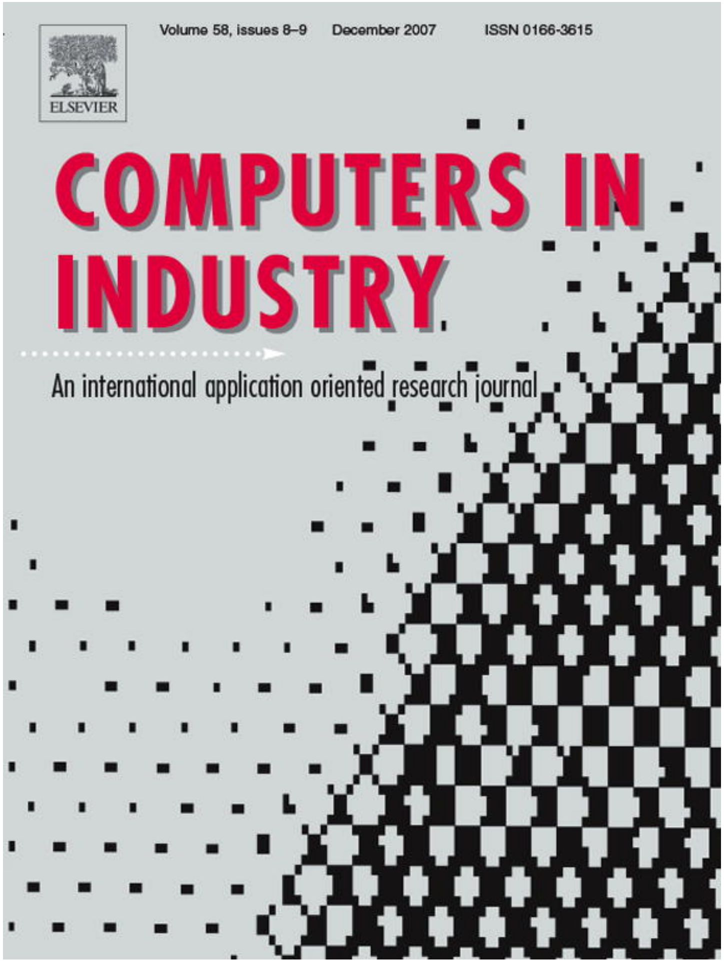
Volume 58, issues 8–9 December 2007 ISSN 0166-3615
ELSEVIER
COMPUTERS IN INDUSTRY
An international application oriented research journal

Available online at www.sciencedirect.com
ScienceDirect
Computers in Industry 58 (2007) 794–813
COMPUTERS IN INDUSTRY
www.elsevier.com/locate/compind

# Enterprise modeling of a project-oriented fractal company for SMEs networking

M. Mercedes Canavesio [a], Ernesto Martinez [a,b,*]

[a] UTN, Facultad Regional Santa Fe, Argentina
[b] CONICET, Instituto de Desarrollo y Diseño, Avellaneda 3657, Santa Fe (S3002GJC), Santa Fe, Argentina

Received 20 February 2006; received in revised form 26 September 2006; accepted 8 February 2007
Available online 26 March 2007

**Abstract**

To address competitive threats and concentrate in their core competences and strengths networking is the alternative of choice for survival and prosperity of most small and medium enterprises (SMEs) all over the world. For networking advantages to be fully grasped, an enterprise model describing the main organizational structure and relationships, information flows, management roles, actor behavior and constraints in the network is required. This work presents a conceptual model for SMEs networking based on the fractal company approach and concepts like projects, resources, goals, specialized actors, plans, and relationships thereof. The fractal company idea is a conceptual enterprise design that seeks to achieve a high degree of flexibility to react and adapt quickly to environmental changes using decentralized and autonomous organizational units known as *fractals*. In this work, each fractal management unit is modeled as a *project*. The fractal company is thus seen as a temporary set of client–server and delegation-to-do relationships among *project-managers* interacting so as to diversify product portfolio, gain economy of scale and share expensive costs. The proposed enterprise model can be used to specify the requirements of an integrated information and management system for virtual SMEs networking. The enterprise model is exemplified using a case study related to process development of a new pharmaceutical product in a SME network of the specialty chemical sector.


*Keywords:* Fractal company; Project management; Systems thinking; Enterprise modeling; SMEs networking

## 1. Introduction

At present, manufacturing companies face an unprecedented process of change in their business environment. This change is mainly caused by globalization, product customization, variations in demand patterns and rapid technological developments. This situation is placing all manufacturing enterprises under significant competitive pressure. To survive, companies must increase product portfolio, reduce time-to-market, shorten product-life cycles and at the same time maintain good product quality and reduce investment costs. Competitive threats are much worse for small and medium enterprises (SMEs) which are reengineering their production and management systems to compete successfully. SMEs have to organize themselves into effective production networks [1] to achieve a higher degree of flexibility, agility and low costs to cope with the increasing rate of change and complexity of a highly competitive environment. To address competitive threats and concentrate in their core competences and strengths networking is the alternative of choice for each individual SME survival and prosperity [1–3]. For the advantages of networking to be fully grasped, an information/management system describing roles, functions, tasks, objectives, goals, etc., of all actors and resources involves is required.

Objectives of a SMEs network include increased agility to competitive threats, a more comprehensive pool of skills and resources, economy of scale and product portfolio diversification. The main benefits of networking for each individual SME are [4]:

- Access to new markets, by integrating product portfolios that are out of reach for a single SME.

* Corresponding author at: CONICET, Instituto de Desarrollo y Diseño, Avellaneda 3657, Santa Fe (S3002GJC), Santa Fe, Argentina.
Tel.: +54 342 4555229; fax: +54 342 4553439.
*E-mail addresses:* mcanaves@frsf.utn.edu.ar (M.M. Canavesio), ecmarti@ceride.gov.ar (E. Martinez).

0166-3615/$ – see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.compind.2007.02.005

- Productivity improvement, by increased usage of each node productive capacity.
- Improved stability, through joint response to perturbations that would be unbearable for a single SME.
- Sharing expensive costs, such as a new product design or an advertising campaign.

There have been many attempts to establish SMEs networks but they have not produced so far the expected benefits and network life cycles are rather short. Failure of enterprise networks can be partially explained on the grounds of:

- Loss of autonomy and identity of each SME because there exists a central node that controls and distributes tasks to each node in the network.
- Loss of individual competitive capacity of each SME. Each enterprise may perform only some tasks of the production cycle.
- Too much of an emphasis on cooperation rather than competition, so that inefficiency costs are distributed among nodes.

To grasp all the benefits of enterprise networking and to mitigate its drawbacks is mandatory to define an enterprise model to influence by design the behavior of each SME and relationships thereof. There are different proposals with names like bionic [5,6], holonic [7–10,35], and fractal [11] manufacturing that can be used to define a management system for distributed manufacturing in a SMEs network. The underlying idea of a flexible network made up of autonomous, yet interdependent manufacturing units is a common feature to all of them. Of the detailed comparison of these approaches made in [12], the fractal company idea is the more appealing one from the standpoint of modeling management tasks since self-organizing and self-optimizing unit characteristics allow more room to differentiate goal management from resource management in SMEs networking (see client–server and delegation-to-do relationships proposed below).

The fractal unit has a high level of autonomy to achieve its goals whereas the autonomy in the holonic and bionic units is inherently limited by its internal design. More specifically, a bionic unit has a behavior pre-defined by its designer and an external supervisor whom regulates the relationships among interacting units. On the other hand, holon decision-making options are constrained by a set of fixed rules that define the permissible steps in performing its control role whereas externally its autonomy is limited by the cooperation rules that define the holarchy to which a given holon belongs. In comparison, fractal units exhibit much greater levels of autonomy which provides plenty of room for each unit individual self-learning and self-optimization.

