A framework for connecting work and information infrastructure

J. Goossenaerts
Eindhoven University of Technology, Information & Technology
P.O.Box 513, Paviljoen, 5600 MB Eindhoven, The Netherlands
email: jgo@tm.tue.nl

Abstract

An epistemological framework is defined which unifies information and command requirements of autonomous agents as they involve in enterprises and the life spans of artifacts. The framework assumes a division of the manufacturing industry domain into two sub-domains and four activity-layers. The sub-domains are the cybernetic domain and the physical domain. The activity layers cover observations, operations, improvements and innovations. The linguistic primitives records, proxies, versions and modules and the generic services of browser&report generator, model execution engine, version manager and innovation coach are assigned to the respective activity layers. Faithfulness conditions are defined between (successive) situations and events in the physical domain and corresponding constructions and transformations in the cybernetic domain. Mechanisms and principles for connecting work and information infrastructure are explained.

Keywords

Information infrastructure, CALS, framework for enterprise integration, artifact life phase service body, life cycle modelling.

1 INTRODUCTION

As computers worldwide get connected in a global network and industries move towards an extended enterprise mode of production and development (Browne *et al.*, 1995) and learn to cope with the full life cycle of goods and artifacts (Krause and Kind, 1996), it is expected that information infrastructure services will play an increasing role for the exchange of technical and business data, and for the distributed development and control of business processes.

J. Goossenaerts et al. (eds.), *Information Infrastructure Systems for Manufacturing*

One can wonder how future information infrastructures for industries will look like? What services will these systems offer to enterprises, consumers and public bodies? And how will these services be realized? The MiViPoRo framework (Modules for innovations, Versions for improvements, Proxies for operations, Records for observations) presented in this paper intends to offer guidance in answering these questions. It results from a technology-independent reflection on how to offer information services for citizens and companies as they involve in artifact lives and business processes. A fundamental issue here is that the micro situation in which consumers, companies, or public bodies "work" with goods and artifacts must be linked to information, captured knowledge and computing power that is becoming globally available. An understanding of the mechanisms and principles on which such a connection can be established is necessary for creating and exploiting an information infrastructure.

The Problem Domain

In manufacturing industry an increasing number of players is becoming involved in the design, manufacture, distribution, servicing, collection at end of life, disassembly, recovery or recycling and ultimately safe disposal of artifacts and goods. These players, including companies, consumers and public bodies, have their own values and working modes. They seek to effectively meet a mix of private and public, environmental, socio-economic and sustainability challenges. Their cooperation and fair competition can be enabled by information infrastructure services for sharing information and common methods and telematics applications for order passing, contract negotiation, and cooperation in life cycle oriented product and service development.

Usage of the Framework

The MiViPoRo framework serves a double purpose. On the one hand it guides the requirements definition and development of the generic system services for an information infrastructure for manufacturing industry. The generic services are innovation coaches, version managers, secure model execution engines, and browsers and report generators. They correspond to the activity layers for innovations, improvements and operations which have been consolidated in production management practice (Inagaki, 1993). In addition the framework offers guidance for planning the future development of artifact life span oriented applications for manufacturing industries. It shows opportunities for sharing applications and information.

Related and Future Work

(a) The infrastructure concept meets requirements of the CALS (Continuous Acquistion Life-cycle Support) strategy for sharing integrated digital product data. An Integrated Data Environment (IDE) has been proposed as an end-state of the *Electronic Commerce/CALS and Electronic Data Interchange* (EC/CALS & EDI) vision (OSD-CALS, 1996). IDE services can be defined in the MiViPoRo framework.

(b) Regarding the capture of user requirements, the MiViPoRo framework draws on the ENV 40 003 framework for enterprise modelling (1990). The relation between the cybernetic and the physical domain has also been denoted by the term interflow as defined in (ENV 12 204, 1995). The construction of a generic plant interflow model (Goossenaerts

& Bjørner, 1994) focusses on operations inside a single production system as they pertain to a single phase – production – in the life of an artifact. The MiViPoRo framework provides a basis for expanding that construction to cater also for networks of production systems, consumers and public bodies, and for services spanning the possible lives of artifacts.

