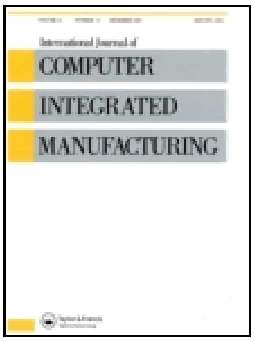

International Journal of Computer Integrated Manufacturing

ISSN: 0951-192X (Print) 1362-3052 (Online) Journal homepage: www.tandfonline.com/journals/tcim20

Combined enterprise and simulation modelling in support of process engineering

K. A. Chatha & R. H. Weston

To cite this article: K. A. Chatha & R. H. Weston (2005) Combined enterprise and simulation modelling in support of process engineering, International Journal of Computer Integrated Manufacturing, 18:8, 652-670, DOI: 10.1080/09511920500159803

To link to this article: https://doi.org/10.1080/09511920500159803

Published online: 17 Feb 2007.

International Journal of Computer Integrated Manufacturing, Vol. 18, No. 8, December 2005, 652 – 670

Combined enterprise and simulation modelling in support of process engineering

K. A. CHATHA and R. H. WESTON

MSI Research Institute, Loughborough University

Industry requires improved means of designing and changing its business processes. This paper describes an enhancement to state-of-the-art enterprise modelling techniques. Use of the approach is illustrated by generating and exercising simulation models that support the design and optimization of example processes.

Keywords: Business processes; Enterprise modelling; Simulation modelling; Feasible resource selection

1. Needed enterprise modelling enhancements

A number of well known approaches to modelling manufacturing enterprises (MEs) are described in the literature. They include: CIMOSA (CIMOSA 1993), IDEF (ICAM 1981a and ICAM 1981b), GRAI (Chen and Doumeingts 1996), PERA (Williams 1996) and IEM (Spur *et al*. 1996). The General Enterprise Reference Architecture and Modelling (GERAM) specification (GERAM 1999) was developed (by an IFAC/IFIP working group) to classify and graphically represent an abstraction of common enterprise modelling capabilities embedded into well known approaches. It follows that GERAM is indicative of the status quo in enterprise modelling. By making reference to GERAM, would-be industry users can select a suitable enterprise modelling approach (be that GRAI, CIMOSA, PERA or some other approach). Any such selection should enable users to deploy well defined and proven decomposition and modelling concepts; thereby users can (a) cope with high levels of complexity inherent in MEs and (b) formally model networks of dependent processes and (human and technical) systems needed to realize specific ME purposes (Chatha *et al*. 2003).

Enterprise models created in this way can coherently represent MEs from different viewpoints and at alternative levels of abstraction that suit needs of model users. Also well decoupled models of process segments and sub-systems can be created and reused; and subsequently can be reintegrated into an holistic enterprise model (Vernadat 1996). This can be viewed as focusing in on processes of concern whilst retaining the ability to view, and interpret the use of process and resource models within a wider (specific) ME context. In principle therefore by deploying a GERAM conformant enterprise modelling approach, the need for large and small ME changes can be specified and communicated effectively throughout the lifetime of ME processes and systems.

In theory therefore state-of-the-art enterprise modelling approaches facilitate the development of better processes and systems, and can improve the timeliness and cost effectiveness of change projects in MEs. But unfortunately in practice the full potential of enterprise modelling has yet to be realized and partial benefits have been realized in only a small percentage of enterprises.

The authors, and many of their research colleagues from around the globe, have observed that although industry at large has become increasingly aware of the need to better understand (model) and improve the processes it uses to realize products and services (short and long term), generally it deploys simple process mapping techniques, in the sense that they do not impose strict modelling formalisms and decomposition principles; such as those embedded into state-of-the-art enterprise modelling approaches. As a consequence the ‘as is’ process maps they develop can only provide limited support for process (and resource system) design and change. The support that is

*Corresponding author. Email: r.h.weston@lboro.ac.uk

International Journal of Computer Integrated Manufacturing
ISSN 0951-192X print/ISSN 1362-3052 online
http://www.tandf.co.uk/journals
DOI: 10.1080/09511920500159803

provided appears primarily to be with respect to (a) enabling groupings of company personnel (with different responsibilities, concerns and points of view) to develop the use of a common picture (of current company practice) and (b) to develop ad hoc ways of realizing localized process change. In theory this *ad hoc*-ary will result in: (1) process designs and implementations that cannot readily be exercised in the form of computer models; that support 'what if' analysis associated with process change and process monitoring and control, (2) inferior process designs and implementations, that cannot readily and effectively be: justified in business terms, nor integrated into the host organization and its business and environmental contexts and (3) process designs that cannot readily and effectively be reused, when at some later time in the life of the enterprise new process change requirements emerge.

Deployed properly, existing enterprise modelling technology has potential to alleviate problems described under (1) through (3). But the process of capturing comprehensive coherent enterprise models is known to require skilled and time consuming effort. Further subsequent reuse and re-integration of resultant models must be well organized and properly supported by tools; but currently there is a lack of technology provision in this respect. Apparently such problems present major hurdles to industry at large. The authors believe that the real power of enterprise models comes from their potential to be reused and reintegrated (with other models) on an ongoing basis and that in general this will necessitate a need to readily develop computer executable models of process segments and related resource systems derived directly from formal descriptions of process networks (i.e. process maps) such that:

(i) predictions can be made about alternative future behaviours of processes, by exercising simulation models to enable selection to be made between alternative candidate process designs and so that the influence of possible, but uncertain future requirements and conditions, can be analysed in advance of their happening, and
(ii) computer executable process models can be 'connected' to the actual resource systems used to realize the business processes, such as by using workflow management systems to exert 'change capable' controls and 'change capable' information gathering and managing facilities.

Here it is re-emphasized that MEs are complex dynamic systems that need to change their behaviours over time. One way of better understanding their dynamic properties is to adopt the use of decomposition principles (embedded into a selected enterprise modelling approach) so that (i) models of process networks that are formally segmented into: essentially separate (or well decoupled) models of segments of processes (and their related activities); and models of dependencies between process segments, (ii) separating the modelling of process segments from models of their candidate resource systems (i.e. resource systems that in principle have capabilities, competencies and capacities needed to realize process segments, and dependencies between process segments) and (iii) separately modelling time dependent flows of material, sub products, products, information, controls and exception handling that place a 'workload' on modelled process segments and their selected resource systems.

Here it is understood that such modelling separations are commonly used in ad hoc ways in many (possibly most) MEs because: processes are naturally modularized to some degree to facilitate their management and development; systems of various types and with various capabilities are separately engineered, selected and attributed to process segments over time; workloads exerted on processes (and process segments) take various forms (that typically vary significantly over time [such as when customer orders change, when new products are introduced and realized, when process improvements need to be carried out, etc]) and require various forms of control strategies and mechanisms to realize workloads effectively, efficiently, robustly and in timely ways.

From their previous experience of using a number of state-of-the-art enterprise modelling approaches, the authors also observed that state-of-the-art enterprise modelling concepts are primarily geared towards capturing and reusing coherent models of static (i.e. relatively enduring) aspects of model types referred to under (i) through (iii); but in general provide little support for modelling dynamic (i.e. time dependent behaviours) of those model types.

Bearing in mind deficiencies of state-of-the-art enterprise modelling approaches outlined above the authors have conceived ways of coherently using enterprise modelling, simulation modelling and workflow modelling concepts to facilitate the systematic capture and reuse of both static and dynamic models of complex MEs. This paper reports on progress made in respect to coherently using enterprise and simulation modelling techniques. In so doing it extends the literature by showing how coherently transformed versions of process and resource system models, which comprise decomposed elements of an holistic enterprise model, can be 'exercised' to realize specific benefits of type indicated under (I).

2. Description of case study problem domain

Since the late 1990's an international consortium of automotive companies and their systems suppliers have been participants in a collaborative engineering programme aimed at *mass customizing four and five cylinder petrol* *engines on a global scale* (Harrison *et al*. 2001). The engineering programme was founded on the notion that a select group of automotive manufacturers could use a *common base of engine technology and production systems* but would remain competitors with respect to the sale of fully packaged cars at locations around the globe. The partners conceived the programme so that it comprised a number of dependent threads of engineering process, including (1) *product design and rationalization processes;* (2) *production process design and globalization processes;* (3) *production machinery design and build processes* and (4) *production facility installation and 'roll out' processes*.

Through their lifetime (order of four years) these engineering process threads needed to be resourced by a core group of around ten internationally prominent companies including automotive product manufacturers and automotive production system vendors. Researchers at the MSI Research Institute at Loughborough University have used enterprise modelling approaches to document various activities carried out by the core group of engineering partners involved with a view to (a) understanding and abstracting current best industry practice and (b) proposing and developing enhancements to that practice.

Necessarily each of the engineering process threads (1) through (4) was complex in its own right. This was because of: the dependent nature of the various engineering activities involved (and dependencies between outputs produced by these activities); because of the need to satisfy multiparty (business partnership) desires and concerns; and to coordinate the geographically distributed efforts of those parties. Also it was observed that naturally complex dependencies link engineering activities that need to be carried out in different 'parallel' process threads. For example, product design decisions may impact significantly on the design and implementation of human and technical systems used to achieve production processes, and vice versa.

Early engineering work of this automotive consortium rationalized and enabled a shared use of a *common set of car engine components* amongst a small number of car manufacturers. The aim here was to seek competitive advantages relative to other car manufacturers. To achieve standardization of engine parts, the product designer and production facility designer and developer members of the consortium needed to impose constraints on product customization. But they also needed to ensure that sufficient *product flexibility* and *production flexibility* was retained so that all needed makes and models of car engine can be produced in economically viable quantities using standard production processes that can be deployed on a global scale.

