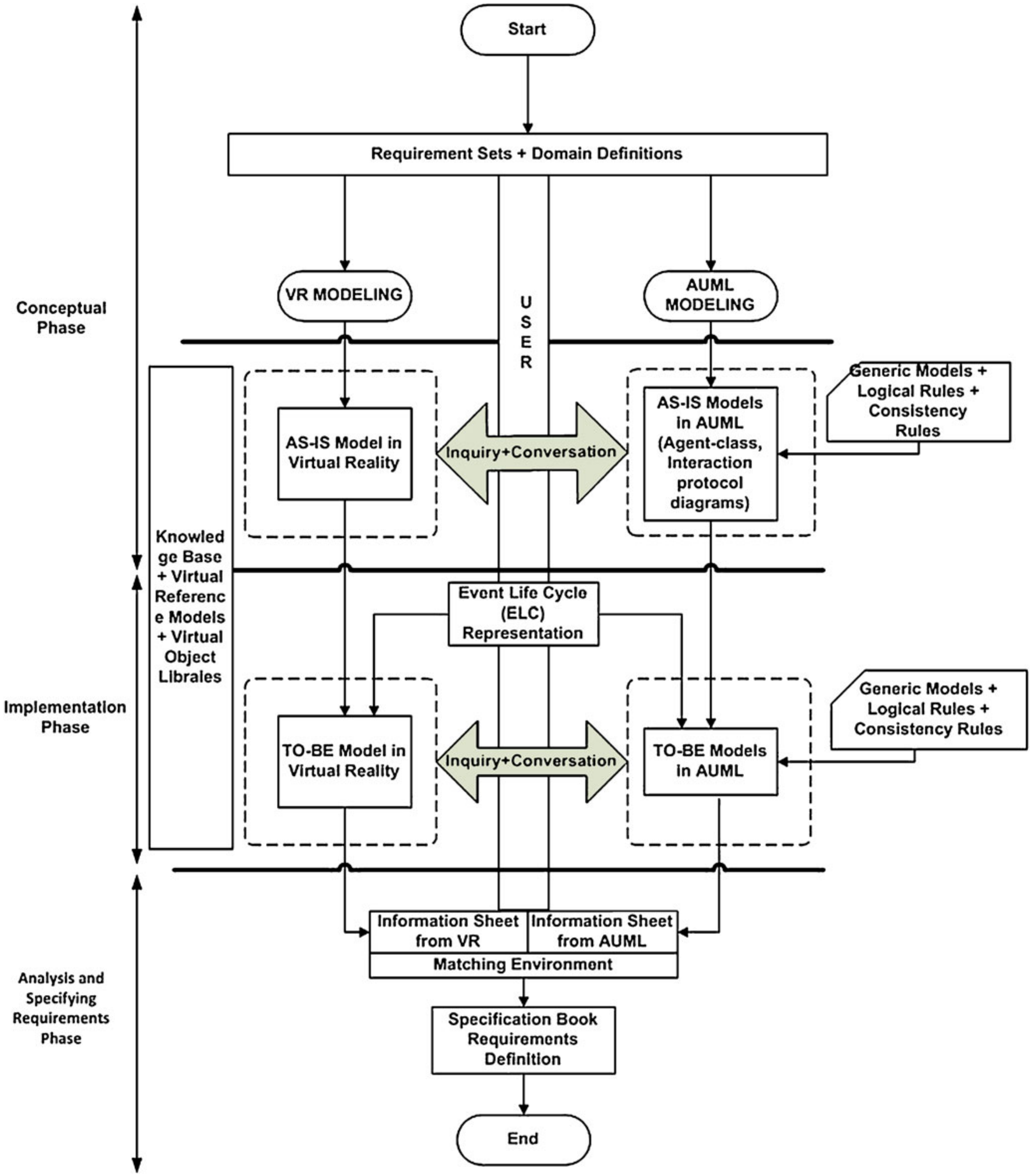

Fig. 1 The overall methodology

tasks in the factory and corresponding activities. A well-known and widely used adopted notion for designing, specifying, visualising, constructing and documenting artefacts of a MAS is the agent unified modelling language (AUML) [25, 26].

Agent UML (AUML) is a graphical modelling language that is being standardised by the Foundation for Intelligent Physical Agents Modelling Technical Committee (Modelling TC) [27] and has been proposed as an extension of UML. The most important advantages of choosing AUML for

MAS modelling are that different diagram types allow different aspects or viewpoints of a system to be modelled (functional, static and dynamic view), and that there is extensive experiences in applying AUML to real-world systems design [28–30]. In addition, AUML is widely used in industrial software projects and users who are familiar with UML will not have any difficulties understanding understand AUML. Furthermore, UML 2.2 [31] already contains many of the agent-related features that AUML requires. We adhere to the notation standards of AUML in the agent-oriented analysis. The agent-oriented framework of the methodology for development of information system domain is realised in four stages: (modelling diagrams from a die-casting factory case study is used to illustrate the stages and due to space constraint, diagrams are represented at a high level of abstractions):

Stage 1: At this stage the primary requirements are elicited in order to define the requirements set and problem domain. The main goal of this stage is to decompose the system in order to identify the initial agents. Some guidelines have been defined to help analysts with requirements elicitation. These are:

- Pre-conditions
- Main flow of events
- Alternative flows
- Exceptional flows
- Post-conditions

Stage 2: Developing an agent-class diagram based on the elicited information of the previous stage (Fig. 2), which represents autonomy and goal-driven execution of each agent [32]. This stage provides a template for the agent's overall structure. Figure 3 shows a sample of an agent-class. The following steps are used for completing this stage:

a. *Group and role assignment*

Each agent is assigned a group/groups and delineated different roles to plays. This step provides a template for the agent's overall structure. Figure 3 shows a sample of an agent-class definition that was used in this study.

a. *Service description*

Different services that an agent—groups, roles or the agent itself—can provide are described. These services are supported by related protocols to facilitate information exchange.

b. *Events*

In this step, all the events, which an agent participates in within its environment, are compiled.

c. *Knowledge*

Knowledge corresponds to the general description of the agent.

d. *Protocol description*

Protocols facilitate information exchange. Protocols can be extracted from the service definition and roles.

Stage 3: Designing interaction protocol diagrams, which represent valid sequences of messages between agents. Interaction protocol diagrams contain the participant's interactions along with message sequences [32]. Figure 3 illustrates three participants from the die-casting factory example.