(c) The MiViPoRo framework treats only the generic aspects of a framework for connecting work and information infrastructure. In order to cater for manufacturing in all its variety and particularity, the framework should be extended and applied. Artifact (life phase) models should be extended by drawing on progress in product modelling (Krause *et al.*, 1993) and the related STEP standards (Gielingh, 1993). Process models should be enriched by drawing on CIMOSA (AMICE, 1993).

(d) A detailed study of the transformations of possible lives models, and the manner in which an innovation coach (preferably supporting a concurrent engineering mode of design) could support it, could be based on the General Design Theory (Yoshikawa, 1981), as indicated elsewhere (Goossenaerts *et al.* (1996)). Innovation and improvement layer transformations and the realization of new "interflow" systems in enterprises – business reengineering – on the basis of new or revised modules are part of enterprise life cycles (Williams, 1994).

(e) For its operation layer, the framework assumes mechanisms for the computer network based execution of enterprise and artifact possible lives models. The concepts of interflow system and the execution mechanism for enterprise formulae (Goossenaerts, 1993) offer some relevant insights. Other inputs are expected from advances in system interoperability based on object-oriented interfaces and the developments on open distributed processing (ISO/IEC DIS 10746-3).

2 THE MiViPoRo FRAMEWORK

An information infrastructure system for manufacturing industry is a large and complex system without centralized control. It supports and enables industrial processes that are purposeful (answering the why question), proceed in time (when), are situated in chains of cells (where), depend on the initiative and effort of agents (who), and transform or use artifacts and materials (what). These processes show forms of cooperation between several players having their own independent interests, values and modes of operation with respect to artifacts, goods and services. The MiViPoRo framework supports straightforward and infrastructure-wide reasoning and computation on transformations, queries and answers involving purpose, (points and intervals in) time, places, agents and artifacts and their attributes, as knowns and unknowns.

The present description of the framework looks at the linguistic primitives that are typical for the different activity layers, and the faithfulness conditions on the physical and cybernetic domain that are defined using them. The meta-linguistic connections between different primitives are not covered in detail.

2.1 MiViPoRo , its Generic Services & Linguistic Primitives

The MiViPoRo framework divides the problem domain of manufacturing industry into two sub-domains and four activity layers (Figure 1). The sub-domains are: the *physical domain* comprising the physical space, time and matter, with artifacts, goods, agents and cells (spatial units) having their lives in it; and the *cybernetic domain* which adds memory, communication, monitoring (including knowledge-capture) and control (computations) services to the physical domain. The term *interflow* (short for *inter*active *flow*) denotes the coordination and monitoring of time&space&matter situated physical processes by means of computational processes. The concept of physical process is wide and includes processes which are (usually) not modelled (such as for instance the thought processes of individuals who work in improvement or innovation layer). The concept of a computational process is kept narrow, it covers only transformation or storage of digital representations as they are processed by computer programs.

The four activity layers span the two sub-domains and cover observations, operations, improvements and innovations. In each of these layers, work – action in the physical domain – has to be connected to computations and communications in the cybernetic domain. This is done by the generic services: *innovation coach, version manager, model execution engine* and *browser&report-generator.* These services provide interfaces between objects existing on the physical and cybernetic sides of the activity layers.

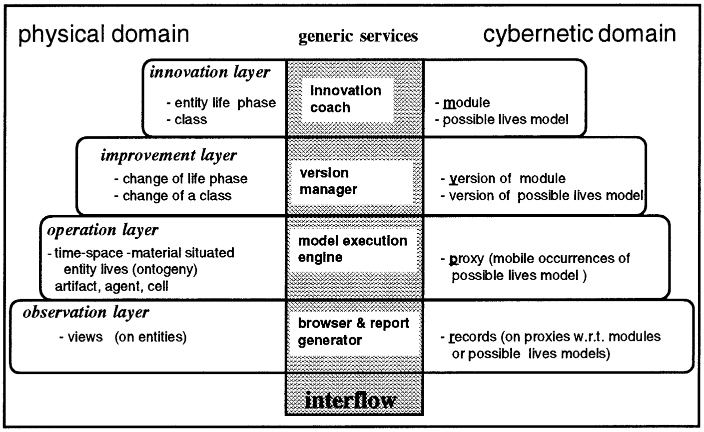

Figure 1. Activity layers, generic services and linguistic primitives in MiViPoRo.