This work presents an enterprise model for SMEs networking based on the fractal company idea [11]. The fractal management unit is modeled as a *project* [14,15]. In the model, each project is seen as an autonomous, temporary entity within the enterprise network, in which different types of expertise are combined to achieve a concrete goal(s) or deliverable(s) (e.g. completing a given order, product development, advertising campaign, satisfying resource usages as scheduled, etc.). The proposed model of the project-based fractal management system separates the management of goals from the management of the resources needed to obtain such goals but in both cases the fractal management unit is conceived as a *project*. To do this, the enterprise model is made up of a hierarchy of autonomic management units with a project manager and a managed object. The managed objects are either *ends* or *means* whereas actors assuming project manager roles are *ends-manager* or *means-manager*, respectively. An ends-manager is driven mainly by efficacy in project delivery through proper application of the best and most effective resources. Also, if a project manager is playing the means-manager role, he/she seeks to guarantee resource usages as scheduled while maximize resource utilization over time. In the proposed enterprise model SMEs networking is seen as collection of temporal relationships between actors playing project manager roles. These project managers may belong to the same or different autonomous and interdependent SMEs (Fig. 1). The key to the project-based fractal company is establishing *vis-à-vis* client–server relationships between an ends-manager actor and a means-manager actor in an open market economy. Also, delegation-to-do relationships are established by project manager at different abstractions levels. The conceptual model is exemplified below using a case study related to process development of a new pharmaceutical product.

## 2. Project-based fractal enterprise model

An *enterprise model* is an abstraction that describes and represents the main structures and relationships, information flows, roles, goals, resources, behavior of actors, and constraints within an enterprise [16,17].

The main advantages of an enterprise model for SME networking are:

- Allows providing accurate responses to queries such as: *What actor X is doing? Who is doing the activity Y? When, Where, How and Why is the activity Y done*?
- Clearly states the requirements and design specifications of an information system for distributed decision-making and coordination control.
- Favor a systems thinking and modeling perspective to describe and analyze enterprise network dynamics and constraints.
- Emphasize collective learning and knowledge accumulation.

### 2.1. Project-based fractal management unit

The SMEs network model that is presented here is based on the fractal company idea which was proposed by Warnecke [11]. The fractal company is a conceptual enterprise model that aims to achieve a high degree of flexibility to react and adapt quickly to environmental changes using decentralized and autonomous organizational units known as *fractals*. The term

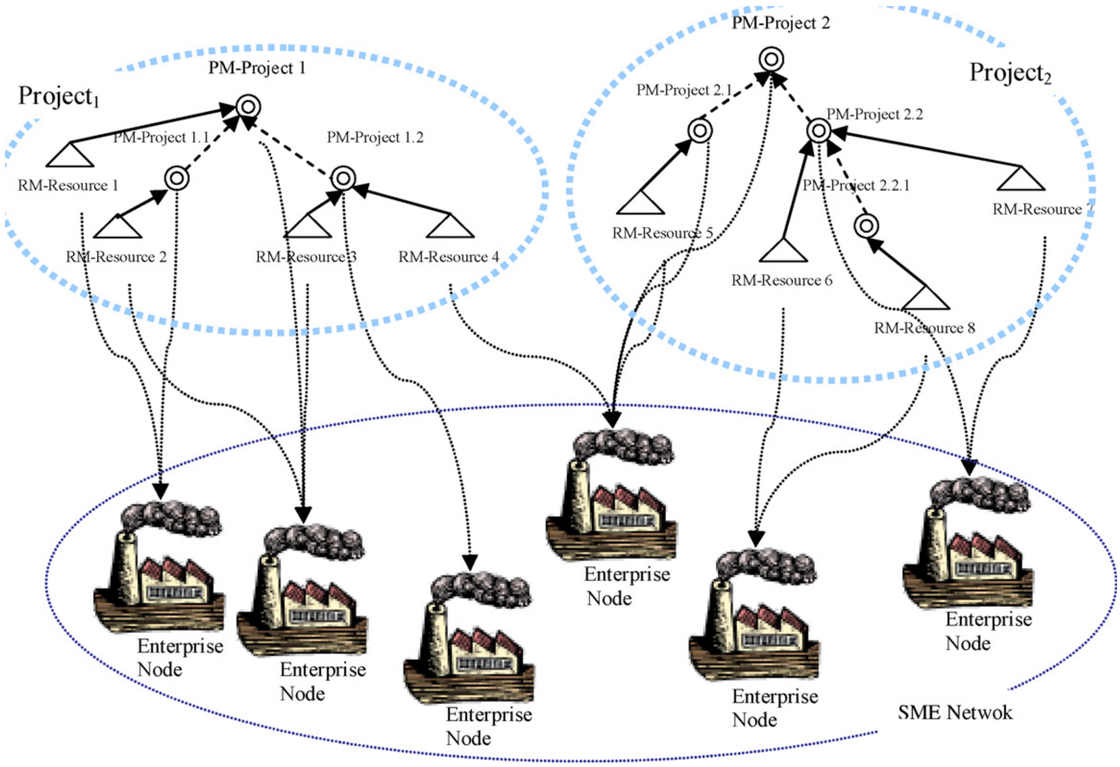


Fig. 1. SMEs networking using projects.

'fractal' is borrowed from the mathematical domain. It is defined as a shape or structure that describes an identical pattern that replicate itself at distinct abstraction levels in a recursive way [18,19]. In the organizational domain [11], a fractal unit is defined as "an independently acting corporate entity whose goals and performance can be precisely described". Each fractal unit acts independently while attempting to achieve the enterprise goals [11,12,19,20]. However, there not exists yet a detailed specification regarding the fractal design, i.e. internal organizational structure, and the type of interactions fractals can experience in their lifetime. A fractal is generally seen as an organizational unit which has unique objectives, achieves concrete results and acts autonomously in a self-optimizing way whilst interacting with other fractals [27]. However, with few exceptions the fractal company model has not been defined with enough details required for its implementation. An interesting application of the Warnecke's idea is a control architecture of a flexible manufacturing system where the fractal unit is constituted by five modules [13,29]. Also, in the ADRENALIN project [30] the fractal company principles are used to embed a process-oriented approach in modeling the supply chain dynamics, emphasizing self-similary, self-optimization, self-organization in the information system design for enterprise integration.

In the proposed enterprise model for SMEs networking, the fractal management unit is modeled as a *project* [14,15]. Thus, each project fractal is seen as an autonomous, self-optimizing, self-learning and goal-driven entity, in which different types of expertise are combined to achieve a concrete goal or deliverable (e.g. completing a given order, product development, advertising campaign, satisfying resource usages as scheduled, etc.). Also, there exists a project manager role play by an actor or agent that possesses all the freedom to manage the usage of resources so as to make possible the achievement of its project goal. In the project-based fractal enterprise model, both ends and means are managed by a project. Thus, the fractal management unit is conceived as constituted by a project-manager and a managed object – ends or means – (Fig. 2). The project manager is able to sense the present situation of the fractal unit, to interact with other project managers, to monitor the managed object and its external environment in order to construct and execute plans based on an analysis of this information. Furthermore, learning allows the accumulation of knowledge based on the actions executed by other project managers.