Stage 4: Designing activity diagrams to define and integrate goals, plans and actions within the agent shell and to provide a record of events following the order in which they occurred [32]. Action is the primary unit in a goal, which corresponds to specific task that the agent must carry out to achieve its objectives (see Fig. 5). Agent-class, interaction protocol and activity diagrams cover all relevant system views.

2.2 Physical system domain development

The physical system domain development focuses on building a "virtual factory" in a MAS platform. The basic agents in the modelled factory are to be defined first, before designing and integrating the basic components. Next, the events agents are described. The event description is the characterization of tasks by agents in VR under specific rules and conditions. All the operations have their own visualisation in the VR environment. These operations are defined by appropriate motion elements. Behaviour description constitutes the simulation of the agents in a VR environment given the assigned operations. For instance, when the die machine receives the *proceed order()* command, the assigned operations producing part on this case, should be visualised. The molten metal should be seen poured into the die machine and then gradually injected into the mould and eventually unloading the produced part.

In the context of utilising virtual reality for holonic manufacturing systems implementation, Bal and Hashemipour [33] present a virtual reality-based methodology for enhancing the implementation process of holonic control systems in manufacturing practice. They introduce a virtual holonic factory, which integrates VR-based modelling with an agent-based platform in order to realise a holonic manufacturing control system within a discrete event simulation environment.

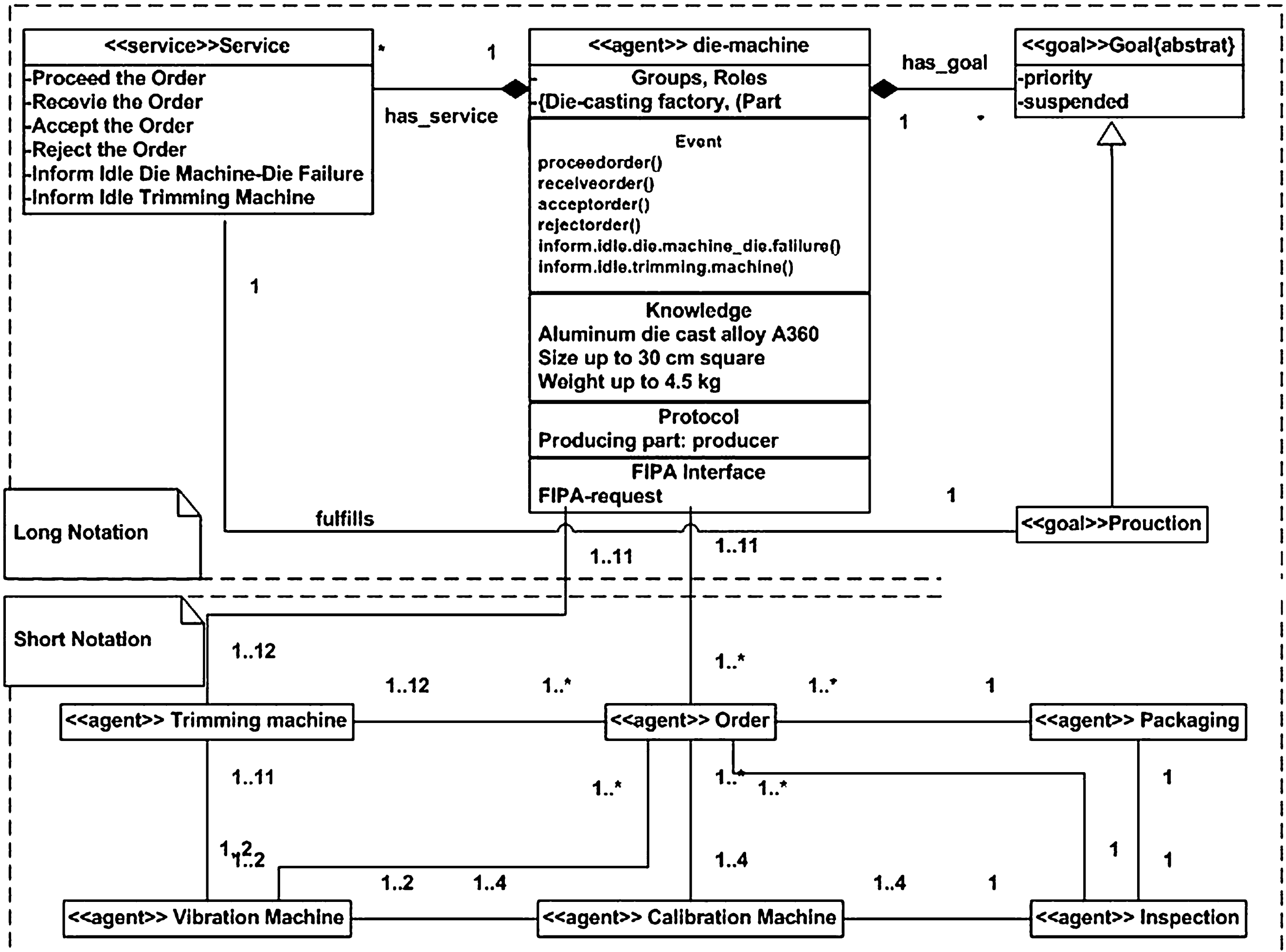

Fig. 2 Agent-class diagram of the die-casting factory

2.3 Conceptual phase: building the initial models

The methodology is based on the PROSA architecture [34] for HMS development. Therefore, a generic requirements model is necessary and helpful in the first step. This helps by decomposing the system in order to identify the basic holons and to provide an initial holon and holarchy specification from the physical and informational point of views of PROSA. The physical view defines the resource holons. For instance, the manufacturing equipment comprising the machines material handling equipment such as Robots, automated guided vehicles (AGVs), forklifts, etc, are regarded as the basic resource holons. The information view supports the development of the information model of the domain analysed at the requirements definition stage. This involves the flow of the information through the enterprise. The following example exemplifies this information: *product holon* contains the process and product information to assure the correct making of the product with sufficient quality; *order holon* holds all the knowledge related to the all the tasks in the manufacturing system; *staff holon* provides assistance to the other holons in performing their tasks.