On the physical side, the term *entity* denotes an artifact, agent (person, machine), or place which has a definite physical existence in itself. At any point in time, an entity is present at exactly one place. It is characterized by an existence in matter and a mobility in space (except for some places relative to each other). An agent has powers which determine the actions it can execute. Entities are grouped in *classes* on the basis of similarities in their time&space&matter situated possible lives. At a point in time, the place where an entity is, will accommodate certain possible or required life phases of the entity. This set of life phases is called the *local type* of the entity and determines the next events in which the entity may involve. An entity has a life span, a succession of space-time situated events in which other entities may be involved. The (local) type of an entity will change during its life time.

The linguistic primitives on the cybernetic side are: (i) *modules* are created and transformed in the innovation layer. One module typically captures time-space-matter independent aspects (knowledge) on one possible or required phase of the life span of a class of entities. Modules are combined into a *possible lives model* of the class. (ii) *Versions* of modules and possible lives models capture modifications as they are performed in the improvement layer. (iii) A *proxy* exists in the cybernetic domain as a unique representant of a physical entity. It captures knowledge on the entity and the modules (each belonging to the possible lives model of the class of the entity) for which it has been earmarked. Proxies exist at the operation layer. (iv) A *record* on a proxy is created in memory of the proxy's past or current flow through the cybernetic domain, in reflection of the time-space-matter situated life span of the corresponding entity. Records are recorded during operation layer activities, for further use in observations.

The activity layers, generic services and primitives provide us with a basis to specify common arrangements concerning artifact lives and the interaction between processes in the physical domain and processes in the cybernetic domain.

Simplifying a bit, one could state that in the physical domain, each player has a territory where its entities exist, or where its modules prevail. An example of the first case is the citizen who keeps belongings in a house or locker. The latter case is illustrated by a public authority who issues the rule that security belts should be present on the seats in motorcars, and defines conformance tests for this rule. The modules managed at the public body constrain the outlook and usage of cars. Similarly, each player owns a fraction of the MiViPoRo-conformant "cyberspace". These are the own proxy and the proxies of the entities that are owned by or entrusted to him or her. The cybernetic domain of the manufacturing company includes modules of the production processes it can sustain. It also includes proxies of the machines, products in stock and in production, and performs the production planning and control functions. The cybernetic fraction of a public body could include modules specifying rules and procedures to enforce them, as well as files and registers listing records on the subjected entities (and proxies) in a territory.

2.2 Mi-:Modules for Innovation

Innovation processes use modules for the time&space&matter-independent modelling of cells, artifacts and agents. Modules are the first deliverables of innovation processes. A module describes a possible or necessary phase in the possible lives of one or more artifacts, agents, and/or cells. In an industrial context where the use of product and process models is mature (e.g., one uses digital mock-ups of products and process models) and material and work are expensive, innovation processes will create and evaluate modules first, before physical entities are produced, or processes are carried out in reference to the modules. An innovation coach is a generic service which supports design, for instance in a concurrent engineering style, with global sourcing of design data (modules) and artifact data (proxies and records).

Definitions: *(i)* A *module* is a space&time&matter independent description of a possible or required phase in the life span of one or more entities. It describes what one should know to handle an entity in a certain situation. If a place has been equiped to accommodate a life phase of an artifact (e.g., an assembly line is equiped to produce a type of car) then the place is said to be earmarked for the life phase. On the cybernetic side the place's proxy is earmarked for the module describing the life phase. If an entity enters the place (e.g., a car enters the repair shop) earmarked for one of its possible life phases, then the entity's proxy will be stored at the cell proxy, and the module's definition will determine the activities for the entity.