It was realized that technical and commercial benefits of standardizing and sharing engine components arose for three connected reasons, i.e. from improved 'production process decoupling'; 'production process simplification'; and 'production process standardization'. An outcome of product modularization and standardizing interfaces among components is that the manufacture of engine components is decoupled from the manufacture of other engine components and from the configuration of customer specified car engines. In this case six well decoupled threads of production (or process modules) were identified that afford new opportunities to mass produce engine components whilst customizing the manufacture of final engine products in smaller (possibly much smaller) batches. The consortium refers to this phenomenon as 'mass customization'. In theory at least the resultant process decoupling can suit both customer needs (for product variety) and manufacturer needs (for cost effective production).

Production process simplification is also a natural outcome of product modularization and standardization, and the process decoupling it enables. This is because if a relatively small number of standard components can be well defined then the production and logistical activities needed to make and transport these components can be simplified. Particularly because of resultant process decoupling one can surmise that no longer will there be such complex interdependencies linking 'make' and 'transport' activities.

The reader can deduce that process simplification and decoupling can naturally lead to process standardization. By such means it may become possible to define sub processes (and associated 'make' and 'transport' activities) more completely, to communicate process requirements and problems more effectively and to distribute responsibilities (in space and time) for process realization.

In tandem, therefore, the process of decoupling, simplifying and standardizing production and logistical processes can be key to realizing complex products on a global scale. These three outcomes from product modularization are likely to much simplify engineering projects, processes and activities associated with the design, realization and change of production processes and facilities. In principle therefore good product and process decompositions are likely to much simplify subsequent process change, process development and ongoing process improvement.

Many discussions with consortium partners helped to qualify aspects of a common industry problem which is centred on current lack of technology provision needed to facilitate: *reasoning about;* and *ongoing maintenance* of dependencies between engineering activities carried out both within and between multiple process threads. This paper reports progress made by MSI researchers in addressing aspects of this problem, which is presumed to be a common one in consortia of companies wishing to interoperate on a global scale.

3. The multi-process modelling approach

Successive process modelling studies by staff and students in the MSI Research Institute led to the specification and a variety of uses of a Multi-Process Modelling (MPM) method, described in (Chatha *et al.* 2003). MPM was originally conceived to lend a methodological structure to the use of state-of-the-art enterprise modelling concepts and tools and in so doing to help organize the reuse of (1) a semantically rich picture of multiple dependent processes, in such a way that key dependencies can be explicitly represented and maintained during process lifetimes; (2) coherent multi-perspective views of process models for different enterprise engineering purposes. Essentially MPM structures and extends the use of previously available CIMOSA modelling concepts, which provide a backbone of representational primitives. Following are the five stages recommended by MPM.

1. Elicit and record multiple understandings about current business processes deployed by the engineering partners, with the aim of developing a unified set of process representations that collectively form a static pool of enterprise knowledge that can be reused for various purposes.
2. Reuse the data elicited to populate and validate multiple 'static views' of 'as is' business processes that collectively and coherently provide a 'semantically rich picture' of relatively enduring enterprise entities and their interrelationships that can be reused by different enterprise personnel in support of their various roles.
3. Recode selected elements and aspects of the static representations of 'as is' processes into computer executable models with capability to simulate process behaviours when loaded by one or more product or project flows, thereby providing new insights into 'as is' process design, process resourcing and process operation. Initial dynamic model analysis and development is focused on model validation with subsequent analysis on identifying possible constraints arising from 'as is' practice.
4. Based on knowledge gained about 'as is' process properties (static and dynamic), develop new 'to be' business process scenarios and exercise these scenarios using suitable simulation modelling techniques. This provides metricated analysis of alternative: process designs; attributions of resources to process elements; and process operations.
5. One important potential use of 'as is' and 'to be' static and dynamic process models is to manage and control workflows. At this modelling stage selected model fragments (previously captured and validated) should be recoded so that they can be executed by a suitable workflow tool. This allows computer executable models to be linked to the actual process and its resource entities.

Figure 1 summarizes complementary strengths of these three modelling approaches. MPM was conceived to combine those strengths within a single modelling approach, capable of representing and reusing:

(A) 'static models' of process networks, comprising multi-perspective views of dependent processes at needed abstraction levels;
(B) dynamic descriptions of process and resource model elements (belonging to segments of the modelled network) and their time dependent loading by workflows (arising for example from multiple product flows); that can be executed via well proven simulation tools so as to replicate, predict and improve upon real process and system behaviours, with reference to their wider impact on the parent process network;
(C) dynamic descriptions of workloads and their impact on process and resource segments, that can be executed by proven workflow tools; so as to provide mechanisms to enact process and resource models and thereby achieve improved control and monitoring of actual ME workflows.

3.1. Use of enterprise modelling to capture context dependent 'as is' process maps

The first two stages of the MPM method centre on 'as is' process mapping and lead to the capture of a pool of static[†] data, which incorporates various views of multiple processes so that alternative uses of the information pool can be promoted. The CIMOSA enterprise modelling architecture was found to provide suitable formalisms to (a) decompose complex systems (of processes) into sub-systems (or many largely self-standing sub-processes) that can be analysed independently and (b) during later stages of analysis and project engineering to recompose sub-systems into a collective whole. More specifically MPM builds upon the use of four types of diagramming template specified within the CIMOSA function view, namely: 'context-diagrams', 'interaction-diagrams', 'structure-diagrams' and 'activity-diagrams'. This set of CIMOSA diagramming concepts and graphical modelling constructs was observed to provide effective means of graphically representing key

[†]In this context the term 'static' indicates that model capture is focused on representing relatively enduring process entities and entity relationships, this being centered on the normal flow of activities carried out. Such a focus of attention is distinct from that of modelling dynamic (short term) properties of processes such as when modelling specific activity states and state transitions required to make a particular engineering decision.

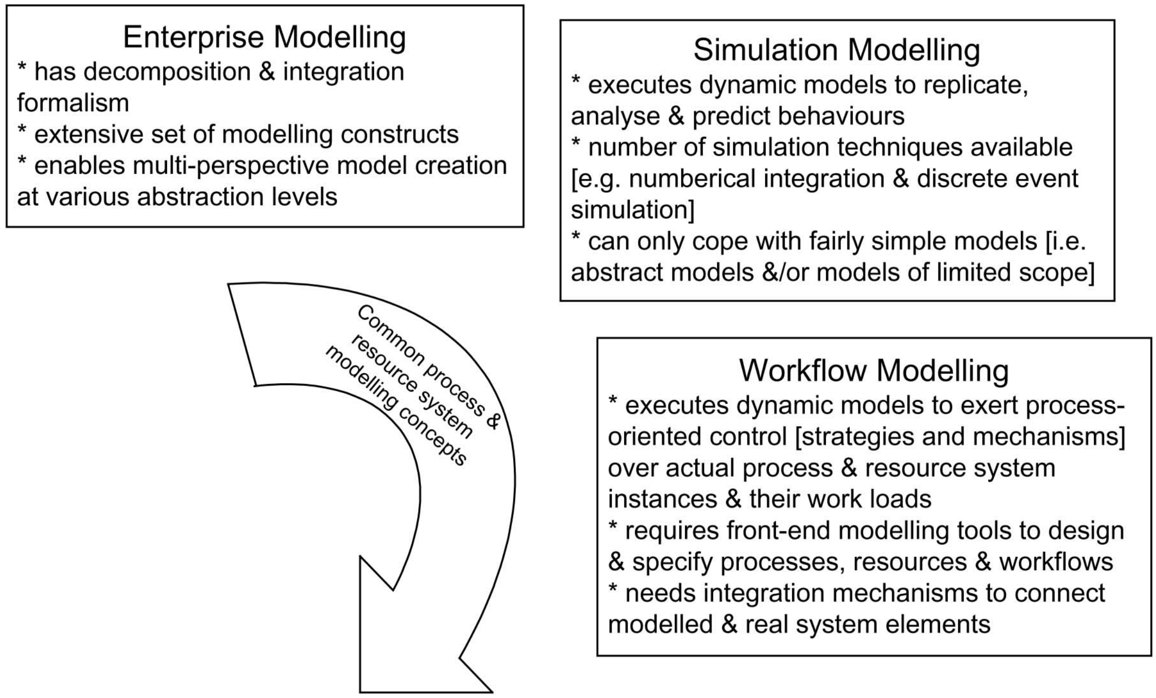

Figure 1. Integrated modelling concepts underpinning MPM.

attributes of the 'as is' engineering processes used in the automotive consortium problem domain described in section 2. Out of these four diagrams 'activity diagrams' were deployed to formally attribute descriptions of precedence links between business processes and enterprise activities. Typically therefore when MPM is deployed an 'activity-diagram' is created for each process segment of concern to enable the order of execution of a set of related enterprise activities to be defined. The status of precedence links (which might be specified in terms of designated process conditions, actions and events) can determine and/ or modify the order in which activities are carried out during process execution.

CIMOSA's activity diagramming templates were found to provide effective and practical ways of developing mid to long term views of dependency relationships, that couple complex process segments (groupings of activities) and resources that collectively constitute the engineering processes used by members of the case study automotive consortium. Figure 2 shows one of many diagramming templates used to formally map process segments used by one machine-builder partner of the consortium described in section 2. This particular machine builder designs and builds the production machines and lines used by a number of well known car manufacturers. A more complete CIMOSA model of the processes used by this machine-builder partner can be found in Chatha *et al.* (2003) and Chatha (2004). It was observed that the resultant static models, coded graphically by many related template model fragments, can be followed from top to bottom: both in respect to individual partners and with respect to interactions between consortium partners. Thereby these static models were found to represent and communicate multiple pictures of dependent processes in a coherent, step-by-step manner at different levels of abstraction. Figure 2 is in fact one of a number of activity diagrams created to define the sequence of domain processes, business processes and enterprise activities utilized by the machine-builder company modelled. The elemental activities of this figure can be interpreted within the context provided by related context, interaction, structure and other activity diagrams. Further, section 3.2 explains how information formally encoded by this diagram was reused, so as to create dynamic models that enable process behaviours to be simulated.