Each instance of a project fractal has the following characteristics that are present at whatever abstraction level:

- *Autonomy.* In each project, the project manager possesses enough freedom to execute activities and control resources in order to achieve the project goal.
- *Goal-oriented*. Each project produces a specific deliverable for a clearly identified client and might need as inputs deliverables of one or several other projects.
- *Self-similarity.* A project is constituted by a project manager and a managed object – ends or means – (Fig. 2). This structural pattern repeats itself at all abstraction levels. Thus, the similarity characteristic allows defining *recursive relationships* between projects. In other words, each project can be defined as a part of a super-project or as containing

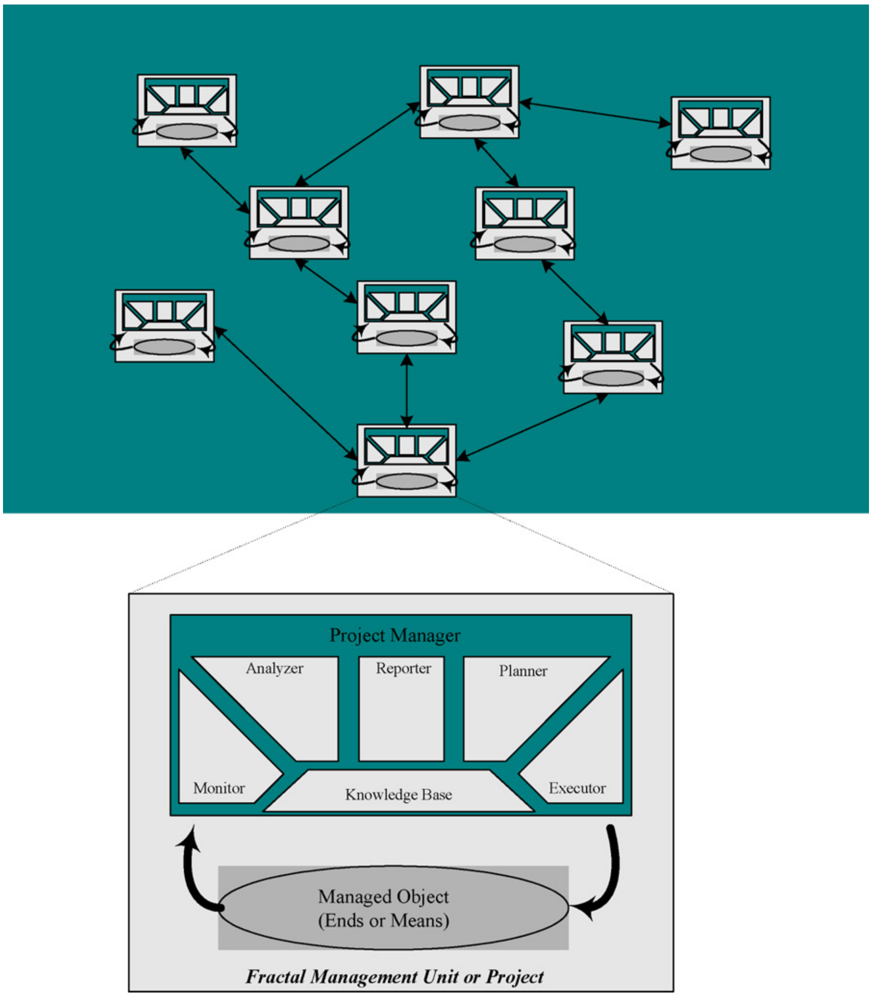


Fig. 2. Internal structure of a project as a fractal management unit.

distinct levels of sub-projects (sub-projects, sub-sub-projects, etc.). Each particular instance of the project is in itself a partially ordered set of activities conducive to an objective or goal which can be in turn an input to another project instance (Fig. 3).

- *Learning*. Each project stores project-relevant knowledge and information that allow project managers to improve their own decision-making.
- *Self-optimization*. The project manager performance is under constant assessment and evaluation. For that, project manager has to review his/her goals and objectives in order to fit them to the requirements of his/her environment improving thus his/her efficiency and effectiveness.
- *Self-organization*. Ref. [28] defines self-organization as the integration of autonomy and learning capabilities within autopoietic entities. The project manager is an autonomous and intelligent actor or agent whose behavior is not strictly programmed, nor perfectly definable *a priori* and it is not predictable. This allows the project manager permanently to adapt his/her behavior and links with other project managers always in order to achieve the project goal. The self-organization characteristic allows the fractal company to supports the dynamic reconfiguration of network connections between projects and the creation of project instances in order to self-adapt internal and external changes. Thus, constantly new project instances and relationships between project managers are created and at the same time many of them disappear. The case study (Section 3) will illustrate the dynamic restructuring process in the project-oriented fractal company.
- A project is a *temporal* entity within the SMEs network. It reaches a certain objective (e.g. completing a customer's order or new product design) in a limited amount of time.
- Projects have a *life cycle* reflecting the five ever-present phases in carrying out an activity (Fig. 4): (1) Specification-What? (2) Planning-How? Who? When? (3) Execution; (4) efficacy and efficiency control; (5) closing. Phases (3) and (4) are concurrent. Should an abnormal situation is detected or a non-planned event happens, the project manager is responsible for repairing the plan by taking proper actions that guarantee the achievement of project goal(s). Thus, phases (2)–(4) are repeated again in a distributed environment which favors unforeseen event handling based on local-based repairs of plans.
- *Holistic view*. As mentioned before, the project manager decides freely based on his/her own experience and knowl-

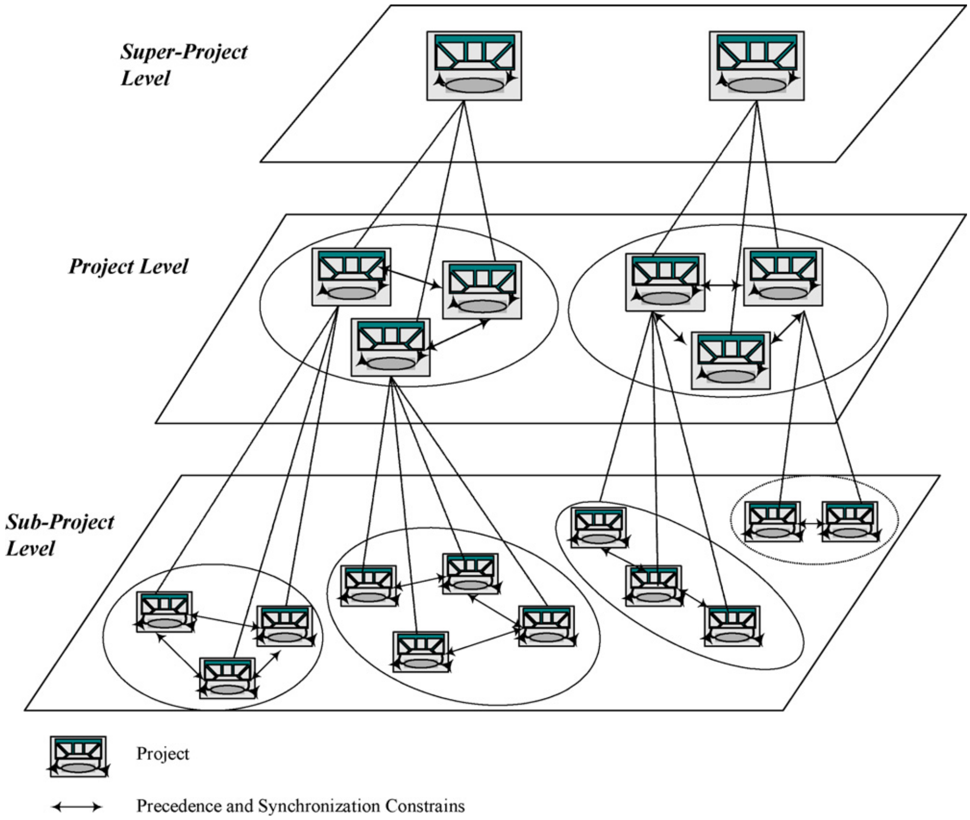


Fig. 3. The recursive relationship in the project-based fractal company.

edge base, the way to obtain the assigned goal and how to improve permanently his/her productivity but in view of reaching the objectives of the fractal company as a whole. The holistic view of the project-oriented fractal SME network generates the idea of management of a set of companies linked virtual and temporarily to achieve business goals, in spite of the decentralization in the management and control of sub-goals that each one is responsible for.