(ii) A *possible lives model* is a set of modules which have been defined for a class of entities, or in which the entities of the class can be involved (when a module describes a phase in which entities of different classes are in a relationship). Modules may describe concurrent, successive, alternative or mutually exclusive phases in the possible lives of entities. For instance, the possible lives model of a product could include modules for the phases: need evaluation (prefeasibility, feasibility, project definition), design & development, production, transportation, usage, maintenance (preparation, repair, modification, supply), and dis-engagement.

(iii.a) The *capability* of a cell is a set of life phases which are accommodated at the cell. It is described by modules in the cell's proxy. In principle there is no limit on the number of modules in a cell proxy (a garage may be equiped to repair cars of several brands and types).

(iii.b) The *type* of an entity is the time&space dependent set of the artifact life phases in which the entity is participating. It is described by all modules for which the entity is earmarked. In principle there is no limit on the number of modules in the type of an entity. The type of an entity may change as time proceeds and the entity moves through the physical domain. E.g. a person who is citizen of one country, becomes a visitor of another country when crossing its border. While being entitled to engage in a labour contract in the own country, the person may be excluded from this right in the other.

(iv) The term *typogeny* (combining type and -geny) denotes the origin, development and evolution of a type.

(v) The term *typology* denotes the theory or study of types and typogenies. It addresses the relations between typogenies, and their module-wise changes as new versions of artifact types, agent types and cell types are developed.

Innovation depends on the transformation and appraisal (to select among alternatives) of the modules which are part of possible lives models. New modules must be better or enable new functions while respecting boundary constraints (technology, economy, regulations, mechanisms of nature).

Innovation coach systems can support innovation by allowing one to unify modules (from registers, own or subcontractor warehouses or servers) which are relevant for innovation in a certain life phase (production, maintenance, use, disposal). The design engineer can then replace, refine or improve the module. The coach should offer specific functions to evaluate proposals w.r.t. objective criteria, such as conditions for production, transportation, distribution and recycling.

A possible formalism to express modules is that of enterprise formulae as proposed by the author (1993). Enterprise formulae feature high information density, modular structure and strong context integration. This formalism allows one to combine in a single modular run-time independent expression, the three views that are commonly used to reduce the complexity of enterprise process models, i.e. the data view, the function view and the organization view (see Scheer, 1994, pp 10-13).

2.3 -Vi-: Versions for Improvement

Improvement processes develop new versions of existing modules, proxies or possible lives models. A version manager supports modifications while guarding compatibility among the modules and the proxies created in reference to them.

2.4 -Po-: Proxies for Operation

Operations comprise agent-artifact-infrastructure activities which are situated in the physical and cybernetic time&space domain and which create and transform entities and corresponding proxies (and records). Entities are occurrences - exemplars or instances - of the classes, their proxies progress within the constraints set by a possible lives models. Activities may require the retrieval, glueing together and transformation of information (on proxies, modules, records) in the cybernetic domain, or may be driven by it. Typically, the retrieved information will underpin a decision. Transformations will update data kept in accordance with constraints and possible behaviours defined in modules.

Definitions: *(vi)* A *proxy* or *information proxy* is a time-space situated linguistic object which exists in the cybernetic domain and is given the exclusive role to represent a unique current or future time-space-matter situated entity of the physical domain. An *artifact proxy* represents an artifact, an *agent proxy* represents an agent or (legal) person and a *cell proxy* represents a cell (or spatial unit). For each entity there is at most one proxy, by definition. Proxies and the corresponding entities exist independently from the life phases (modules) for which they can be earmarked, and unmarked (the type of the proxies can change). They can have a local relevance (e.g., in a production system, a part's proxy may be removed after it has been assembled in an assembly). A proxy may hold a reference to the modules for which it has been earmarked. A cell proxy that holds a module will have records on the proxies that are earmarked for the module.