CIMOSA's well defined and well structured approach proved capable of elaborating an holistic (but static) picture of the automotive engineering programme described in section 2. Its modelling concepts cut through much of the complexity that previously had inhibited collective understanding and the coherent development and description of more focused concerns within a well defined common context. The capture and subsequent reuse of the static data pool was also found to prove effective in identifying, visualizing and communicating the purpose of groupings of activities and thereby possible roles that might

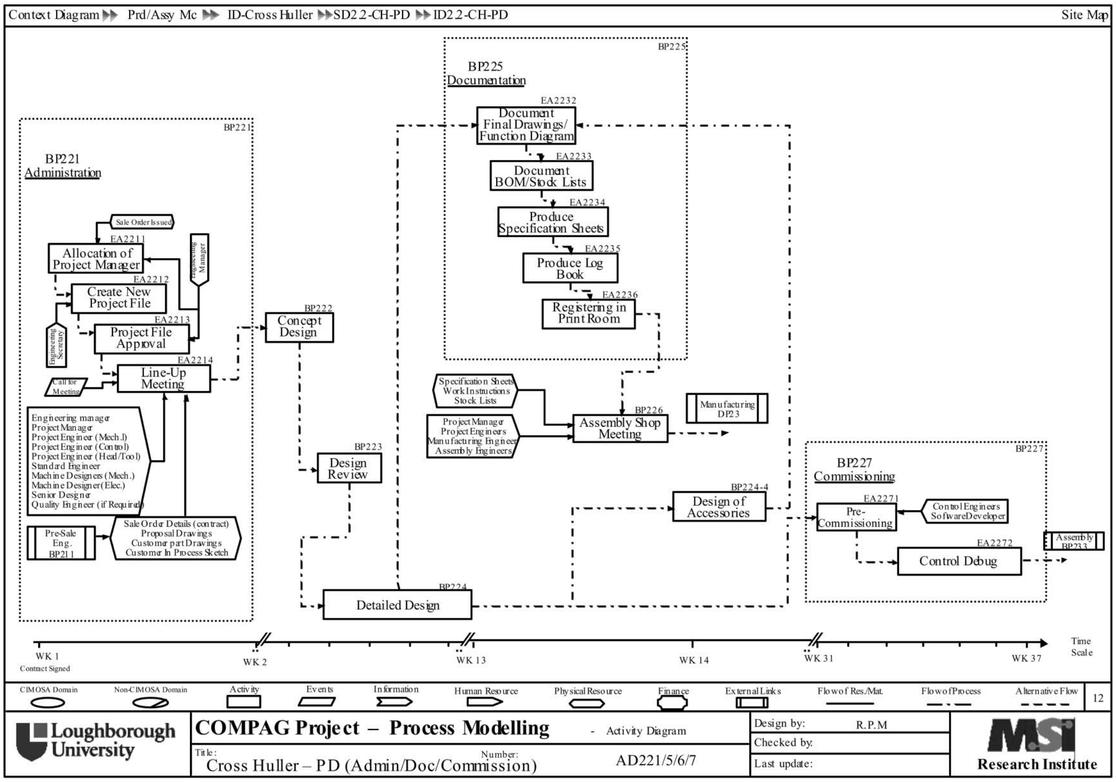

Figure 2. Example 'activity diagram' captured to graphically represent a segment of a production machine design and build business process deployed by one of the machine builder partners.

be played by different resource groupings. On the other hand, use of the diagramming templates alone was not found to provide qualitative means of exercising 'what if' thinking. Nor do they directly encode certain real time working concerns, such as about how individuals or dependent processes might alternatively behave in response to changes in process inputs, process flows, operating conditions and so forth.

This led to an identified need for decision making capabilities which can help to (i) gain new insights into current and new process designs, process resourcing and process operation and (ii) realize metricated analysis of possible alternative scenarios discovered under (i). Therefore stages 3 and 4 of the MPM method (see section 3) were designed and developed to complement earlier stages of modelling. Here it was decided that MPM development should be based on the combined use of (a) CIMOSA model 'generation', 'instantiation' and 'derivation' principles applied to the static data pool and (b) dynamic '*Systems Thinking'* and the selective use of modelling concepts and mechanisms previously developed at MIT in the USA (Forrester 1968). *System Thinking* concepts were adopted because they naturally support the process of generating clear mental pictures about complex systems. They provide means of understanding the effect and strength of *cause and effect relationships* that otherwise may not be readily apparent. When they were used in combination, CIMOSA modelling and 'qualitative' and 'quantitative' versions of systems thinking concepts and computer tools were found to provide effective means of developing and reusing computer simulation models that draw from a pre-established pool of relevant information about dependent processes. By creating simulation models with respect to the holistic information pool it became possible to exercise simulation and workflow enactment models, within a well-defined organization, business and environmental context.

Many different software tools could usefully have been selected and used to simulate dynamic behaviours of alternative process designs and configurations and thereby to lend support for stages 3 and 4 of the modelling method. However a perceived strength of the selected *ithink* tool was its ability to help visualize process behaviours and process dependencies in a way that can unify views and concerns of different stakeholders (iThink Software 2002). Being essentially a tool for solving time dependent sets of differential equations that describe interrelated physical process behaviours, it provides excellent behaviour simulation capabilities at an aggregated level. It was observed that a developed use of the tool usefully supported hypothesis and new policy testing associated with 'could be' and 'to be' multi-process scenarios. However the tool was found to have limited ability to support detailed aspects of process and human and IT system design, hence there was also an observed need to develop the complementary use of other commercial simulation tools within the context of the MPM method.

3.2. *Use of simulation modelling in MPM*

During stages 3 and 4 of the multi-process modelling method, an underlying purpose of modelling was to better understand the dynamic nature of domain processes and their interdependencies from viewpoints of concerned parties. These new understandings led during stage 3 to a measured assessment of the effectiveness of current practice and in stage 4 to assessments of benefits and risks of alternate practice. Specific modelling studies related to the engineering processes of the automotive consortium confirmed the complex and time dependent nature of process dependencies, showing how decisions made and action taken by a responsible party in one segment (or sub-domain) of a domain process could significantly influence the magnitude and distribution of costs and lead-times associated with partner roles in other process segments. To satisfy that need new simulation methods were developed based on the combined use of CIMOSA and *Systems Thinking* concepts.

The static data pool (coded via CIMOSA graphical templates) provided source knowledge about how 'as is' human and technical resources performed enterprise activities, whilst acting and generating information flows, control flows and material flows, so as to generate suitable process outputs. Also coded within that data pool were trigger events and ending statuses that called and terminated process interactions. However, cause and effect model concepts and graphical modelling primitives used to model systems in *ithink* required quite distinctive ways of visualizing and representing processes, and particularly were centred on multiple differential equation solving to exhibit complex system behaviours. *Systems Thinking* did not define specific semantics of any given problem domain, such as domain processes, business processes, enterprise activities etc. Nor did it enforce any problem decomposition rules, such as a need in the case of CIMOSA modelling for separated modelling of: process requirements; and resource (and related control system) solutions. Rather it facilitated the use of very general modelling constructs which included: variables, links, loops, stocks and flows. The *ithink* modelling tool offered three levels of modelling abstraction to support the efforts of model users and model developers. It followed that CIMOSA and *System Thinking* concepts were complementary but that high levels of skill and human intuition was needed to convert process understandings coded into the static data pool into effective simulation models that could be exercised for some purpose using the *ithink* tool.

Examples of transforming static models into *ithink* simulation models are detailed in (Chatha *et al*. 2003). This transformation facilitates certain dynamic behaviours to be produced. However the types of and constraints placed on those behaviours will depend upon the choice of simulation approach and on the modelling capabilities embedded into the simulation tool used. This is illustrated by figure 3.

Assuming a modeller has a good working knowledge of the selected EM formalisms and simulation modelling (SM) concepts and tools, when achieving EM to SM transformations the modeller must:

1. Create a simulation model, that provides a baseline structure to capture the dynamics of a specific process segment, by deploying modelling concepts and constructs provided by the selected simulation tool
2. Specify and input specific simulation model details related to specific process segments in order to produce those dynamic behaviours of concern to would be 'clients' of the SM.

From experience gained from mapping between CIMOSA activity diagrams and the *ithink* tool stages (1) and (2) were usefully informed in respect of objectives (A) and (B) mentioned in the start of section 3. Common understandings gained about these transformations are described below:

1. When using specific *ithink* modelling layers (so called Interface Layer and Model Construction Layer of the *ithink* tool) to create dynamic models of process segments at suitable levels of abstraction it is necessary to:
   (a) understand the modelling concepts and capabilities embedded in the *ithink* tool;
   (b) understand CIMOSA decomposition principles and particularly its domain process (DP), business process (BP) and enterprise activity (EA) concepts;
   (c) match *ithink* capabilities at each layer of its modelling layers to representational requirements of CIMOSA DPs, BPs and EAs;
   (d) deploy appropriate *ithink* modelling constructs to achieve conceptually similar functionality between CIMOSA and *ithink* model entities.
2. When modelling CIMOSA entities at the Interface Layer of the *ithink* tool (a layer that can be used at a