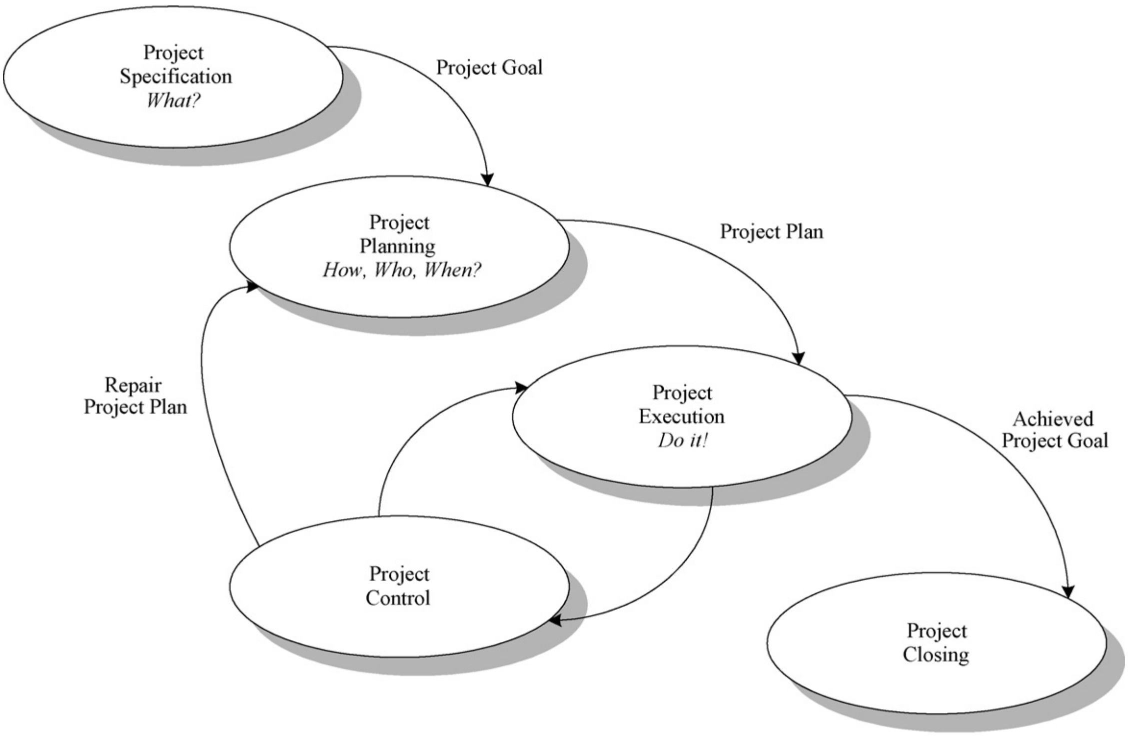


Fig. 4. The project life cycle.

One of the attractiveness of the project-based fractal approach is being a *recursive management structure* [14,21]. This recursive structure is both a generator of efficiency and absorber of complexity to embed a highly adaptive and responsive organizational design that can balance each SME internal and external perspectives on both the short- and long-term horizons. Another advantage of the proposed recursive structure is that naturally lends itself to distributed decision-making [15–22].

### 2.2. Conceptual design

In order to achieve high degree of flexibility, effectiveness and efficiency in the allocation and utilization of core resources and core competences of SMEs the proposed model highlights the difference between *ends* or *goal management* and *means* or *resource management* [14,15]. Each fractal management unit or project is made up of a project manager and a managed object (end or means). Project managers are active, autonomous, and intelligent entities in the SMEs network that make decision and perform tasks in order to achieve project goals. In order to do this, the project manager defines a plan and negotiates resources with other actors, then he/she controls plan execution and manages allocated resources. The managed objects are passive entities and their states change as the result of external events that take place in each unit environment or decisions taken by project managers.

A fractal management unit or project is created to achieve a specific goal. This *goal* describes the desired result of a project and guides the design and execution of fractal task plan. Therefore, a goal comprises a set of attributes that describes the required result and economic and temporal aspects that constrain the budget of time and money that can be spent as part of plan execution. The goal attributes and constraints, in turn, are guided by customer requirements. A *task plan* is a network of ordered tasks that should be executed to achieve the project goals. In this network, each task has an assigned duration of time, inputs and outputs constraints, and resources that will be used or consumed during the execution of a task. In order to reduce project complexity and allow project manager to control project performance and its completion, goals are broken into several sub-goals which in turn should be achieved by other project instances. Fig. 5 shows the fractal management unit or project model using UML (unified modeling language [25]), a standard to draw object-oriented diagrams.

#### 2.2.1. Project manager, roles and relationships

In the proposed model, the project managers are autonomous and intelligent actors or agents that belong to any SME of the network. These actors can be human or artificial actors (e.g. a software agent) and they have to have skills that are required to play as project manager (Fig. 6). When a project is closed the project manager performance is evaluated by other project managers. Then, figures of merit are recorded and make public within the network as a basis for future actor selections.

The main responsibilities of a project-manager are to achieve the project goal, define the task plan, receive messages and information from managers of its father project (if any), answer requests to its own father project (if appropriate), and negotiate with other project managers playing means-manager role the allocation of the needed resources. Also, a project manager has to receive messages and information from its child projects, detect and predict abnormal situations, and if it is necessary to repair the project task plan (Fig. 5).

When a project manager is playing a means-manager role has to ensure resources usages as scheduled, maximize resource utilization over time, repair out-of-service resources, plan and perform preventive maintenance and calibration of instruments or tools. Formally, a role is defined by a set of responsibilities, skills and communication links (Fig. 6). Each role comprises a set of skills and capabilities that are necessary to fulfil role responsibilities. Role communication links defines the protocol that allows project managers to communicate among them in the SME network.

As mentioned before, project managers can assume either an ends-manager role or a means-manager role. Depending on the role that the project managers are playing two types of relationships are established among them: *client–server* and *delegation-to-do* relationships (see Fig. 7 for details).

The *client–server relationship* is established between a project-manager that plays ends-manager role and other actor that plays a means-manager role. In this relationship the ends-manager is the consumer or client for a given resource and the means-manager is the supplier or server of that resource. Then, the means-manager is only responsible for satisfying resource availability and the ends-manager is responsible for good usage and management of the grabbed resource.

The *delegation-to-do relationship* is established between two project managers that are playing ends-manager roles at different abstraction levels (project and sub-project). In the enterprise model, there exists a constraint on project managers who establish delegation-to-do relationships: a project manager cannot be the project manager of another project that is a descendant to the project which it manages. As the project-manager of a child project has enough freedom to decide how to achieve its assigned goal, a delegation-to-do relationship only allows the project-manager of a father project to execute coordination actions between its child sub-projects.

The two relationships defined above are established by free negotiation among actors of SMEs that are interested in exploiting a given business opportunity. Also, if the actors involved in a given relationship belong to the same company, the relationship is *intra-company*, otherwise the relationship is *inter-companies*. Examples of intra-, inter-companies, client–server and delegation-to-do relationships will be discussed using a case study in the next section.