(vii) The term *ontogeny* (combining onto- (ens) and -geny) denotes the origin, development and evolution of a single entity. Ontogeny denotes the progression of an entity as it is earmarked for certain life phases and unmarked. At any point in time and place, an entity may be committed for a number of life phases. This is reflected in the cybernetic domain by the proxy being earmarked for the corresponding modules.

(viii) The term *ontology* denotes the theory or study of ontogenies, i.e. of those properties - having an origin, development and evolution - that all entities have in common.

In the context of a manufacturing system, the *materialization process* (production) will map a proxy of a future entity to the entity; this process is guided by the (production) modules in the possible lives model, and executed at cells applying their capabilities. A proxy of a future entity may be created in response to an order, the production process is executed in accordance with the explosion of the order as described by Goossenaerts and Bjørner, 1994.

Possibilities for the shared use of modules are illustrated: The possible lives model *car* includes the models *Xcar* (car of brand X), as well as *Ycar* (car of brand Y). *Xcar* and *Ycar* differ from each other because they are produced according to different *production* modules and have to be used according to (partly) different *usage* modules. Car entities that are earmarked for market *A* should meet particular regulations (safety, environmental, performance) and rules for trading and using them. This suggests the shared use of the corresponding module (e.g. module: *A_market_rules_for_cars*) for both the *Xcar* entities and *Ycar* entities for market *A*. If a car entity is earmarked for the market *A* it will be registered in reference to the module *A_market_rules_for_cars.* Entities of the class car can be created in reference to a possible lives model composed from a number of alternative modules (such as *Xcar*, *A_market_rules_for_cars*, etc.). Changes affecting one or more life phases of *car*, can be localized in a small number of modules. Per module, past and current versions are to be managed. Compatibility rules must be observed for all modules defined for a class.

Definition: *(ix)* An *interflow event* is the constituent event of a process in the operation layer. The event may involve one or more entities, their proxies and relevant modules and records. It achieves a synchronous transition between successive situations at a physical cell and the corresponding transition of the proxies and records in its (cell) proxy. For non-trivial events, the following protocol can be implemented: the pre-conditions of the event cause an infrastructure-wide collection of relevant records and modules (life-histories and possible lives are constructed) . Eventuallly the pre-conditions are verified, and the modalities of the transaction (e.g., the price of a purchase) are determined, then it is committed. Finally, the post-conditions of the transaction result in the sending of messages (with or without delay) informing the relevant public and private bodies of the transaction and its modalities (e.g., the tax authorities will be informed of the value added of the transaction and accordingly collect taxes, etc.).

2.5 -Ro: Records for Observation

A single entity has a unique physical manifestation in the physical world (at any point in time it can only be present at one place), similarly, a proxy has a unique manifestation in the information infrastructure. In contrast with the unicity of a proxy for an entity, several records denoting the same entity may exist in the information infrastructure, typically one per module or life phase for which the entity (and its proxy) is or has been earmarked.

Definitions: *(x)* A *record* is a unit of information (text, number, picture, sound) that is kept in reference to a module, on an entity and its proxy, after the entity has been earmarked for the module (i.e., it has been decided that the module will describe (or prescribe) a phase in the life cycle of the corresponding entity). Modules that describe relationships (e.g., the marriage relationship) will hold records on all entities and proxies that are related to each other according to the module.

(xi) In the cybernetic domain, *observation* is the act through which an agent reads a record without affecting its value. In the physical domain, observation may involve the measurement of one or more properties of an entity, for instance with the purpose of entering data into a computational system.

2.6 Faithfulness between Physical and Cybernetic Domain

The linguistic primitives and their constructions give rise to different faithfulness relations between actual and possible situations in the physical domain and corresponding constructions in the cybernetic domain.