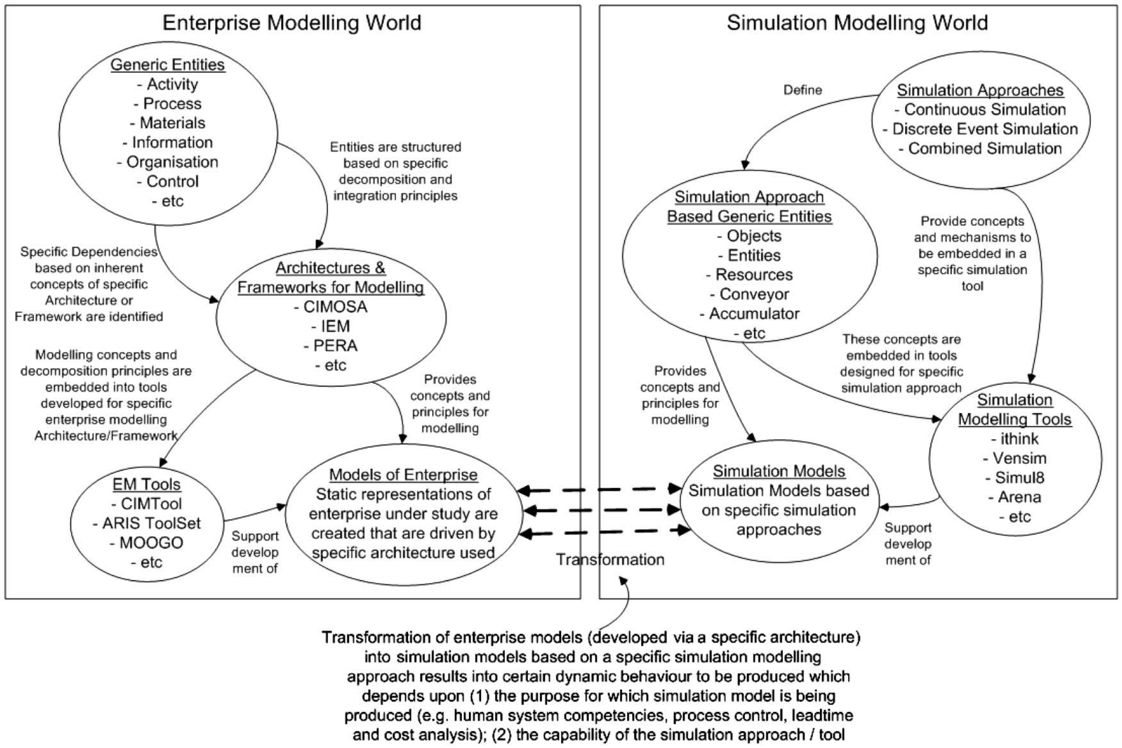

Figure 3. Transformation of enterprise models into simulation models.

high level of abstraction to present managerial information) of the *ithink* tool it is necessary to:

(e) understand the purpose and capabilities of *ithink* Interface Layer modelling constructs;
(f) choose appropriate *ithink* modelling constructs to model entities and attributes of BPs;
(g) choose appropriate *ithink* modelling constructs to model information and control flows;
(h) map BP attributes to *ithink* modelling construct provisions.

3. When modelling EAs at the *ithink* Model Construction Layer (a layer that can be used for lower dynamic systems modelling and development) it is necessary to:
   (i) understand *ithink* modelling construct provision at the Model Construction Layer;
   (j) choose appropriate *ithink* modelling constructs for modelling EAs;
   (k) choose appropriate modelling constructs to model information and control flows;
   (l) choose appropriate *ithink* modelling constructs to model resources;
   (m) develop formalisms to represent resource flows;
   (n) map EA attributes to *ithink* modelling construct provision.

When transforming CIMOSA activity models into ithink simulation models in general:

- BPs were modelled at the Interface Layer of the *ithink* simulation tool;
- EAs were modelled at the Model Construction Layer of the ithink simulation tool;
- Dependencies among BPs, EAs and resources were based on CIMOSA concepts;

Various models of process segments were developed to analyse ways of achieving process leadtime and cost improvements for the case study consortium. However, it became evident that CIMOSA based activity diagrams provided little support for modelling time dependent attributes of processes. The real time working of processes requires (1) means of specifying and facilitating the flow of control and flow of data; (2) means of specifying, monitoring and reporting upon states and statuses of each activity and of parent sub-processes and processes; (3) means of specifying and handling the kinds of exception that can occur during process execution; (4) mechanisms that enable communication and collaboration among resources to be supported; (5) means of defining functional capacity, performance attributes and constraints; (6) types of event to be specified that can impact on activities, sub-processes and processes; (7) means of specifying how process elements should be resourced; (8) specifying behavioural aspects of resources. None of these issues (that are not claimed to comprise an exhaustive list) are supported adequately by CIMOSA modelling concepts and constructs.

It follows that opportunities exist to improve the ease with which, and completeness of which transformations are achieved between segments of enterprise models and simulation and workflow models. Also observed is that a semantically enriched enterprise model could facilitate improvements.

4. Rationale for developing E-MPM: to further enhance the combined use of enterprise and simulation modelling

As explained in section 3, case study modelling of process networks used by MEs uncovered new opportunities to improve upon the design of MPM and this led to the development of an Enriched version of MPM, namely E-MPM. Relative to MPM, E-MPM facilitates simulation modelling and workflow modelling by improving upon the ease with which segments of static enterprise models can be transformed into equivalent computer executable forms. The main enhancements made centre on use of a new form of 'semantically enriched activity diagram', which replaces the use of CIMOSA activity diagrams. This new modelling template is termed an E-MPM 'process diagram' and is designed to encode enriched representations of process segments that can be used coherently with established CIMOSA 'context', 'interaction' and 'structure' diagramming formalisms (Chatha 2004).

4.1. General process modelling language

Associated with semantically enriched 'process diagrams' is the use of a new generic process modelling language, which formalizes aspects of 'model creation' and subsequent 'model execution'. E-MPM development assumed that semantic richness is needed to represent activities, dependencies between activities and other related entities that constitute a process so that resultant representations could naturally form the basis of computer executable process models that have multiple purposes. A second underlying assumption pertains to the need for sufficient modelling concept types including the use of sub-concepts like 'supplier-customer relationships' and 'communication' between modelled entities. Like MPM, the new approach requires use of 'business process', 'enterprise activity', and 'resource' modelling concepts that conform to the CIMOSA specification. Also vitally importantly it enforces an application of these concepts, and their method of decomposition, so that simulation modelling is carried out in a well defined contextual setting; in the case of the automotive setting this context was a specific concurrent engineering one, involving inter-business interactions and therefore was subject to various business aims and objectives. In addition, E-MPM process diagrams are created by utilising new semantic attributes and definitions that facilitate run-time flexibility of processes by exploiting role based resource matching and exception handling mechanisms (see Chatha 2004). Effectively, E-MPM's generic process modelling language is capable of generating static models of processes from different stakeholder viewpoints, which include a process execution perspective. An example process diagram, developed using the E-MPM generic process modelling language is shown in figure 4.

A general template was created in order to formally represent E-MPM process diagrams. That template was designed to serve the purpose of: identifying the domain under consideration; providing details about the model designer; and history data about model developments, amendments and approvals. A number of case study processes were selected as the subject of study and this led to many example populated model templates. The illustrative process diagram shown in figure 4 was generated by the following modelling steps:

1. Case study data was elicited and marshalled to provide necessary information to develop the static models.
2. The information needed was gathered and organized in tabular form so that it could be fed straightaway into the static models.
3. E-MPM's Process Modelling Language principles were applied to create the static models by deploying the Visio^{TM} tool.

Having followed these modelling steps, process diagrams encode information about starting-points, end-points and precedence relationships governing groupings of enterprise activities that comprise specific process segments being modelled. The process diagrams also encode requirements information about the active resources needed to realize process segments, about passive-used resources and passive-processed resources that need to be transformed into 'products'. EA, BP and resource (active or passive) modelling constructs are defined as part of the E-MPM modelling language and are used by the process diagram builder to encode key attributes of process segments that will structure, enable and constrain process behaviours during their subsequent execution by a selected simulation modelling tool.

Consider the example 'Schedule & Monitor Engineering Performance Process – BP222-22' modelled by figure 4. This corresponds to one segment of the wider-scope process map described by the CIMOSA activity diagram shown in

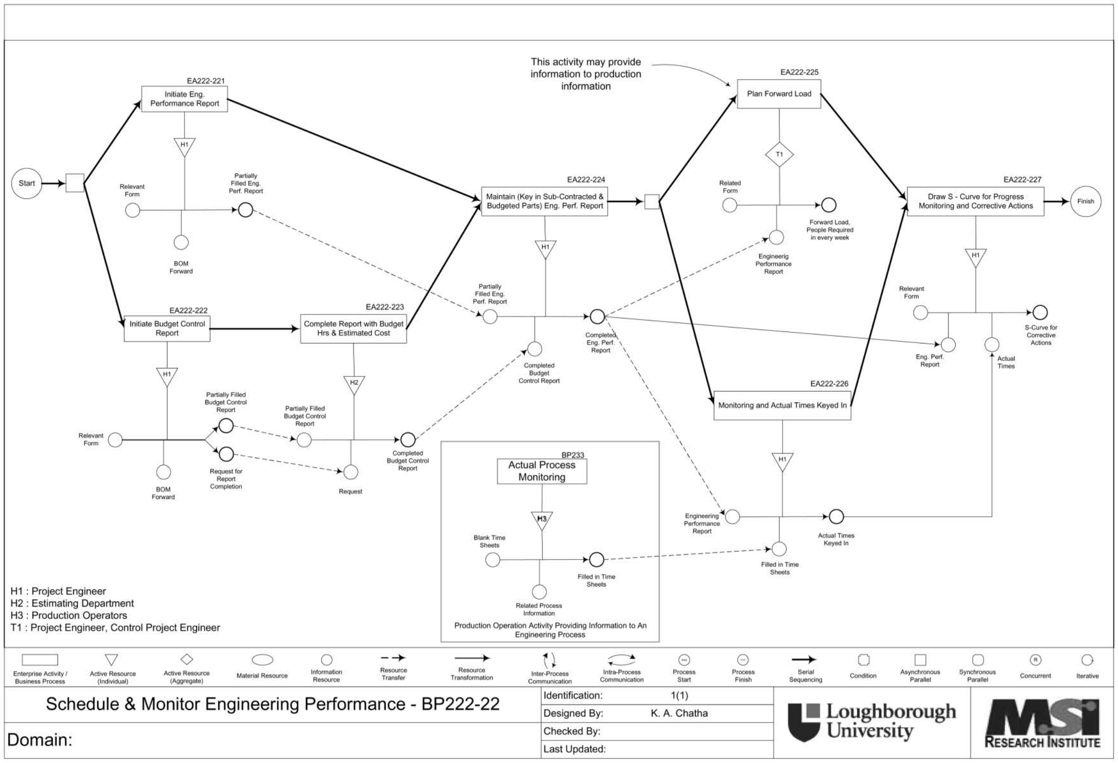

Figure 4. An example static model created using E-MPM activity modelling formalisms.