As-is modelling includes two main activities, namely the creation of the as-is virtual reality model and as-is information model. The as-is VR model consists of a dynamic VR model of the manufacturing system under analysis. The VR model is constructed in a way to represent the current physical configuration and operations within the factory. The VR model can be constructed either by selecting a reference model from the VR factory model or by using virtual object libraries, which provide virtual objects of the manufacturing shop floor elements. Virtual objects can be dragged into the virtual environment from the virtual object libraries. The virtual objects can also be assigned attributes, list of events that can be requested, conditions and the ways each object reacts. The virtual reference models are previously created generic shop floors. A number of virtual factories are prepared as a virtual reference model and a knowledge base to help the user

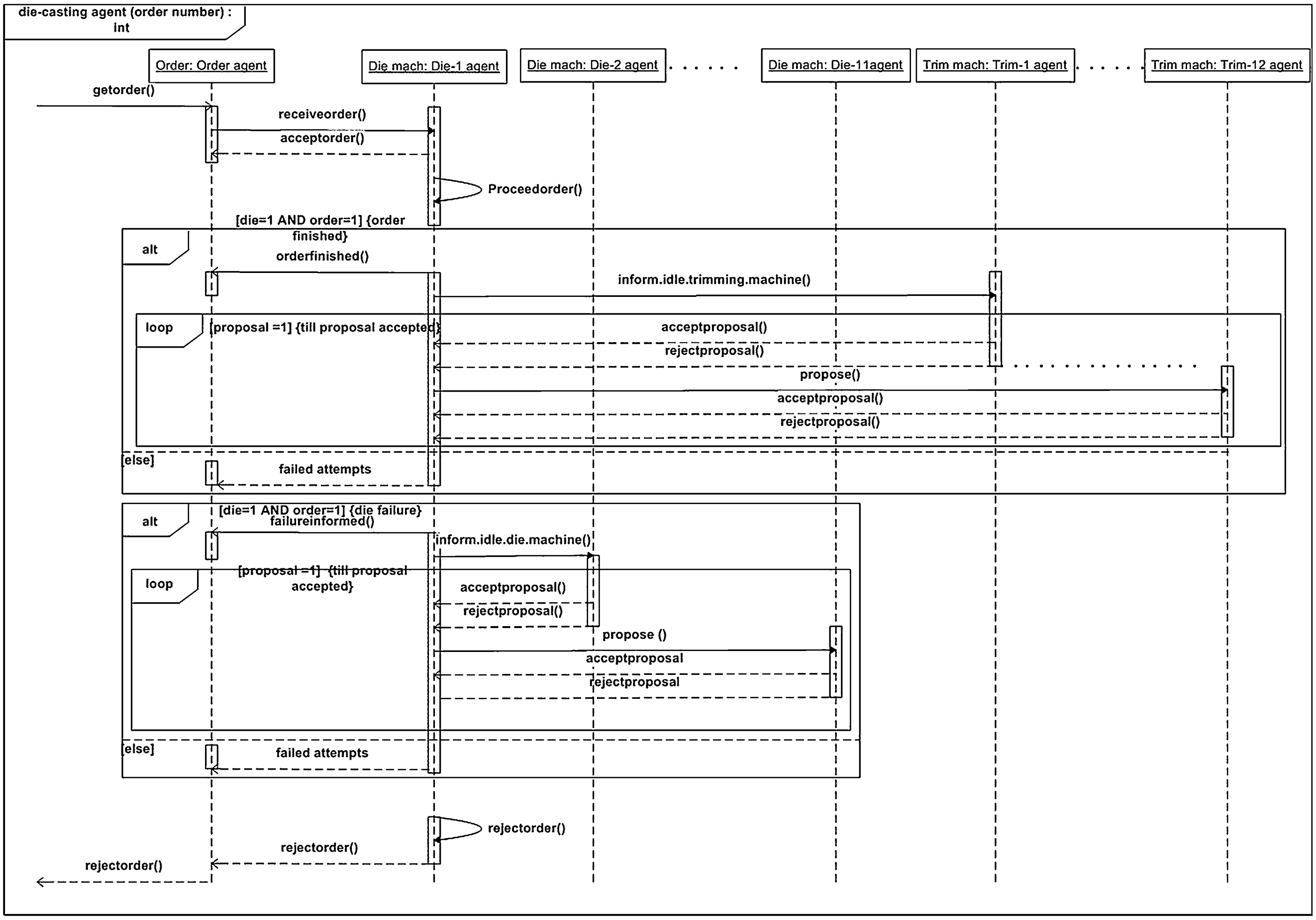

Fig. 3 Interaction protocol diagram of the die-casting factory

select the related virtual factory as a virtual reference model. A similar approach was adopted by Xu et al. [35] for the construction of a virtual environment using a reference model and a knowledge base.

The selection of the reference VR model of the factory is carried out through a multi-dimensional selection scheme introduced by McCarthy and Fernandes [36]. The scheme is implemented in a way that the user defines the proper specifications of the manufacturing system according to elicited information regarding pre-requirements of the manufacturing system. Once the reference model is selected, candidate objects can be used in constructing the as-is model. The creation of the shop floor with necessary details in the VR environment is the result of the as-is VR modelling.

Figure 4 shows the construction process of the as-is model. Analysis starts with the communication of the analyst and the user, and then the analyst gets help from the knowledge base system to select a proper generic model. Once the model is selected the analyst refers to the virtual object libraries to complete the details. Analysis continues until the user and the analyst agree on a satisfactory representation.

The next process is to construct the reference information model using AUML, getting help from a typical operation in the virtual environment. This is performed with the active involvement of the user. The process repeats until the user and the analyst agree on a satisfactory representation. As-is AUML models represent a situation as it exists in practice. That is, the interrelation between activities, information and facilities that make up the system. An as-is AUML model can be created from the VR environment. The virtual setup of the factory under analysis is translated onto an AUML model. A set of the AUML reference model archives (agent-class, interaction protocol, activity diagrams) enables quick progress at this stage of the process. AUML models established through the VR representation are the outcome of interactive communication between the analyst and the user over the VR objects.

Two sets of rules are used for the consistency checks of the AUML diagrams [37]. The first set is referred to by the name of the message: “Name of the message in interaction protocol diagrams must match an event in receiver agent-class”. The second set is related to the direction of the message: “Calling direction of the message must match an association”.