Definition: *(xii.a)* There is *observational faithfulness* between on the one hand: a current, past or future situation or a succession of situations in the physical domain ((successive) arrangements of entities); and on the other hand: a construction or succession of constructions in the cybernetic domain ((successive) arrangements of proxies and records), when, in terms of a number of modules, the records and proxies, the constructions in the cybernetic domain correctly describe the (succession of) situation(s) in the physical domain.
(xii.b) There is *operational or ontogenic faithfulness* between physical flow of entities in a physical domain, and computational flow (flow of proxies, change of records in a cybernetic domain), when, in terms of a number of modules, the successive transformations of the proxies in the cybernetic domain correctly describe the transformations in the physical domain.
(xii.c) There is *model or typogenic faithfulness* between arrangements of entities in a physical domain, and a model – consisting of modules, capabilities, types and proxies – in the cybernetic domain, when, the model correctly describes the entities, their capabilities and types, in the physical domain, and therefore enables the operationally faithful flow of physical and computational processes.

Given observational faithfulness between initial physical situation and cybernetic construction, then, operationally faithful interflow will achieve observational faithfulness at succeeding points in time. Operationally faithful interflow builds on typogenic faithfulness between models – proxies and enterprise and artifact possible lives models – and arrangements in the physical domain. It also assumes a regular interaction between the cybernetic domain and the physical domain.

2.7 Achieving Operational Faithfulness: Cyber-networks

A pre-condition for creating an information infrastructure is that one is capable of achieving operational faithfulness between activities in the physical domain and computations in a MiViPoRo conformant cybernetic domain. Because operations are distributed in space it is necessary to handle the interface between the two domains by means of a network op cell proxies, offering some of the generic interface services between the two domains. Such a cell proxy is called a cyber-cell. A network of cyber-cells is called a cyber-network.

Definitions: *(xiii)* A cyber-cell is a system that offers one or more of the generic services, and has the capability to communicate with other cyber-cells. The agent proxies at the cyber-cell represent the entities that are working at the place and the artifact proxies represent the artifacts that are there (being worked on or with). The records at the cyber-cell stand for proxies which have been or may be at the cell, or which the cyber-cell knows about. Agents and artifacts can enter and leave the cell, and likewise do proxies enter and leave the cyber-cell.

(xiv) A *model execution engine* is a software system that manages the computational side of operational faithful interflow at a cyber-cell. It will manage the access rights of

agents, the access possibilities of artifact proxies, and the transformations of proxies and records held in reference to the modules. Modules, together with the proxies present in a place, and stored records will determine or constrain the behaviour (possible events) of the entities and their proxies. The possible correlations of events are derived from the behaviours as defined in the modules (locally) and their grouping in possible lives models.

(xv) A *artifact life phase service body* (or ALPS) is a cell which undertakes to provide services – with both cybernetic or physical aspects – for a number of life phases of a number of entity classes. When it has a cyber-cell such that there is typogenic faithfulness between its modules and the capabilities of the cell, it can achieve a local connection between work during artifact and agent lives and an information infrastructure. The cyber-cell manages the local computational flow and the communications with other ALPSs. Examples are garages which offer repair services for cars, and national registers of private citizens which determine, among others, which adult citizens can participate to elections.

(xvi) A *cyber-network* is network of cyber-cells. It interfaces a distributed construction of modules, versions, proxies and records that constitute a MiViPoRo conformant cyberspace with users in the corresponding physical domain. Model execution engines at the cyber-cell may implement protocols in support of the global sourcing of possible lives models and life span data in support of (operation layer) decisions and work, as well as the global propagation (to other cyber-cells) of the consequences of such decisions.