figure 2. It can be observed that BP222-22 consists of seven enterprise activities that interact within a single business process thread. BP222-22 is itself an 'engineering process' which interacts with a production process (e.g. assemble machine process – BP233). As part of BP233, actual timing sheets are filled in, and data is fed to the 'Schedule & Monitor Engineering Performance Process' for the purpose of scheduling resources, measuring the performance of production operations and further planning. The active resources (individual and grouped) and passive resources (used and processed) required to carry out each enterprise activity and business process were represented using E-MPM process modelling formalism. Key properties of each enterprise activity or business process, are represented either by a transformation line or by means of a transfer line. Transfer of passive resources from one activity to another is also represented by means of a transfer line. Wherever an interaction is necessary it can be represented using inter-process or intra-process communication lines.

4.2. *E-MPM simulation modelling example*

To help systemize the generation of *ithink* simulation models from E-MPM process diagrams the present authors derived formal mappings between common process model building blocks (formed from E-MPM modelling constructs) and equivalent *ithink* model building blocks (formed from *ithink* modelling constructs). It was presumed here that these mappings could enable the process of transforming EMs into equivalent SMs by hiding detailed understandings about *ithink* and E-MPM modelling principles and mechanisms. The mapping formalisms shown in table 1 were determined and tested. Following which they were adhered to when deriving *ithink* model case study process diagrams. To illustrate the principles involved there follows an example case of creating a unitary enterprise activity (EA) model (represented using an *ithink* modelling block) by using mapping formalisms defined by table 1.

The specified EA modelling block is shown in figure 5. Here the example EA ('Assemble Column') is represented by a standard *ithink* 'Oven' modelling construct and other attached *ithink* modelling constructs. Related passive-used resources needed by this specific EA (namely information about Sales Orders, Machine Drawings & Machines Specifications) are encoded by the attached *ithink* 'Converter' modelling constructs, while passive-processed resources and EA outputs (namely Machine with Column Scraped & Machine with Column Assembled) are encoded by standard *ithink* 'Stock' modelling constructs. At this stage of modelling the *ithink* EA modelling block is incomplete because as yet active resources and procured

Table 1. Mapping from process diagramming concepts to *ithink* modelling concepts.

E-MPM Process Diagramming Concepts	***ithink* Modelling Concepts**
Enterprise Activity	Oven
Individual Active Resource	Converter (Standard)
Active Resource (Team)	Converter (sum)
Passive-Processed Resource	Stock, Converter (Standard)
Passive-Used Resource	Converter (Standard)
Transfer of Resource	Connector, Inflow/Outflow
Transformation	Implicit in Oven ---
Start Activity	Converter (Standard)
Simple Sequence	Connector
Condition	Logic created using Converters and Connectors ---
Asynchronous Parallel Activity	Logic created using Converters and Connectors ---
Synchronous Parallel Activity	Logic created using Converters and Connectors ---
Concurrent Activities	Logic created using Converters and Connectors ---
Iterative Activity	Logic created using Converters and Connectors ---
Process Finish Activity	Converter (Standard)
Intra-process communication	Transfer of information = Connector; Collaboration = --- ---
Inter-process communication	Transfer of information = Connector; Collaboration = --- ---

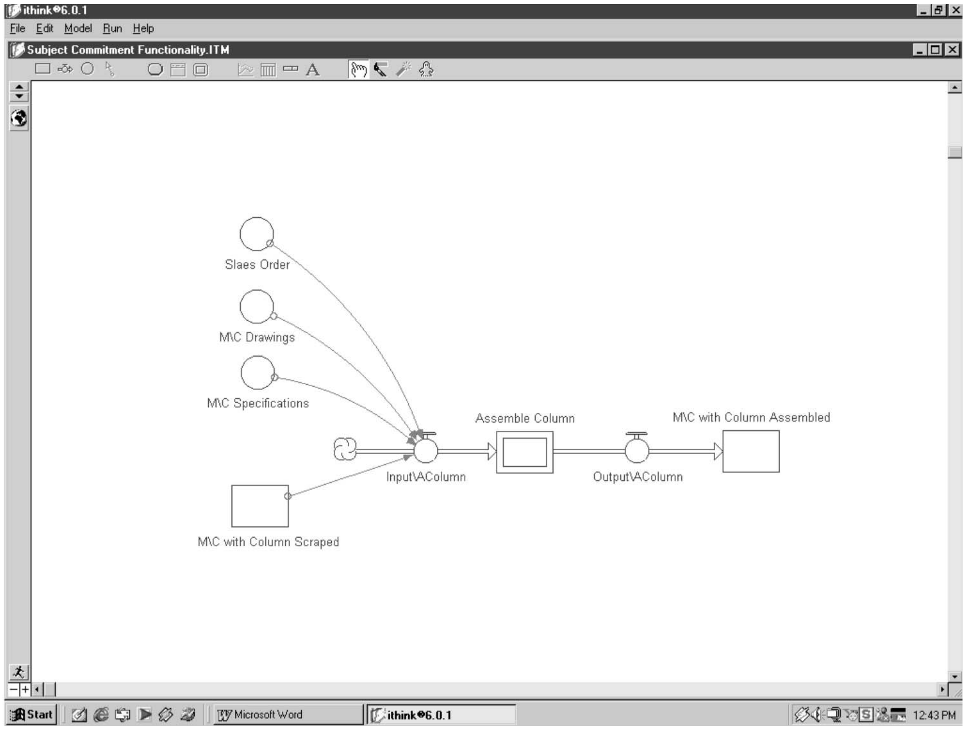

Figure 5. An example enterprise activity modelled in *ithink*.

passive-processed resources associated with this specific EA have not been represented. EA resource requirements will have previously been defined when building the equivalent process model of the EA; in terms of the functions the EA should achieve and related performance attributes and operating constraints. The defined mapping dictates that these resource requirements be represented by a standard *ithink* 'Array-Variable'; which in figure 6 is named 'Required AR'. Use of the array variable offers means of reducing the visual complexity of *ithink* models and is particularly useful where there are repetitive model structures. *Ithink* array variables may have one or two 'dimensions' and each dimension can have a number of 'elements' related to it. By relating modelling concepts present in array variables to attributes of an EA, it has proven effective to define EA active resource requirements and to represent them as one single dimensional *ithink* array variable, where this dimension can have three elements in it. Those elements are used to define functions, performance attributes, and constraints as shown in top left corner of figure 7. Array element values are entered in numerals.

Similarly active resources associated with a unitary *ithink* EA are modelled from three viewpoints, where these viewpoints are represented by a standard 'Array-Variable'. In the example case shown in figure 6 the array named 'Provided AR' has three elements to which numerical values are attributed. Subsequently EA requirements may be checked against characteristics of available resources in the modelled system by comparing 'Required AR' numerals with 'Provided AR' numerals. For this particular purpose *ithink's* communication functionality has been exploited, which is designed to allow data communication with other applications in a modelled system. The most appropriate choice of active resource (with respect to specified EA requirements) may also be encoded by the 'Provided AR' array-variable. Attribution of this most appropriate AR is conditional on its ability to fulfil defined EA requirements and if this condition is met it may commit itself to the EA. A similar procedure is used for passive-processed resources which are represented by 'Required Object', 'Provided Object' and 'Committed Object' types of array-variable.

Besides defining functional requirements of resources the example unitary *ithink* EA model also sets time constraints on the commitment of resources. For example an EA may need to be completed in a specific time period and before some deadline. These requirements may be represented using standard *ithink* converters, as shown in figure 7.

In figure 6 the 'Time Required of AR' is defined by a 'Standard *ithink* Converter'. This enumerates and represents time requirements of the example EA, i.e. 'Assemble Column'. The time period for which an active resource is required is computed using two variables, namely 'tr\l' and

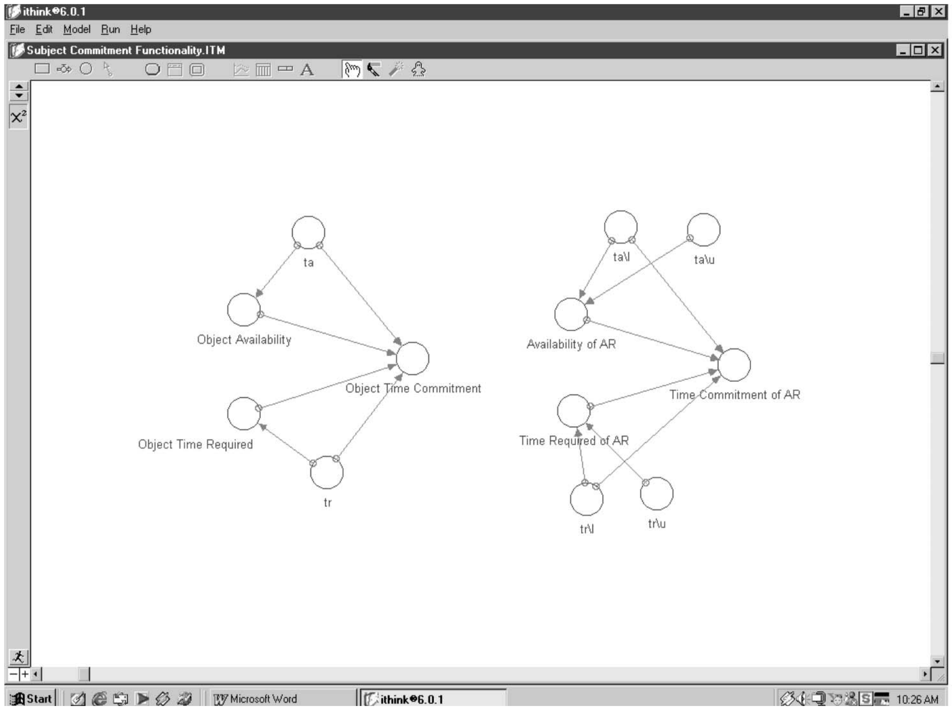

Figure 6. Active resource (AR) commitment.