#### 2.2.2. Project-oriented management

As mentioned previously, the proposed enterprise model separates the management of ends from the management of the required means to achieve those ends and in both cases uses the project as the fractal management unit. Accordingly, the goals of a project fractal may seek are of two different nature. It is possible to identify goals whose achievement is restricted to

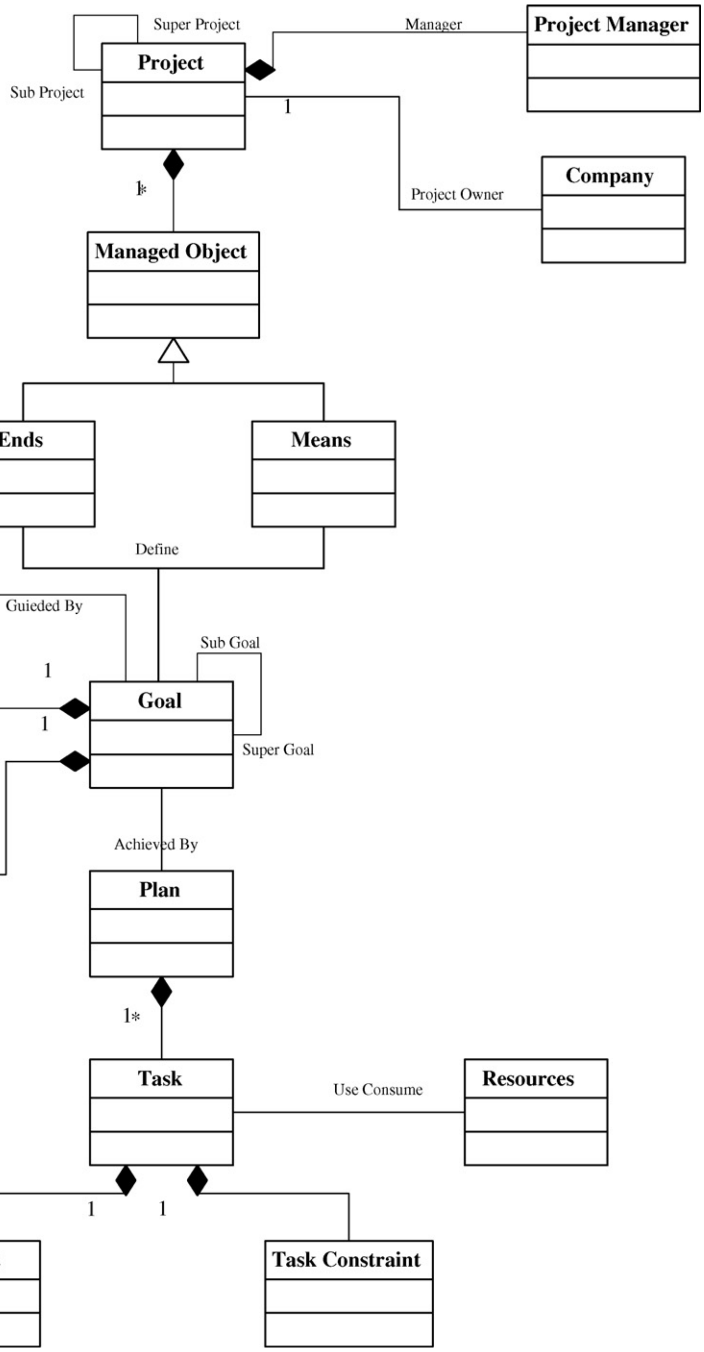


Fig. 5. Class diagram of the project model.

perform tasks that merely use or consume resources, whereas other goals are related to the timely provision of a given resource to be used or consumed by a given task. For instance, let P1-SBB-1 be the project whose goal is *to manufacture 10 batches of SBB-1 compound before 24/03/05 and deliver them to the clinical test group*. To achieve this goal, a specific pilot plant and its operators must be secured. Therefore, the P1-SBB-1 project-manager establishes a client–server relationship with a pilot plant manager. In order to accomplish the availability of the required resource, the pilot plant manager creates the R1-PP-SBB-1 project whose goal is *to deliver the pilot plant and its workers on agreed conditions in the contract before 15/02/05*. This is a goal related to the provision of a given resource to the P1-SBB-1 project. Thus, the managed object of this fractal unit is the compound resource: pilot plant plus operators. Then, the pilot plant manager is seen as means-manager by the manager of the P1-SBB-1 project and as ends-manager of the R1-PP-SBB-1 resource usage project.

The project management defines the framework for the fractal company management. Within this framework, project managers have enough freedom to act and to make decisions. A brief description of the project-oriented framework has been summarized by the activity diagram in Fig. 7. An activity diagram is an UML diagram that illustrates the dynamic view of

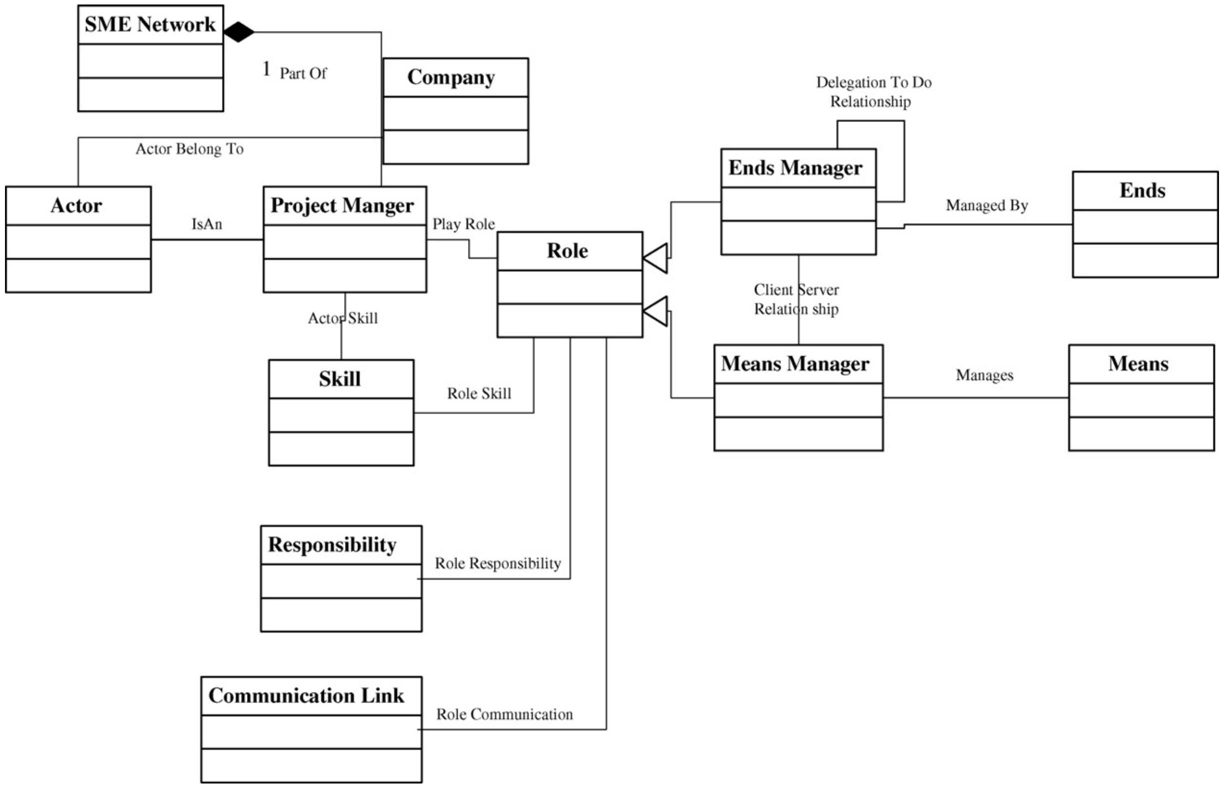


Fig. 6. Class diagram of the role model.