2.8 A Cyber-network as an Information Infrastructure

An information infrastructure is a cyber-network which is maintained by a loose federation of agents – companies, public bodies and private citizens – and which is constructed such that the services of its corresponding ALPSs *extend over the possible life phases of a wide range of artifact types* as well as *over the relevant events in the lives of artifact occurrences (artifact histories)*: for each possible or required phase in the life of any artifact and for the agents involved, there are ALPSs that can provide the relevant support. The infrastructure, seen as a whole, constructs and maintains a dynamic map of past and future artifact and agent motions and transformations, as they matter for achieving individual or public goals in the territory. The infrastructure is distributed and should show robustness, modularity, maintainability, simplicity and efficiency. It should offer a coherent access to services. It should show typogenic faithfulness for a wide range of manufacturing resources and artifact types in a territory and achieve operational faitfulness for the lives of companies, consumers, public bodies and artifacts in the teritorry. Preferably with a minimal duplication of modules and data and cost of communication. An information infrastructure supports the global sourcing of artifact and process model data (modules), artifact histories (occurrences), and workflow elements (occurrences) in support of identification, decision, planning and action, prior to the proper propagation of the consequences of decisions and actions. It plays an enabling role for any major transaction involving private citizens, public bodies, companies, products or materials, or their models. If based on streamlined and harmonized product and artifact and enterprise models and data, the combined service offered by the information infrastructure takes away major non-productive burdens from (small and medium) enterprises and consumers, while providing them with the relevant information, on the spot, whenever they need it.

3 MiViPoRo AND ENV 40 003

Several aspects of MiViPoRo have been influenced by the framework for enterprise modelling ENV 40 003. Only the differences are highlighted.

(a) Possible lives models in MiViPoRo are intrinsically distributed and oriented towards the reuse of partial models (modules).

(b) In the dimension of Model of ENV 40 003 (with stepwise derivation from requirements model, over design model to implementation model) MiViPoRo emphasizes the requirements model. It anticipates model execution engines working directly with possible lives models, including those of ICT resources (processors, networks and peripherals).

(c) In the dimension of views, ENV 40 003 selects the views Organisation, Resource, Information and Function. This selection has been biased by traditional solution techniques. In comparison MiViPoRo's concepts are closer to the problem domain:
- the general categories of agent, artifact and *cell* and the involvement of agents and artifacts in time-space-matter situated events: the category of cell reflects the organisation view of ENV 40 003 cells have capabilities; agents group the functions which they can execute, and artifacts can be present as resources or products (different modules from their possible lives model).
- MiViPoRo emphasizes a distinction between the information that is kept on artifacts, agents and cells on the one hand, and the physical manifestation of these entities on the other hand. Thus all model-components (modules, versions, proxies and records) belong to the information view.

(d) ENV 40 003 assumes that there are generic (basic) constructs from which partial and particular constructs are derived. MiViPoRo's generic concepts can be identified in any model-based infrastructure system, or components of it. The generic concepts include: *(i)* the distinction between the cybernetic domain (modules, versions, proxies, transactions and records) and the physical domain (classes, entities, events, processes); *(ii)* the concepts of agent, artifact and cell; *(iii)* the concepts life phase, occurrence and possible lives model (a possible lives model is described by modules that meet certain syntactic rules of well-formedness); *(iv)* the modalities for the interaction of agents, artifacts and cells (events involve agents and artifacts and are situated in space, time and matter); *(v)* the distinction between static model describing possible dynamics, and the dynamic model (the "run time") which changes in response to physical flow, conform the static model.

(e) The role which partial models play in ENV 40 003's stepwise particularization process, is replaced in MiViPoRo by the role of cell-proxies and the modules which they contain. If one company manages modules for certain life phases of artifacts, then its suppliers or customers can define own, complementary modules for the life phases to which their core competences pertain.

(f) ENV 40 003 announces executability of enterprise models, but lacks the views that model execution engines and innovation coaches will have on the models. The MiViPoRo framework supports a more precise definition of the concept of a "model execution engine". A cyber-network plays for (distributed) possible lives models the same role as the "universal computing machine" (or Turing machine) plays for a computational algorithm.

Computations in the cyber-network are different from those of a universal computing machine because of the distributed memory, non-determinism, concurrent computing, dependability, ...

(g) The MiViPoRo innovation process combines – in a pragmatic way – features of step-wise derivation (ENV 40 003 dimension of model) and particularization (dimension of genericity).