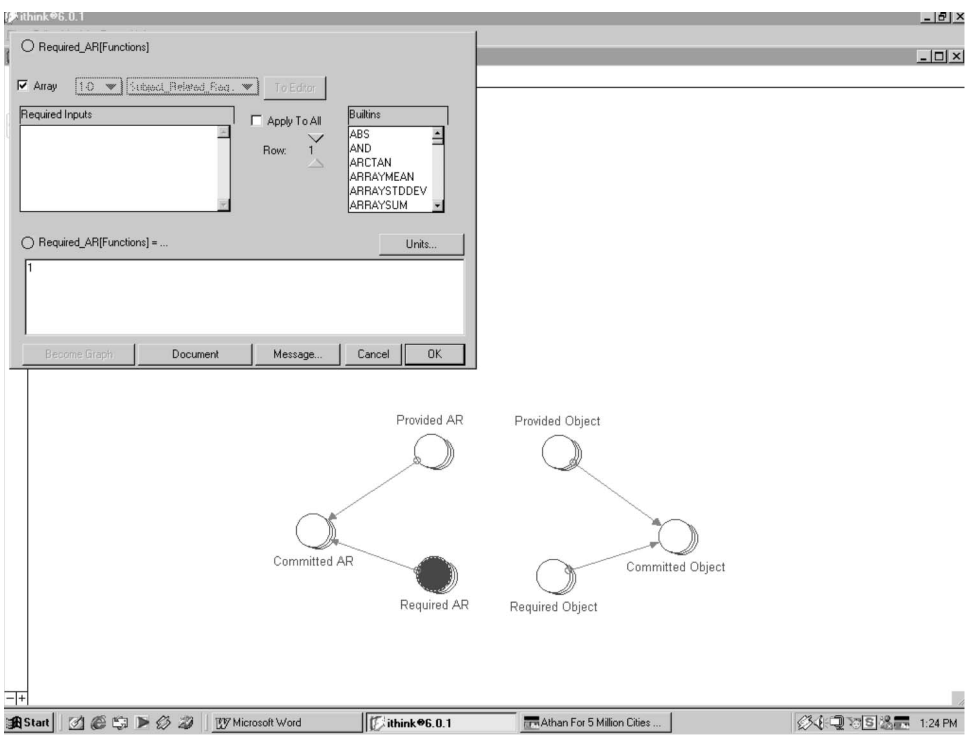

Figure 7. Time commitment of active and passive (objects) resources.

'tr\u'. 'tr\u' represents the 'upper limit of time required', i.e. the time by which an EA should be completed. This time may be entered as a numerical value representing hours, days, weeks or months, as defined by other simulation parameters input to the *ithink* tool. Similarly, 'tr\l' can represent the 'lower limit of time required' or the time by which an EA should have started for it to be able to be completed by the defined deadline. This time is also entered as a numerical value. In this way time bounds can be placed on active resources as they carry out EAs. Active resources also have availabilities that (like EAs) can be modelled using standard *ithink* converters. In figure 6 the 'Availability of AR' converter represents the duration for which a resource is available. This time period is computed using variables 'ta\l' and 'ta\u' in a similar way to which the required time period is computed where 'ta\l' is lower limit of time on which a resource is available, whereas 'ta\u' is upper limit for which a resource is available. A match between 'Time Required of AR' and 'Availability of AR' can result in the commitment of a resource to an EA. The outcome of this comparison, and decision to commit, is represented by the 'Commitment of AR'. Similar logic and model elements can be deployed to model the commitment of passive resources or objects to EAs. The only notable difference is that in the case of Objects there may be no need to define 'tr\u' or 'ta\u' limits.

When appropriate active and passive resources (i.e. resources that meet EA requirements) have been selected and are committed, they enable progression of the EA, in the example case the 'Assemble Column' activity. This commitment of resources is signalled via a 'connector' in the enterprise activity model in the manner shown in figure 8.

It should be noted that the graphical representation of activity requirements, resource availabilities and their subsequent matching and commitment to EAs make the resultant *ithink* model complex in view. However the matching and committing modelling structures are an essential part of each EA because of the approach chosen to model EAs in *ithink*. Another way of looking at this is that each EA can be considered to comprise a module of an *ithink* process model, because in general any process model will be composed of some organized set of EAs and therefore some organized set of EA modular building blocks. A resultant outcome is that if the full details of each EA module are shown it makes *ithink* process models difficult to understand and interpret. Therefore, the modeller may choose to hide details about resource selection and commitment associated with EAs. This hiding can be realized by exploiting the 'Space Compression Object' facility of the *ithink* software tool.

The Space Compression Object facility, as the name suggests, compresses the graphical representational space of a model by hiding the model complexity and its underlying detailed logic, by maintaining and executing the full model but only providing an abstract graphical representation of the outcomes (see *ithink* analyst-technical documentation 1996). From a representational standpoint, Space Compression Objects have two states, 'Open' and 'Compressed'. In the open state a rectangular space is

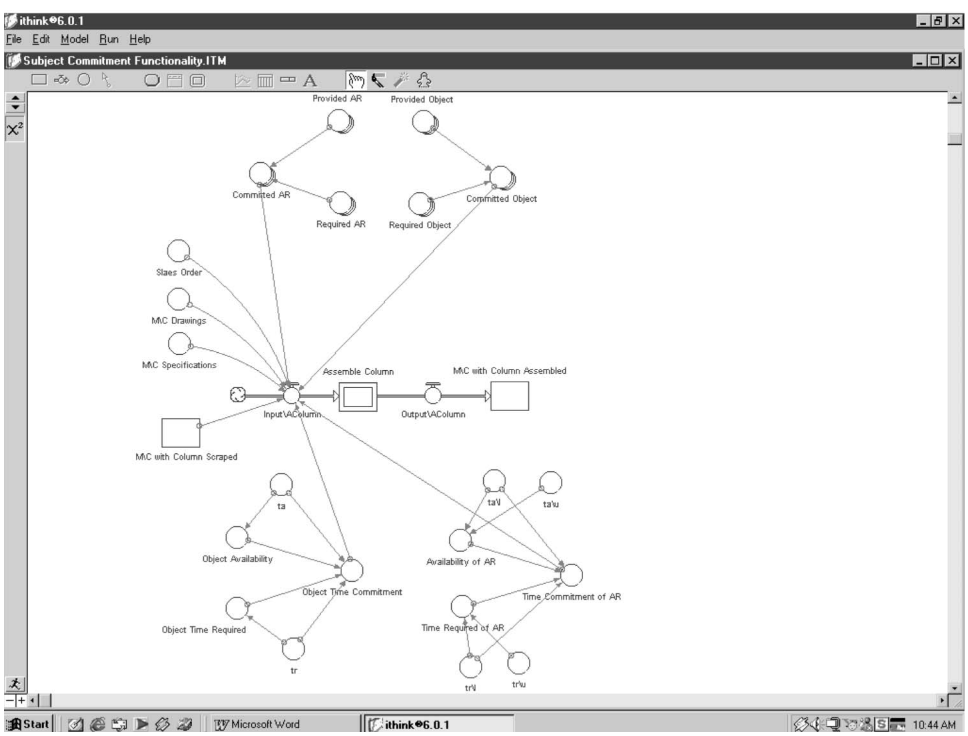

Figure 8. Active and passive resources fed to enterprise activity.

presented which is used to display part of the model details that can be hidden. When compressed the rectangular box disappears, but a small double-lined box remains visible which shows that some module of the model exists in greater detail than will be managed by the *ithink* tool. An example Space Compression Object is shown in figure 9 in both compressed and uncompressed states. Comparison between figures 10 and 11, which are two graphical representations of the same *ithink* model, shows the difference achieved in representational simplicity and understandability as a consequence of using the Space Compression Object facility.

Similar steps were needed to transform process diagram model blocks to *ithink* model blocks to those previously used to transform activity diagrams into equivalent *ithink* simulation models. However the combined use of E-MPM process diagram and the mapping formalisms shown in table 1 demonstrated much enhanced capabilities (relative to the use of activity diagrams) because: it systemized EM to SM transformations; it led to the creation of modular simulation models that were known to operate effectively; it helped systemize the design and capture of key (dynamic and static) simulation model attributes and parameter values; and it minimized the need to know about modelling fundamentals and specifics mapping between governing EM to SM models and EM to SM tools, respectively. Because of the foregoing relative to MPM, E-MPM enabled better (i.e. fit for multiple simulation purposes) and faster SM modelling whilst maintaining key information about the context in which simulated process segments would be used.

E-MPM's process modelling language, and modular approach to converting process diagrams into *ithink* simulation models has proven effective with respect to investigating dynamic behaviours of business processes used by the case study vendor of automotive production machines. Figure 10 illustrates such a model segment. This has two sector frames corresponding to two 'loosely coupled' process threads encoded by the process diagram of figure 4. One of these sector frames relates to the 'Schedule & Monitor Engineering Performance' process as a whole (labelled BP222-22), while the other sector frame represents the 'Actual Process Monitoring' (BP233) carried out as part of the 'Production Operation' domain process. Interaction between these two process threads is represented by an *ithink* 'Connector' model building block, which transfers necessary information between the 'Actual Process Monitoring' and the 'Actual Process Times Keyed In' process threads.