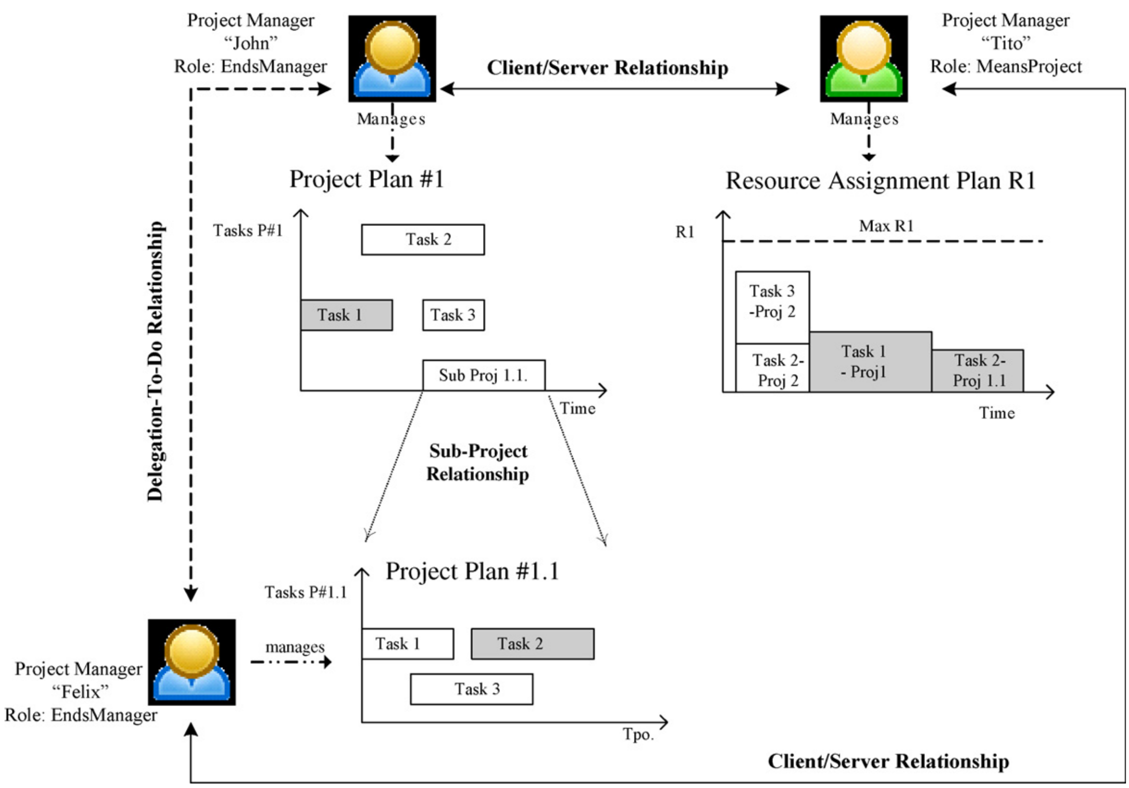


Fig. 7. Relationships among project managers in the project oriented fractal SME network.

a system [25]. In this activity diagram the main actions and decisions facing a project manager are depicted. Also, the diagram reflects the evolution of the project through its life-cycle highlighting the successive states that a project may be in as result of the actions taken by its project manager.

Each project instance is initialized when a goal has to be achieved. Thus, the first state a project enters in is "*ProjectInInitialization*". Then, an actor is assigned to the project manager role and the project state changes to "*InitializedProject*". When, the project manager starts to figure out a project plan so that the project state changes from "*InitializedProject*" to "*ProjectInPlanning*". While the project is in this state, the project manager defines a plausible task network which aimed at achieving the project goals, through negotiating resources, and delegating sub-projects to different actors as project managers of these (child) sub-projects. Only when all tasks of a given project plan are defined and all the required resources have been contracted with given means-manager actors, project state changes to "*PlannedProject*". In order to begin execution of each project task, the task preconditions must be true. These task preconditions are related to time attributes (start time, earlier start time, later start time, end time, earlier end time and later end time), readiness of inputs (deliverables of previous projects), and availability of resources. Thus, project state changes to "*ProjectInProgressOnSchedule*" when the first start time task and all task preconditions are true or earlier start task time expires. As long as the project state is "*ProjectInProgressOnSchedule*", the project manager monitors plan progress and cumulative resource usage while predicts the impact of execution delays in future tasks. During this project monitoring stage, the project manager knows input and resource states by mean of messages that are exchanged with other project managers. For example, after a set up task has finished in time and the resource is ready to use, means-manager sends ends-manager a message announcing that the required resource is ready. A more detailed description of project manager interactions will be described in the following sub-section. If an unexpected event occurs and causes that one or more tasks have to be interrupted, the project manager must repair the project plan and create a new plan to achieve the project goal. While the project plan is not feasible, the project state changes to "*ProjectNoFeasible*". Only if a feasible project plan is created, project state return to "*ProjectInProgressOnSchedule*", otherwise the project will be closed without achieving its goals. When the project plan execution has finished and the project goal has been achieved, the project state changes to "*FinishedProject*". Then, the project manager announces the closing of the project to the clients who will evaluate the performance of the project manager and stores the given qualification. At the same time, the project manager archives the statistic project data for future project proposals. Thus, the project state changes from "*FinishedProject*" to "*ProjectInClosing*". These activities finish with the closing of the project. Thus, project changes to its last state "*ClosedProject*".

*2.2.2.1. Resource management.* The framework for project management described previously is also applied to the resource management in the fractal company. At any time, the current resource state has to match the current scheduled task on resource assignment plan otherwise an unplanned situation is happening and the corresponding means-manager must look after the causes and repair the resource assignment plan in order to accomplish her/his responsibilities and goals. The state changes of a given resource take place during the stage of the control of resource assignment plan. These changes are not evident in Fig. 8. For that, this section briefly describes the pre-conditions and actions that cause state changes of a given resource. Fig. 9 shows the state transition diagram for a resource highlighting the main actions that are available to the means-manager at each resource state. In the diagram, each state represents a specific resource state at the current time while future and past resource assignments (for using, maintenance, set up, etc.) are stored by a project plan. A state transition diagram is an UML diagram that illustrates the dynamic view of a system [25].

For each new resource entity, a new instance of a project is created to manage it. Thus, the first activity of the project manager is to initialize the new resource data through "*InitializedResource*". At this stage, resource attributes, capabilities, constraints and other data that are interesting for deciding resource assignments are defined. After that, the resource state changes to "*IdleResource*". This state suggests that the resource is not assigned to any task at the current time. From this state a resource may change to "*SetUpResource*," or to "*OutOfServiceResource*". There exists resources that need to be prepared before they can be used by a given project task. This resource preparation is called a set up task. Thus, if a set up task is planned in a resource assignment plan and the resource is idle, or it has finished an out of service period, maintenance, or usage, the resource state changes to "*SetUpResource*". After a set up task has finished, the resource is ready for its use and the means manager sends to the corresponding ends-manager a message announcing that the resource state is ready. After that, the resource is grabbed. Thus, the resource state changes to "*ResourceInUse*". A resource in this state can change to either "*ResourceAvailable*" if the task relinquishes the resource or "*ResourceOutOfService*" for any event that prevents using the resource even though it has been allocated. In both case the ends-manager sends to the means-manager a message indicating the resource state. If the resource is out of service, its manager has to begin the execution of resource maintenance and repair its assignment plan. While a resource remains in the "*ResourceUnderMaintenance*" state, a warning message must be send to all project-managers whose plans may be affected by the current state of the resource. These messages should describe the current resource situation and also propose a new resource supply contract or accept liability costs for non-compliance agreement. When the resource is removed from the center, the corresponding project instance is closed.