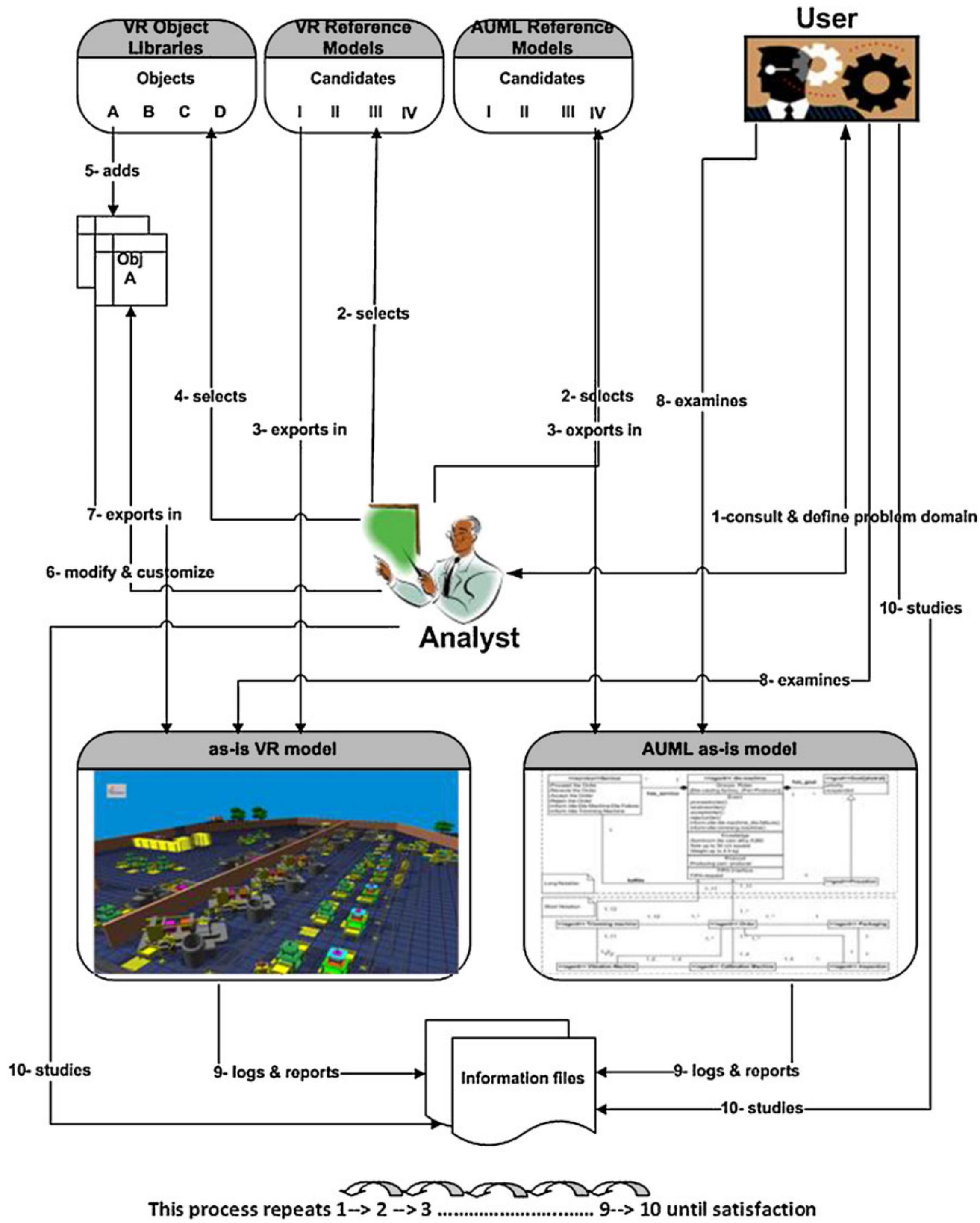

Fig. 4 Construction process of the as-is model

2.4 Implementation phase: to-be model construction

The second phase of this methodology comprises the "to-be" model of the enterprise, which describes the prospective information system and its operations. This section is concerned with the event life cycles (ELCs): the comparative modification in the matching environment.

The ELCs provide chronologically valid models of the world. They represent processes and event they are involved in. AUML activity diagrams are adopted for ELCs representation. Contrary to agent-class and interaction protocol diagrams, activity diagrams systemize the life cycle of the events and processes. Hence, the behavioural descriptions of the agent-oriented framework are expressed in a wider scope to describe the entire system.

Events in ELCs are instances of the agents and actions of the agents. The definition of an event is 'something of interest in the problem domain', or 'a thing about which information is held'. Figure 5 shows a sample activity diagram as an ELC representation of the die-casting factory that was created in the VR environment, from order received by die machine agent until order delivered. If there is a condition to accompany initiation of an order, it is also to be specified. For example, the arrival of a semi-finished part precedes this initiation of process by the die machine.

The ELC is designed with necessary details, conditions and related rules, in such a way that the activation of data generated from the diagram modifies the data files of the VR model and AUML diagrams. Hence all the occurrences defined in the activity diagram can be followed. The requirements analysis system operates under the control of the VR and AUML generated data files.

2.5 Third phase: analysing and specifying information requirements

The focus of the third phase is on the generation of the data files for the VR shop floor data and AUML models data. The information collected from the joint operations between VR and AUML system models is represented in a data file in the extensible mark-up language (XML) file format. XML supports the development of the structure data entities that contain a high level of the semantic content, which is both human and machine interpretable [38–40]. This is widely used as a file format for machine shop data and information modelled using UML and manufacturing simulation software [41, 42]. In the proposed methodology, XML is the encoding mechanism for the exchange of files between system models and matching environment, and is hereafter referred to as the shop data file. The shop data file consists of three groups:

1- Supporting data structures:

- Measurement units (time units)
- Statistics (standard deviations, average, state time, running time)
- Model references (model name, user, date, time)

2- Manufacturing data structures:

- Element name
- Element class

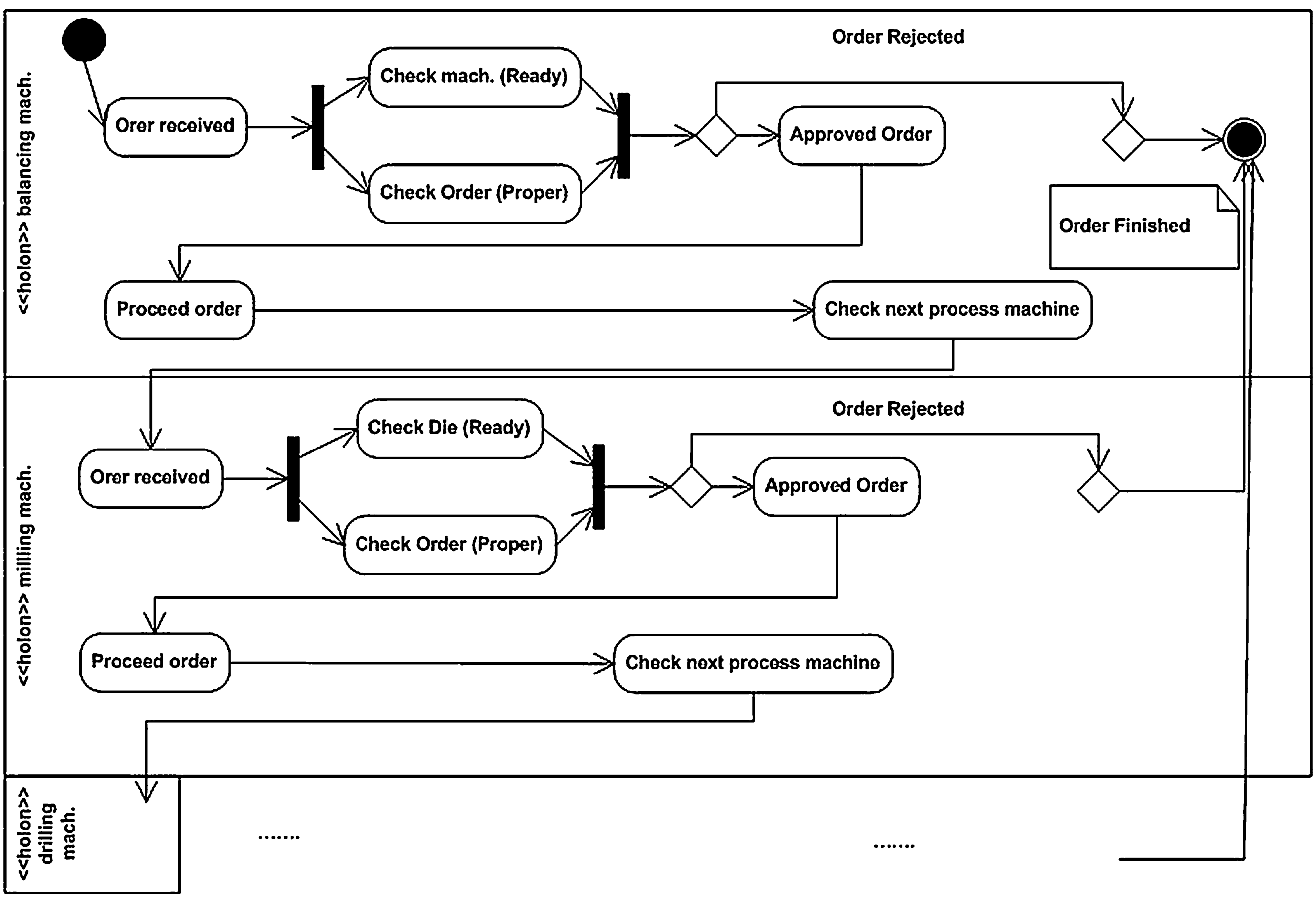

Fig. 5 Activity diagram as event life cycle representation of the die-casting factory

- Resources (machines, labours,…)
- Processes lists
- Events lists
- Element failures
- Daily schedules
- Element busy time
- Element idle time

3- Agents negotiations data structures:

- Message name
- Start message element
- End message element
- Start messaging time
- End messaging time

As the system models are executed, the information collected from the system models is transformed into shop data files. The shop data file of the information domain is the main actor of the matching environment; in particular the shop data file of the interaction protocol diagram is playing an important role since it contains interaction among instances of the agents. After the consistency rules are applied to the interaction protocol diagram, the related shop data file is ready for the matching environment.

The environment that captures the differences of the two shop data files is called the *matching environment*. Discrepancies between operational representation by VR and informational representation by AUML models can be easily captured in the matching environment, (i.e. the differences between the messages, events, roles, actions, etc.). The overall information requirements of the enterprise can be specified by analysing the difference between VR and AUML environments with requirements analysis matching environment. The matching environment can be seen in Fig. 6.