On taking a closer look at these interacting process threads it can be observed in figure 11 that resource attributions flowing between enterprise activities can be modelled by means of *ithink* 'Connectors' and 'inflows'. Here the simulation capabilities of the *ithink* software

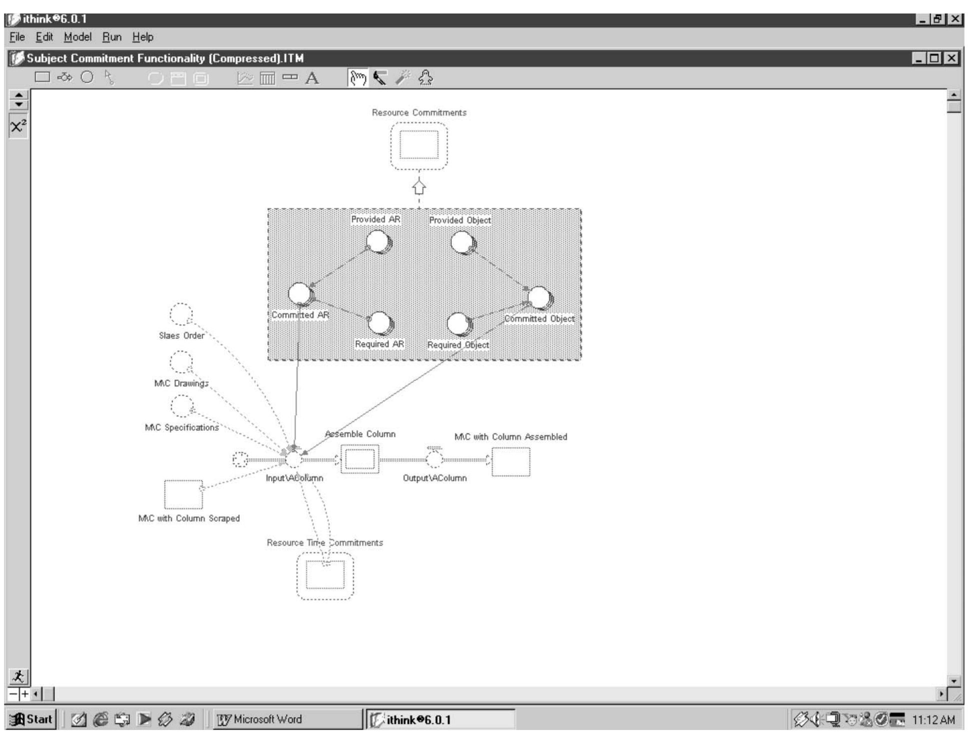

Figure 9. Space compression object hiding model complexity.

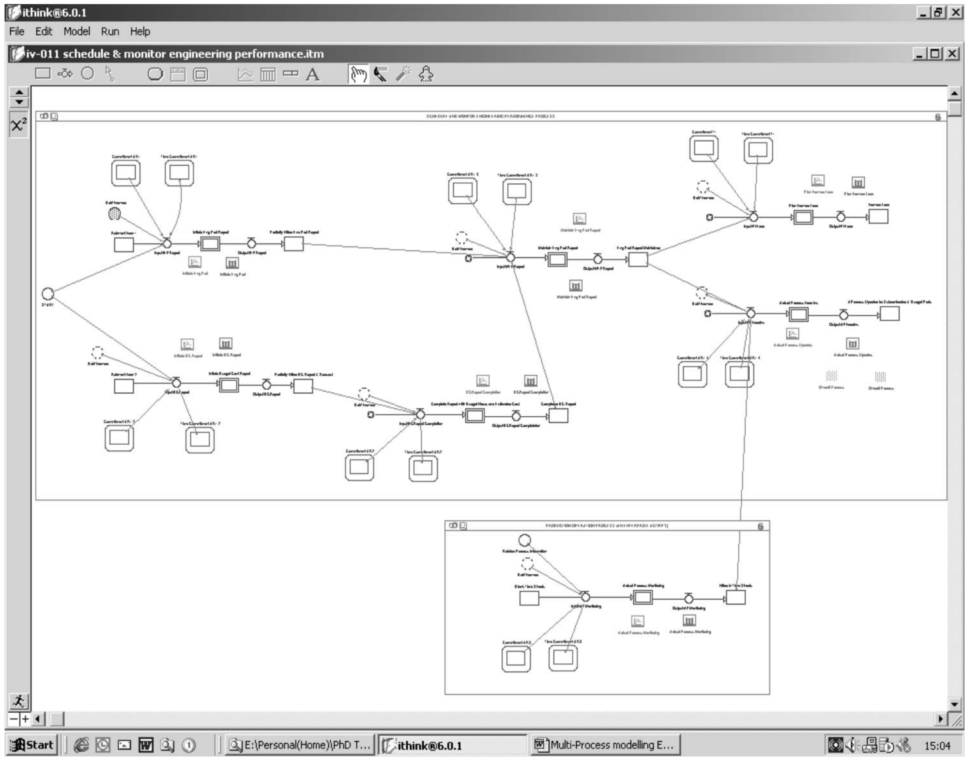

Figure 10. E-MPM simulation model generated for the case study 'schedule & monitor engineering performance' process segment.

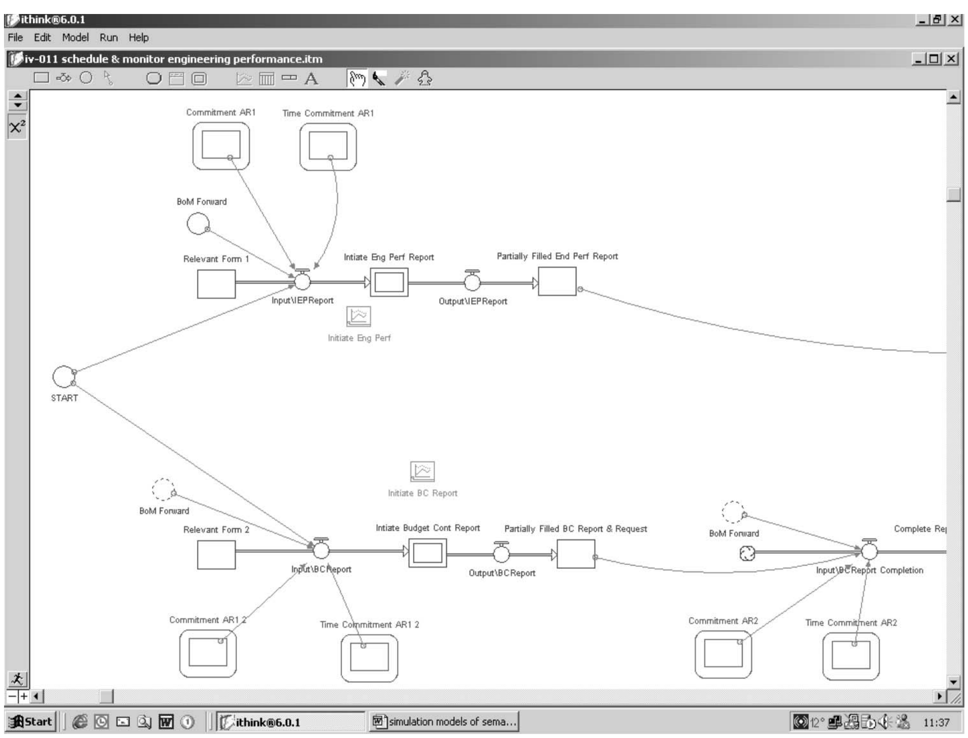

Figure 11. Example enterprise activities involved in the schedule & monitor engineering performance process segment.

allows dynamic behaviours of EAs to be modelled separately or as an integral part of two interacting process threads. Consider for example the 'Initiate Engineering Performance Report' EA (i.e. EA222-221).

Figure 12 shows a graphical representation of timings associated with this EA. When the simulation model is run the *ithink* tool is programmed to display: start and stop times for the EA; the time period for which the EA remains active; and the outputs generated during EA execution. The commitment of resources to this EA can also be modelled graphically, showing the period of time for which selected resources need to be available to execute an EA and the period for which a selected resource actually becomes available. When these two variables are matched, the period for which a resource is committed to the EA, and the period for which resource remains idle can also be computed in the manner depicted by figure 13.

In this specific example case the different equations which determine dynamic behaviour of the process in *ithink* tool are provided in table 2. For other activities similar variables and equations apply but with different values. Of course a significant part of the *ithink* model creation process involved establishing and testing the validity of underlying dynamic equations.

Ithink model results of the form illustrated by figures 14 and 15 were also found to be useful when analysing and predicting process times in order to optimize process designs. The overall process time can be represented in the form shown in figure 14. Also the schedule of individual resources can be represented in the manner shown in figure 15.

5. General observations

This paper explains that in principle significant synergy can be gained from using complementary enterprise modelling, simulation modelling and workflow modelling techniques. It also explains that simulation models can be created which exercise dynamic properties of process segments, which form modular, dependent building blocks of MEs, and that these properties can be investigated with reference to the specific context in which a given process segment will

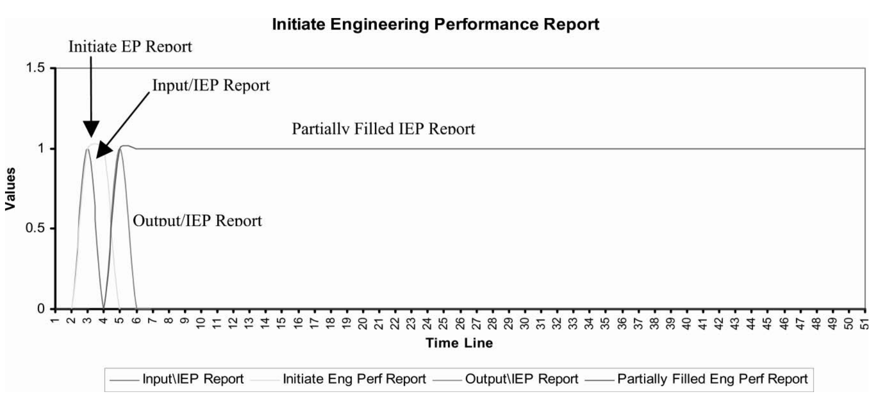

Figure 12. 'Initiate engineering performance report', example EA timings.