### *2.2.3. Project manager interactions modeling*

This section discussed modeling issues for describing fractal unit interactions during the establishment of relationships and the protocol to exchange information after a given relationship

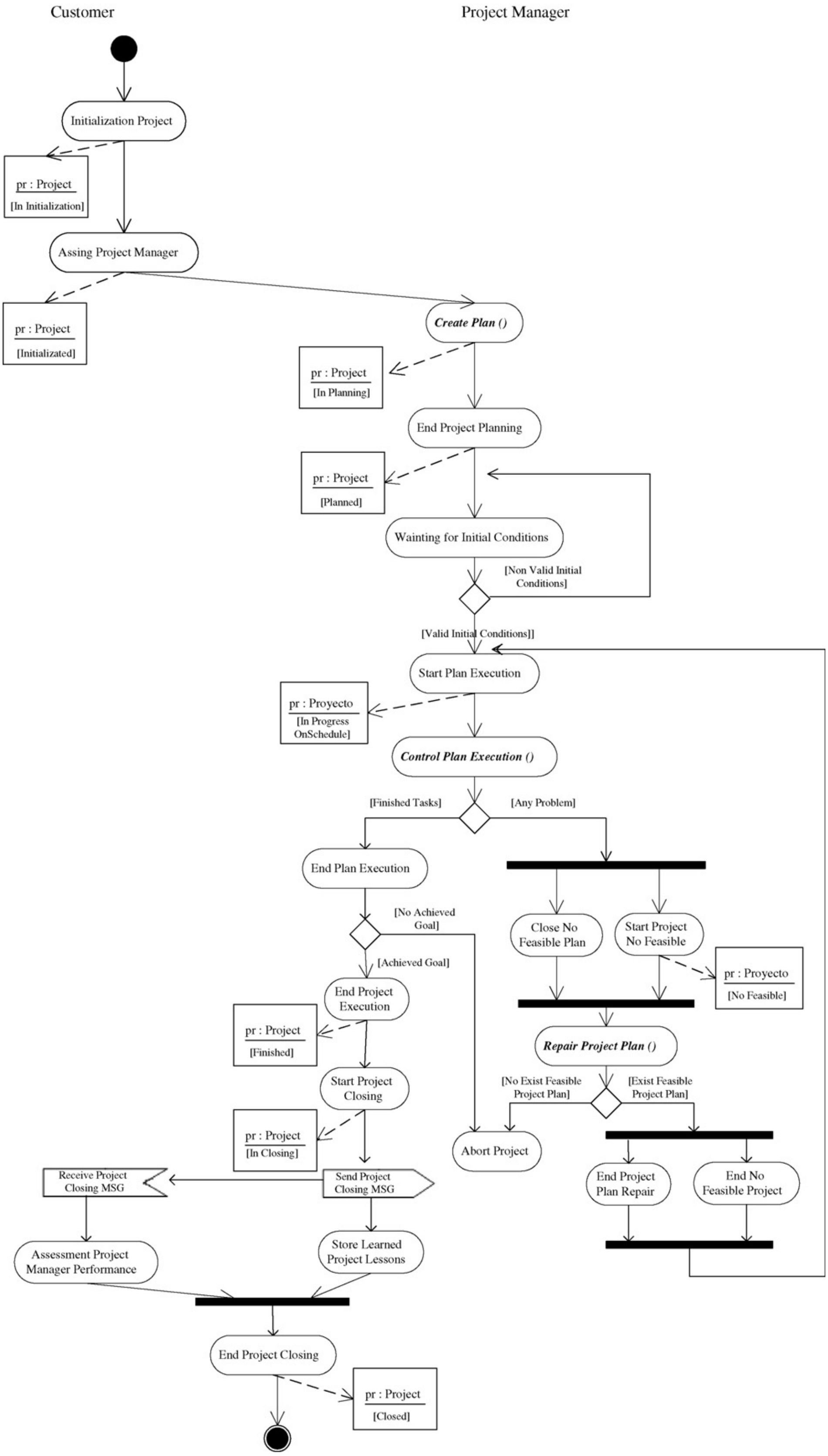


Fig. 8. Activity diagram of the framework for project management.

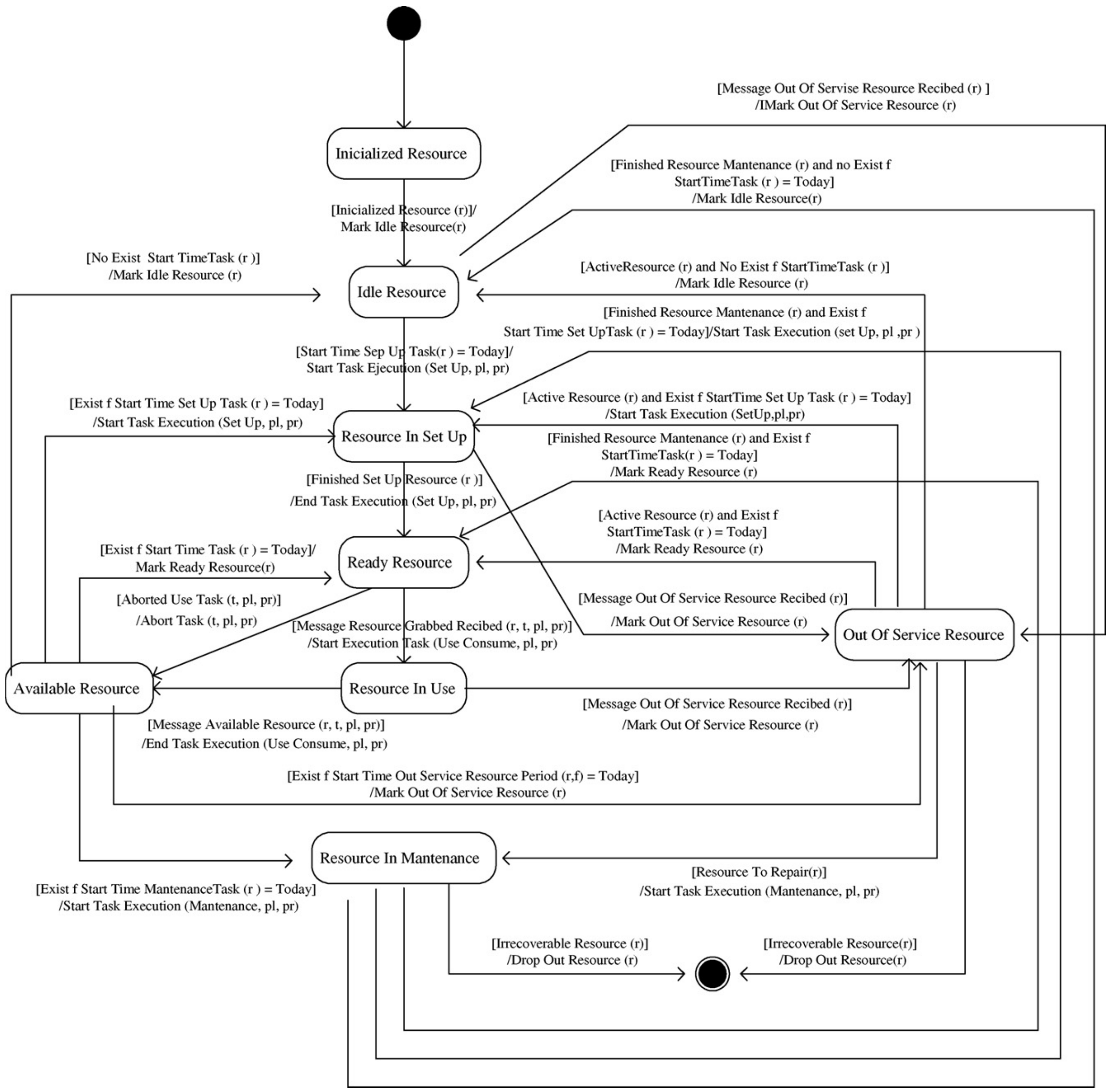


Fig. 9. State transition diagram for a resource.