4 CONCLUSION AND FUTURE WORK

The MiViPoRo framework supports one's reasoning about information infrastructure services for industry. It considers enterprise and artifact models as sets of compatible modules and offers guidelines for the definition and development of model execution engines, browsers and innovation coaches, and version managers. The definition of an information infrastructure – drawn from the framework's concepts – offers some theoretical application and validation of the framework. Further applications and validations should be more practical and address: the use of international standards for product and process modelling in the definition of modules; the specification of model execution engine and innovation coach (considering standards such as those for electronic data interchange and open distributed processing); the precise description of cybercells (including the protocols for interactions with human and other agents, for mutual communication and computation); and demonstration projects.

5 REFERENCES

Alting, L. and Jørgensen, J. (1993) The Life cycle concept as a basis for sustainable industrial production. *Annals of the CIRP*, 42/1.

Browne, J., Sackett, P. J., and Wortmann, J.C. (1995). Future manufacturing systems - towards the extended enterprise. In: *Computers in Industry*, Vol. 25:235–254.

CEN/CENELEC (1990) *ENV 40 003: Computer integrated manufacturing – systems architecture – framework for enterprise modelling.* European prestandard, CEN/CENELEC, Rue de Stassart 36, 1050 Brussels, Belgium.

CEN (1995) *ENV 12 204 Advanced Manufacturing Technology - Systems Architecture - Constructs for Enterprise Modelling.* European prestandard, CEN, Brussels.

ESPRIT Consortium AMICE, editor. (1993) *CIMOSA: Open System Architecture for CIM.* Springer Verlag, Berlin, 2nd, rev. and ext. ed edition, 1993.

Gielingh, W. (1993) Towards an infrastructure for product data technology. In *Realizing CIM's Industrial Potential*, pages 70–81. IOS Press, Amsterdam.

Goossenaerts, J. (1993) Enterprise formulae and information infrastructures for manufacturing. In: Yoshikawa, H. and Goossenaerts, J., editors, *Information Infrastructure Systems for Manufacturing, IFIP Transactions B-14*, Elsevier Science B.V. (North Holland), Amsterdam.

Goossenaerts, J. and Bjørner, D. (1994) Interflow systems for manufacturing: Concepts and a construction. In: Ladet, P. and Vernadat, F.B. editors, *Integrated Manufacturing Systems*

Engineering – Proceedings of the European Workshop on Integrated Manufacturing Systems Engineering (IMSE'94). Chapman & Hall, London, UK.

Goossenaerts, J., Pels, H.J., and Wortmann, J.C. (1996) A research portfolio for product life cycle modelling and execution service engineering. In *Proceedings of the IFIP WG 5.7 International Conference on Advances in Production Management Systems, Kyoto*, November 1996.

Inagaki, K. (1993) The role of information systems infrastructure in production management. In: Yoshikawa. H. and Goossenaerts, J., editors, *Information Infrastructure Systems for Manufacturing, IFIP Transactions B-14.* Elsevier Science B.V. (North Holland), Amsterdam.

Krause, F.-L., Kimura, F., Kjellberg, T., and Lu, S.C.-Y. (1993) Product modeling. *Annals of the CIRP*, 42(2):695–706.

Krause, F.-L. and Kind, C. (1996) Potentials of information technology for life-cycle-oriented product and process development. In F.-L. Krause and H. Jansen, editors, *Life-Cycle Modelling for Innovative Products and Processes - Proceedings of the IFIP WG 5.3 International Conference on life-cycle modelling for innovative products and processes, Berlin, Germany, November/December 1995*, Chapman & Hall, London, UK.

OSD-CALS (1996) *OSD-CALS world wide web site - Home page.* http://www.acq.osd.mil/cals/.

Scheer, A.-W. (1994) *Business Process Engineering – Reference Models for Industrial Enterprises.* Springer Verlag, Berlin.

Williams, T.J. (1994) The Purdue Enterprise Reference Architecture. *Computers in Industry*, 24/2-3, 1994.

Yoshikawa, H. (1981) General design theory and a CAD system. In: Sata, T. and Warman, E., editors, *Man-Machine Communication in CAD/CAM*, Elsevier Science B.V. (North Holland), Amsterdam.