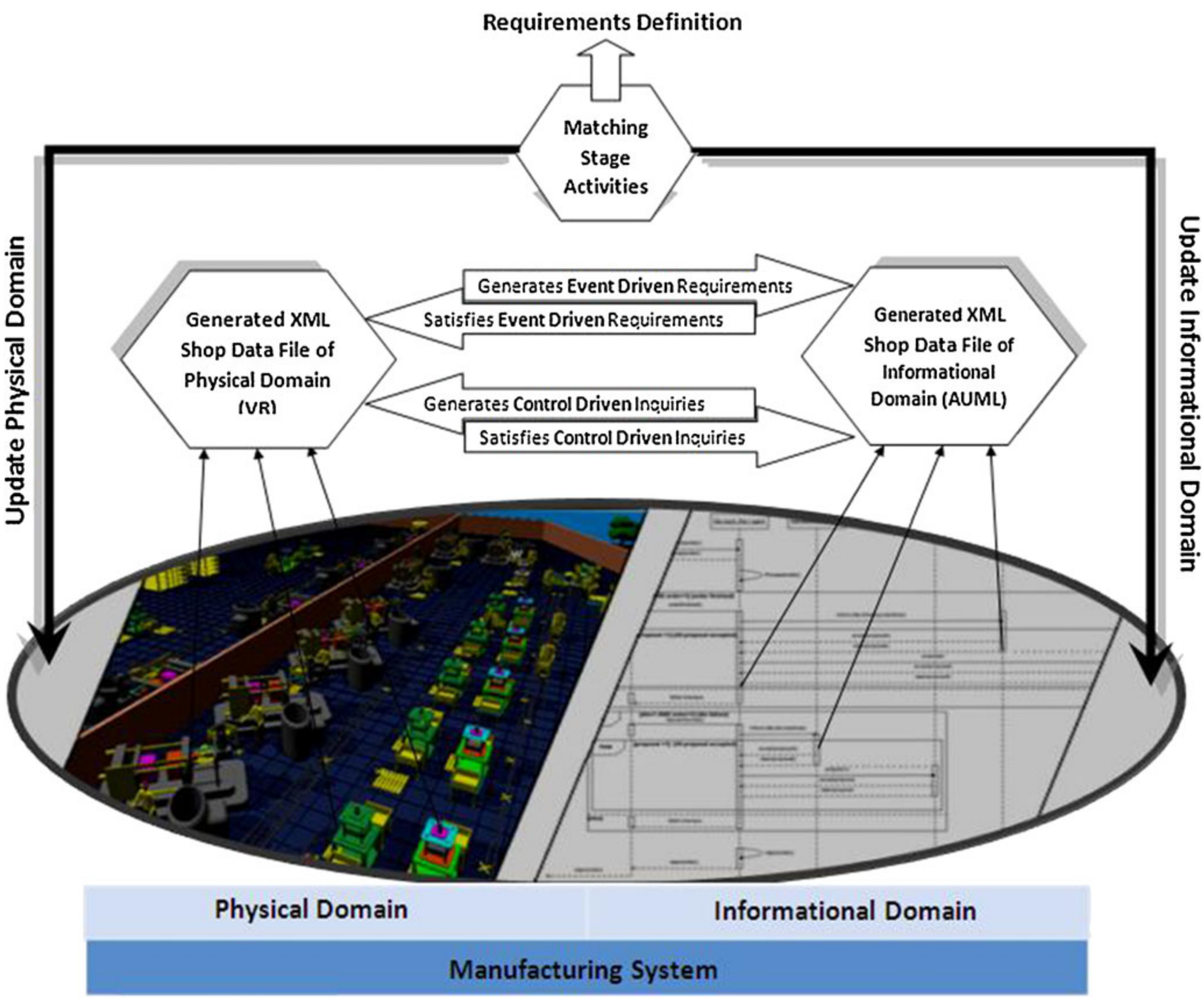

Fig. 6 The matching environment

The virtual HMS environment representing the physical system basically generates event driven requirements, for example the *proceed order()* event raises information needs for the part production initiation (injecting molten metal, press into the mould, cool the part, unload part). The physical domain modelled by the virtual HMS satisfies control driven inquiries of the information system raised by the AUML models through releasing a signal, for example at part completions. There is a reciprocal connection on the data on both sides. The AUML models are generators of the control driven requirements (for instance, the release of a new part to the shop triggered by a schedule) and satisfy event driven information requirements, like availability of machine for new product.

3 A prototype tool: virtual reality holonic manufacturing system requirements analysis

The proposed virtual reality holonic manufacturing system requirements analysis (VR-HMS-RA) prototype tool has been designed and is currently being developed as a software system to realise the features outlined in each phase of the methodology. In this research, the DELMIA™ QUEST package has been utilised for the development and simulation of the virtual model [43]. The virtual factory in QUEST consists of the physical models of manufacturing and material handling equipment, enhanced with the CAD data containing the 3D animation models. The parts to be produced along with manufacturing process attributes such as distributions, constraints and logical dependencies are defined and created through QUEST's user interface functions.

The VR environment and AUML environment are integrated through a graphical user interface (GUI) designed by Java programming language, which provides a graphic environment for the analysis. A sample screen (GUI) of tools can be seen in Fig. 7. The interface is designed in such a way that there are two main divisions: an AUML models editor and a shop data file editor. The division on the left is the informational representation by AUML models. The as-is information models, to-be

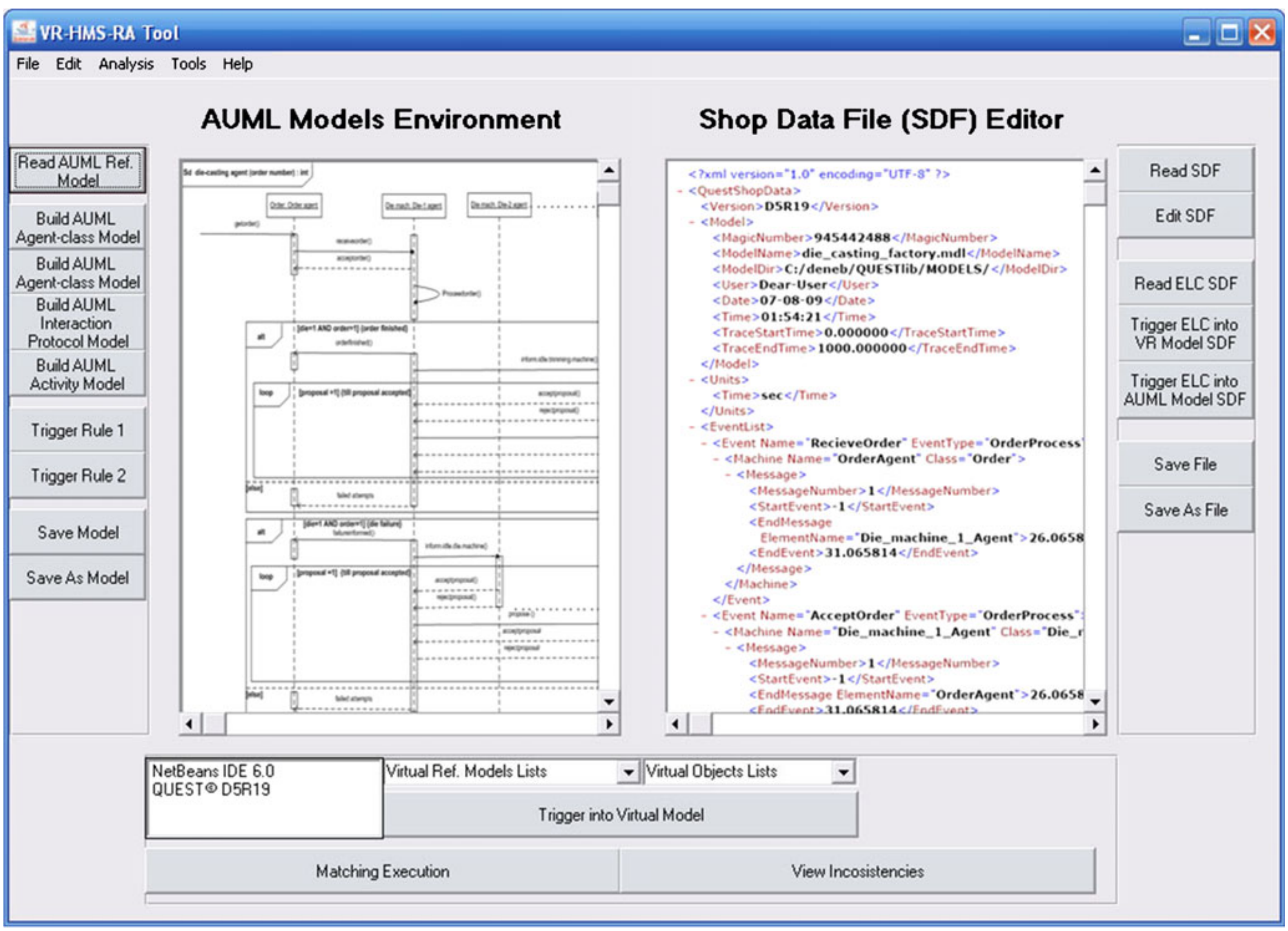

Fig. 7 A sample screen (GUI) of the prototype tool

information models, application of consistency rules and generation of the control driven requirements are the objectives of the division at the left-hand side. The right-hand side is related to shop date file representation, generated by physical and informational environments based on XML format file. ELC representation, ELC execution and the generation of event driven requirements based on shop data files generated by as-is and to-be VR models are the tasks of this division.