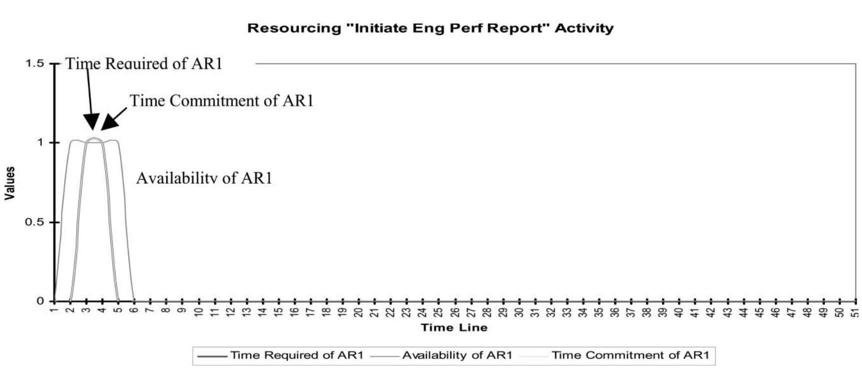

Figure 13. 'Initiate engineering performance report' with associated resource timings.

Table 2. Example equations used at different simulation modelling blocks at one enterprise activity.

Required_AR1[Function] = 1, **Provision_AR1[Function]** = 1,
Commitment_AR1 = IF(Provision_AR1[Functions] > Required_AR1[Functions]) THEN 1 ELSE 0
Required_AR1[Performance Attributes] = 1, **Provision_AR1[Performance Attributes]** = 1,
Commitment_AR1[Performance Attributes] = IF(Provision_AR1[Performance_Attributes] > Required_AR1[Performance_Attributes]) THEN 1 ELSE 0
Required_AR1[Constraints] = 1, **Provision_AR1[Constraints]** = 1,
Commitment_AR1[Constraints] = IF(Provision_AR1[Constraints] > Required_AR1[Constraints]) THEN 1 ELSE 0
tr\l = 1, **tr\u** = 4, **Time_Required_of_AR1** = IF(TIME > tr\l AND TIME < tr\u) THEN 1 ELSE 0
ta\l = 0.5, **ta\u** = 4, **Availability_AR1** = IF(TIME > ta\l AND TIME < ta\u) THEN 1 ELSE 0
Time_Commitment_AR1 = IF (Time_Required_of_AR1 = 1 AND Availability_AR1 = 1 AND (ta\l = tr\l OR ta\l < tr\l)) THEN 1 ELSE 0
Input\IEP Report = IF (START = 1 AND BoM_Forward = 1 AND Commitment_AR1[Functions] = 1 AND Commitment_AR1[Performance_Attributes] = 1 AND Commitment_AR1[Constraints] = 1 AND Time_Commitment_AR1 = 1) THEN PULSE(1) ELSE 0
Initiate_Eng_Perf_Report:
Capacity = 1, **Fill_Time** = INF, **Cook_Time** = 2.5, **INITIAL** = 0
Output\IEP Report = 0
Partially_Filled_Eng_Perf_Report: **INITIAL** = 0

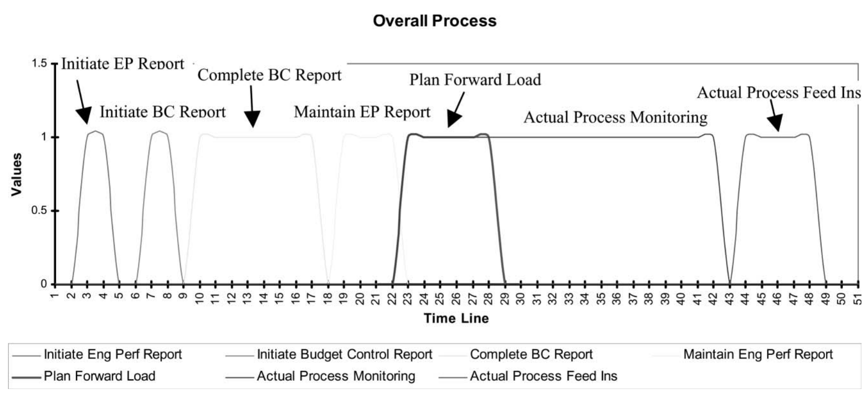

Figure 14. 'Schedule & monitor engineering performance' process time.

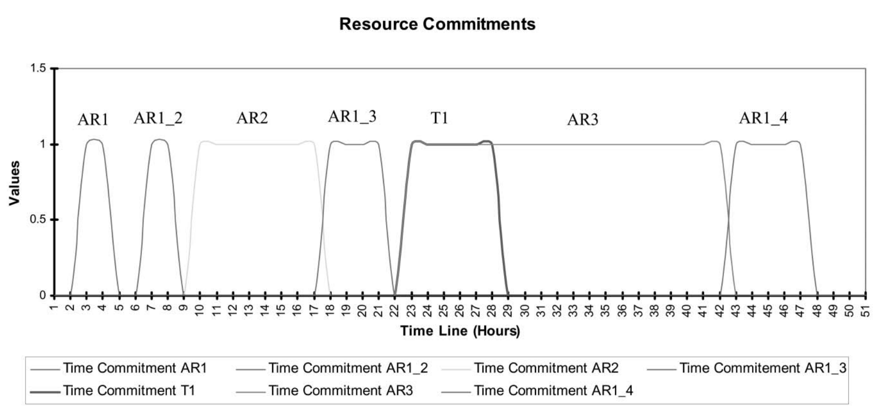

Figure 15. Resources' schedules for the 'schedule & monitor engineering performance' process thread.

be used. To illustrate the arguments a case study problem domain is described which concerns a very complex process network used by a consortium of companies making automotive products on a global scale.

Bearing in mind potential opportunities to improve upon current best practice when creating and reusing process maps, improved multi-process modelling approaches were conceived. Two stages of that development are represented by MPM and E-MPM. Essentially the former approach left the modeller to find ad hoc ways of transforming enterprise models (EMs) into equivalent simulation models (SMs). Whilst E-MPM provides improved means of structuring and enabling EM to SM transformations. The paper describes how E-MPM was designed to meet this purpose and illustrates a case study use of its model structures and enabling mechanisms.

The reader can deduce from the case study illustrations that the support for process simulation provided by E-MPM is more extensive than that provided by its predecessor approach MPM. E-MPM takes into consideration certain requirements of process designers who need to utilize resource related information (captured from knowledge holders/sources in an organization) in order to make more informed choice from amongst resource candidates. Thereby during process design improved analysis and prediction of process lead-times can be achieved. Also relative to predecessor EM approaches, E-MPM provides an enhanced capability to record resource commitments which can directly be used for resource scheduling and also to establish the cost of resourcing process threads. Indeed commonly process designers must take into considerations not only resource capabilities but also their schedules as well as requirements of processes. In ongoing UK Government (EPSRC) funded research E-MPM principles are being applied in support of ongoing organization design and change in a number of MEs of distinctive types and sizes.

References

Chatha, K.A., Multi-Process Modelling Approach to Complex Organisation Design, Wolfson School of Mechanical and Manufacturing Engineering, Loughborough, University, 2004.

Chatha, K.A., Weston, R.H. and Monfared, R.P., An Approach to modelling dependencies linking engineering processes, Proc. of Inst. of Mech. Eng., Part B: *Journal of Engineering Manufacture*, 2003, **217**, 669–687.

CIMOSA, 'CIMOSA: Open System Architecture for CIM', AMICE-ESPRIT Consortium, 2nd Ed. Vol. 1. 1993, (Berlin: Springer-Verlag).

Forrester, J.W., Principles of Systems: Text and Workbook, Wright-Allen, 1968, ISBN: w7735609.

GERAM. Generalised Enterprise Reference Architecture and Methodology, Version 1.6.3 (March 1999). IFIP–IFAC Task Force. http://www.cit.gu.edu.au/~bernus/taskforce/geram/versions/geram1-6-3/v1.6.3.html. 1999.

Harrison, R, West, A.A., Weston, R.H. and Monfared, R.P., Distributed Engineering of Manufacturing Machines, Journal of Engineering Manufacture, Part B2, Vol. 215, pp 217–231, March 2001.

Ithink: Software Version 7, (High Performance Systems, Inc., Hanover, New Hampshire); http://www.hps-inc.com., 2002.

Ithink Analyst Technical Documentation. High Performance Systems, Inc., 1996.

ICAM. 'US Airforce Integrated Computer Aided Manufacturing (ICAM) Architecture' Part II. Volume IV – Functional Modelling (IDEF0). Air Force Material Lab., Wright Patterson AFB, Ohio 45433, AFWAL-tr-81-4023, 1981a.

ICAM. 'US Airforce Integrated Computer Aided Manufacturing (ICAM) Architecture' Part II. Volume V – Information Modelling (IDEF1). Air Force Material Lab., Wright Patterson AFB, Ohio 45433, AFWAL-tr-81-4023, 1981b.

Spur, G., Mertins, K. and Jochem, R., *Integrated Enterprise Modelling*, 1996 (Berlin: Beuth Verlag).

Vernadat, F.B., *Enterprise Modelling and Integration: Principles and Applications*, 1996 (Chapman & Hall, London).