4 An illustrative example

In the following section a real industrial plant, a 'medium-sized die-casting factory', is analysed. This company is *Sahin Metal*, located in Istanbul, Turkey. The company uses a typical die-casting manufacturing machine system for the batch production of several light-alloy automotive variants and various aluminium alloy parts, all exported to Europe. The factory comprises of 12 workstations and is physically configured from a labour-operated die-casting machine park, having 11 machines, accompanied by 11 hydraulic presses for trimming, two vibratory debarring machines, seven presses for calibrating of the die-cast products and an automated guided vehicles used for transporting the raw materials.

The production processes begin with the arrival of the aluminium ingots to the melting pot of the die-casting machine from the storage area. At the die-casting station, castings are produced, batched within baskets and kept for cooling for approximately one and a half hours. This is necessary due to thermal protection before the trimming process. The cooled castings are assigned to trimming machines, where the runner and gate ways of the die-casting clusters are trimmed out. At each stage of production, the non-defective parts are batched into the baskets and then sent to vibratory debarring machines for washing and surface finishing of the castings. The calibration presses then perform finishing of the final products. The products are then sent to the eye inspection

Fig. 8 The as-is VR model of die-casting factory example

area before packaging for testing against the possible defects occurring during processing.

In an effort to the improve productivity and overall performance, the company is currently investing in more automation equipment (e.g. robot-operated die casting, two AGVs for transporting finished parts between machines) in order to form a fully automated shop floor, while upgrading the existing shop floor control system by making use of cost-effective methods. Their main goal is to fully implement holonic manufacturing systems in their factory.

4.1 Construct 'as-is' model

The analysis starts with the communication of the analyst and the user about the physical structure of the factory for requirements elicitation. The analyst needs the information such as available resources, control logic, process flow, products and information flow in order to identify the different types of holons based on PROSA architecture for constructing the VR model.

For the die-casting example, the corresponding VR 'as-is' model of the factory has been built by considering the organisation of the current processes of the shop floor under analysis and divided into the following agents: *Order_agents, Die-machine_agents, Timming_machine_agents, Vibration_-machine_agents, Calibrationg_machine_agents, Inspection_agents, Packaging_agents, Supervisor_agents*, and….

The information given above is classified and used by VR-HMS-RA knowledge base to determine the closest VR reference model from its database. The user can explore and modify the VR environment to suit the as-is model of the factory using the virtual object libraries. For instance, an AGV is added to the selected reference model. The dynamic attributes of the real AGV in the factory differ from the candidate object, therefore, the attributes have to be altered. For example, the AGV object in the virtual object libraries has a unidirectional, but in our example the flow direction of the AGV is bi-directional and runs continuously. These characteristics should be modified on the object. This modification, in turn, updates the VR environment. The created as-is VR model of the factory under study is shown in Fig. 8. The next process is to construct the AUML models, getting help from a typical execution in the virtual environment.

4.2 Construct 'to-be' models

For finalising the analysis and constructing the information 'to-be' models, the VR and information models are run concurrently for the corresponding ELC executions tasks in order to produce output data for matching the information files. The regular operations are considered for the preparation of ELC execution tasks. Also, the irregularities, which may take place such as order changes, machine breakdowns etc., are taken into account. The VR and AUML models generated shop data files are then examined by the user for the consistency and logicalisation. In the case of having discrepancies between the execution outcomes of the models (i.e. events, processes, negotiations, etc. are not matching), the mismatches are corrected by reviewing the earlier phases of modelling iteratively until no mismatches are obtained in the execution of the system models. Finally, the accurate shop data is performed for obtaining "to-be" models of the information model in order to define the requirements of a typical HMS implementation.

5 Conclusions

A VR-based methodology for the requirement analysis stage of the holonic manufacturing systems design and development process, which uses virtual reality for representing a manufacturing system, is presented. Virtual reality is computer-supported and has better interaction effects for representing manufacturing systems in operation than other standard graphical user interfaces. The 3D virtual reality environment of HMS-RA allows the agents to operate in a detailed high fidelity world where geometry, shapes and motion exist. This helps users to collect valid information quickly and in the correct form resulting in rapid development of holonic models. An object library serving as a repository of virtual reference models is developed to help the modelling of the physical domains. A novel matching environment for the physical and the informational model domains is suggested to delineate the requirements. A set of rules and a knowledge base is appended to the methodology. Finally, VR-HMS-RA, a prototype software tool currently being developed to support the methodology, is introduced in this paper. Currently, a VR-based CASE tool is being developed for use in requirement analysis stage of the holonic manufacturing systems. Also, the presented methodology is being evaluated with real case studies; for example: the die-casting factory presented in this work, and a firm of Cylinder Head Manufacturing.

References

1. Brennan RW (2003) Norrie DH Metrics for evaluating distributed manufacturing control systems. Comput Ind 51:225–235
2. Tiwari MK, Mondal S (2002) Application of an autonomous agent network to support the architecture of a holonic manufacturing system. Int J Adv Manuf Technol 20(12):931–942
3. Fletcher M, Dean SM (2001) Common generic model of the holonic manufacturing systems project. Proc Ints Mech Eng B J Eng Manuf 215(9):1293–1297

4. Babiceanu RF, Chen FF, Sturges RH (2005) Real-time holonic scheduling of material handling operations in a dynamic manufacturing environment. Robot Comput Integr Manuf 21:328–337
5. Giret A, Botti V (2006) From requirements to holonic manufacturing system analysis. Int J Prod Res 44(18–19):3917–3928
6. Giachetti RE, Martinez LD, Sáenz OA, Chen CS (2003) Analysis of the structural measures of flexibility and agility using a measurement theoretical framework. Int J Prod Econ 86(1):47–62
7. Requirements engineering for the automotive industry. Published by IBM-America one Jun 02, 2009, http://whitepapers.manufacturing.net/whitepaper5676/, Accessed 1 February 2010
8. Jin JQ, Loftus M, Franks IT (1998) A method for the acquisition of users' requirements in discrete manufacturing cell systems. Comput Integr Manuf 11(3):229–242
9. Giret A, Botti V (2009) Engineering holonic manufacturing systems. Comput Ind 60(6):228–440
10. Blanc P, Demongodin I, Castagna P (2008) A holonic approach for manufacturing execution system design: an industrial application. Eng Appl Artif Intell 21(3):315–330
11. Speck A, Pulvermuller E, Heuzeroth D (2003) Validation of business process models. Proceedings of ECOOP2003 Workshop Correctness of Model-based Software Composition (CMC), Darmstadt, pp 75–83
12. Wand Y, Weber R (2002) Research commentary: information systems and conceptual modelling—a research agenda. Inf Syst Resh 13(4):363–376
13. The Standish Group International, The CHAOS Report, 1994 URL: http://www.standishgroup.com/sample_research/chaos_1994_1.php, Accessed 1 February 2010
14. Billo RE, Rucker R, Paul BK (1994) Three rapid and effective requirements definition modelling tools: evolving technology for manufacturing system investigations. Int J Comp Integ M 7 (3):186–199
15. Schmid R, Ryser J, Berner S, Glinz M, Reutemann R, Fahr E (2000) Technical Report 2000.06. Institut for Informatik University of Zurich, Switzerland
16. Hlupic V (2000) Simulation software: an Operational Research Society survey of academic and industrial users. Proc Winter Simul Conf 2(10–13):1676–1683
17. Yang SM, Ahn B, Seo KK (2–6) Development of a prototype customer-oriented virtual factory system. Int J Adv Manuf Technol 28:1031–1037
18. Cecil J, Kanchanapiboon A (2007) Virtual engineering approaches in product and process design. Int J Adv Manuf Technol 31:846–856
19. Wenbin Z, Juanqi Y, Dengzhe M, Ye J, Xiumin F (2006) Production engineering-oriented virtual factory: a planning cell-based approach to manufacturing systems design. Int J Adv Manuf Technol 28:957–965
20. Peng Q, Yu C (2007) Enhanced integrated manufacturing systems in an immersive virtual environment. P I Mech Eng B J Eng 221 (3):477–487
21. Freiheit T, Shpitalni M, Hu SJ, Koren Y (2004) Productivity of synchronized serial production lines with flexible reserve capacity. Int J Prod Res 42(10):2009–2027
22. Knoll JM, Heim JM (2000) Ensuring the successful adoption of discrete event simulation in a manufacturing environment. Proceedings of the 32nd Conference on Winter Simulation, pp. 1297–1304
23. Giret A, Botti V (2004) Holons and agents. J Intell Manuf 15:645–659
24. Fischer K, Schillo M, Siekmann J (2003) Holonic multiagent systems: a foundation for the organisation of multiagent systems. Proceedings first international conference on industrial application of holonic and multi-agent-systems (HoloMas) (2003) LNAI 2744. Springer, Prague, pp 71–80
25. Huget MP, Odell J (2004) Representing Agent Interaction Protocols with Agent UML. Proc. 5th Int'l Workshop on Agent-Oriented Software Eng. Springer, New York
26. Agent UML Website, http://www.auml.org/, Accessed 1 February 2010
27. FIPA Modelling Technical Committee http://www.fipa.org/activities/modeling.html, Accessed 1 February 2010
28. Bergamaschi S, Gionata G, Guerra F, Vincini M (2003) Experiencing AUML for the WINK Multi-Agent System, in Proceedings of AIIA and TABOO Workshop: From Object to Agents (WOA03), Villassimius, CA
29. Beer M, Huang W, Hill R (2003) Designing Community Care Systems with AUML, Proceedings of the International Conference on Computer, Communication and Control Technologies (CCCT '03) and the 9th International Conference on Information Systems Analysis and Synthesis (ISAS '03). IEEE Computer Society, Orlando, pp 247–253
30. Huget MP (2002) An Application of Agent UML to Supply Chain Management, in Proceedings of the Fourth International Bi-Conference Workshop on Agent-Oriented Information Systems. Bologna, Italy
31. Unified Modelling Language (UML) 2.2, http://www.omg.org/spec/UML/2.2/, Accessed 1 February 2010
32. Huget MP (2004) Agent UML notation for multi-agent system design. IEEE Int Comput 8(4):63–71
33. Bal M, Hashemipour M (2009) Virtual factory approach for implementation of holonic control in industrial applications: a case study in die-casting industry. Robot Com Integr Manuf 25 (3):570–580
34. Brussel HV, Wyns J, Valckenaers P, Bongaerts L, Peeters P (1998) Reference architecture for holonic manufacturing systems: PROSA. Comput Ind Special Issue Intelligent Manufacturing Systems 37(3):255–276
35. Xu ZJ, Zhao ZX (2000) Baines RW constructing virtual environments for manufacturing simulation. Int J Prod Res 38(17):4171–4191
36. Maccarthy BL, Fernandes FCF (2000) A multi-dimensional classification of production systems for the design and selection of production planning and control systems. Prod Plan Control 11 (5):481–496
37. Egyed A (2007) UML/Analyser: A tool for the instant Consistency Checking of UML Models, 29th international Conference on Software Engineering (ICSE'07)
38. DuCharme B (1998) XML: the annotated specification. Prentice Hall PTR, Upper Saddle River
39. Goldfarb C (2002) XML handbook. Prentice Hall, Upper Saddle River
40. Manufacturing XML Promotion Forum: {http://www.mfgx-forum.org/}, Accessed 1 February 2010
41. McLean C, Jones A, Lee T, Riddick F (2002) Manufacturing modelling architectures: an architecture for a generic data-driven machine shop simulator. Proceedings of the 34th conference on Winter simulation: exploring new frontiers, December 08–11, San Diego, California
42. Kojima T, Ohtani S, Ohashi T (2008) A manufacturing XML schema definition and its application to a data management system on the shop floor. Robot Com Integr Manuf 24(4):545–552
43. Delmia QUEST, Dassault Systems. http://www.delmia.com, Accessed 1 February 2010