has been established. The sequence diagram [25] is used to describe a generic project manager interaction.

*2.2.3.1. Establishing relationships among project managers.* Establishing new relationships among project managers is represented through an adaptation of the well-known *contract net protocol* [23,24]. In the original contract net protocol, there are two actors with different roles, *manager* and *bidder*. Four stages are involved in the establishment of a contract between a manager and one or more bidders (Fig. 10):

(1) Request for service (Fig. 10(a)): the manager announces a requirement (task or resource) to all potential bidders. The announcement contains the description of the task or resource.
(2) Submission of proposals (Fig. 10(b)): on receiving the tender announcement, each bidder capable of answering the requirement draws up a proposal and submits it to the manager.
(3) Awarding of contract (Fig. 10(c)): on receiving and evaluating the submitted proposals, the manager awards the contract to the best bidder.
(4) Establishment of contract (Fig. 10(d)): the awarded bidder may either commit the contract or refuse to accept it by sending messages to the manager. For the latter case, the manager will reevaluate the bids and award the contract(s) to another bidder(s).

In the proposed enterprise model, client–server and delegation-to-do relationships among project managers are established

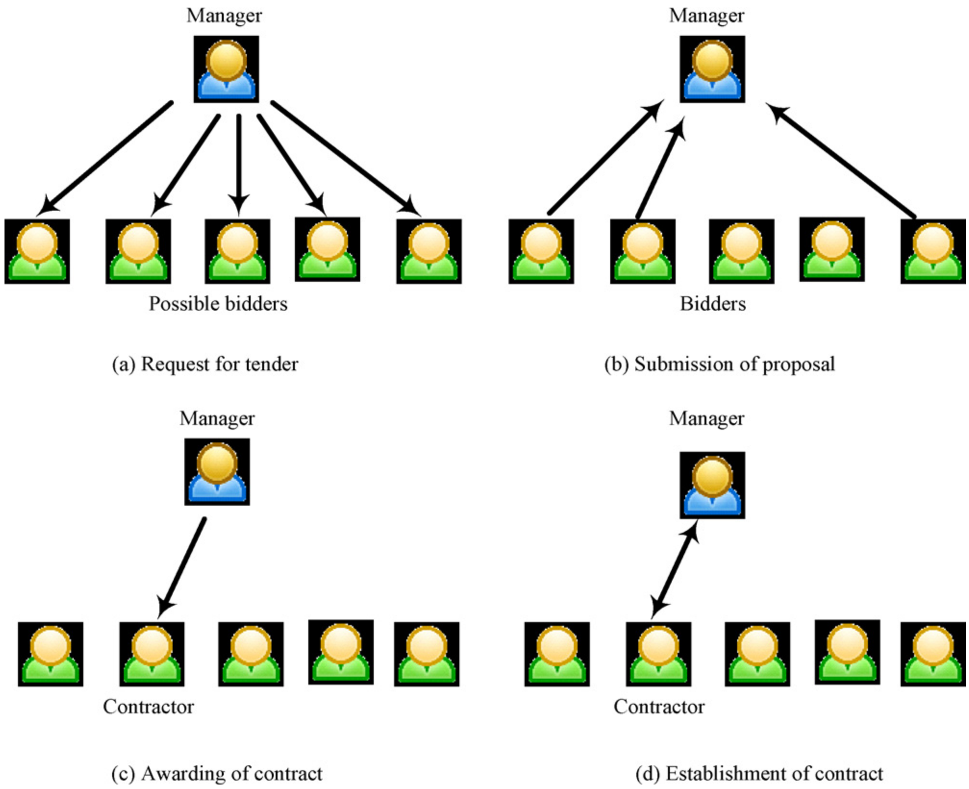


Fig. 10. The four stages of the contract net protocol (adapted from [24]).

through a mechanism of negotiation and selection of the best bidder. Thus, a negotiation stage is introduced between stages (2) and (3) into the original *contract net protocol*. The following paragraphs describe the project manager interaction in order to establish a client–server relationship and it is shown by a sequence diagram in Fig. 11.

While a project plan is created or repaired, the project manager defines a set of resources that is needed to execute the project tasks. Then, she/he has to procure and compete to get the best supplier of needed resources. For that, the project manager sends $RequestForResource(R_{resource}, PM_{project\text{-}manager})$ messages to all concerned means-managers. These messages describe capabilities and skills of the requested resource.

On the basis of resource capabilities or skills that are requested by an ends-manager and resource availability agenda, each contacted means-manager draws up a proposal or an answer which will be submitted to the client (ends-manager). Each means-manager can send the corresponding ends-manager either an $Acknowledge(R_{resource}, MM_{means\text{-}manager})$ message when the means-manager can answer the resource request as it is or a $Propose(R_{resource}, P_{proposal}, MM_{means\text{-}manager})$ message if the resource request can be satisfied by the means-manager but with some changes to its original conditions. A $NotInterested(R_{resource}, MM_{means\text{-}manager})$ message may be issued if the means-manager is not interested in answer the resource request, or simply the means-manager cannot send any positive answer.

For a given period of time, the project manager receives answers and proposals. Then, the received responses are analyzed and ranked to select the best alternative. In order to contract the best server, it is possible for the ends-manager to negotiate conditions for any resource request with one or more means-managers. During this negotiation, the ends-manager sends $Propose(R_{resource}, P_{propose}, PM_{project\text{-}manager})$ messages if it is necessary to introduce a change in any condition of the resource request or $Query(R_{resource}, Q_{query}, PM_{project\text{-}manager})$ messages for any query. These messages are answered by the corresponding means-manager.

When the project manager has defined a ranked list of means-managers, he/she sends an $Award(R_{resource}, PM_{project\text{-}manager})$ message to the means-manager which has been awarded the contract. The ends-manager looks forward to an answer from selected supplier. A negative answer from this means-manager triggers a re-evaluation of the remaining candidates and the awarding of the contract to another means-manager. The awarded server evaluates the contracts. If the means-manager can still schedule usage of the required resource, it marks the selected time interval in the resource agenda as occupied and sends an $Accept(R_{resource}, MM_{means\text{-}manager})$ message to ends-manager and a client–server relationship is established between them, or else that if it cannot carry out the contract, it sends a $Refuse(R_{resource}, MM_{means\text{-}manager})$ message.

In order to establish a delegation-to-do relationship between two project managers the same protocol that has been described for establishing client–server relationships is used. As it has been described previously, a project manager who is candidate to play an ends manager role of a sub-project accepts to achieve a sub-goal within the time framework stated by the manager of the father project.

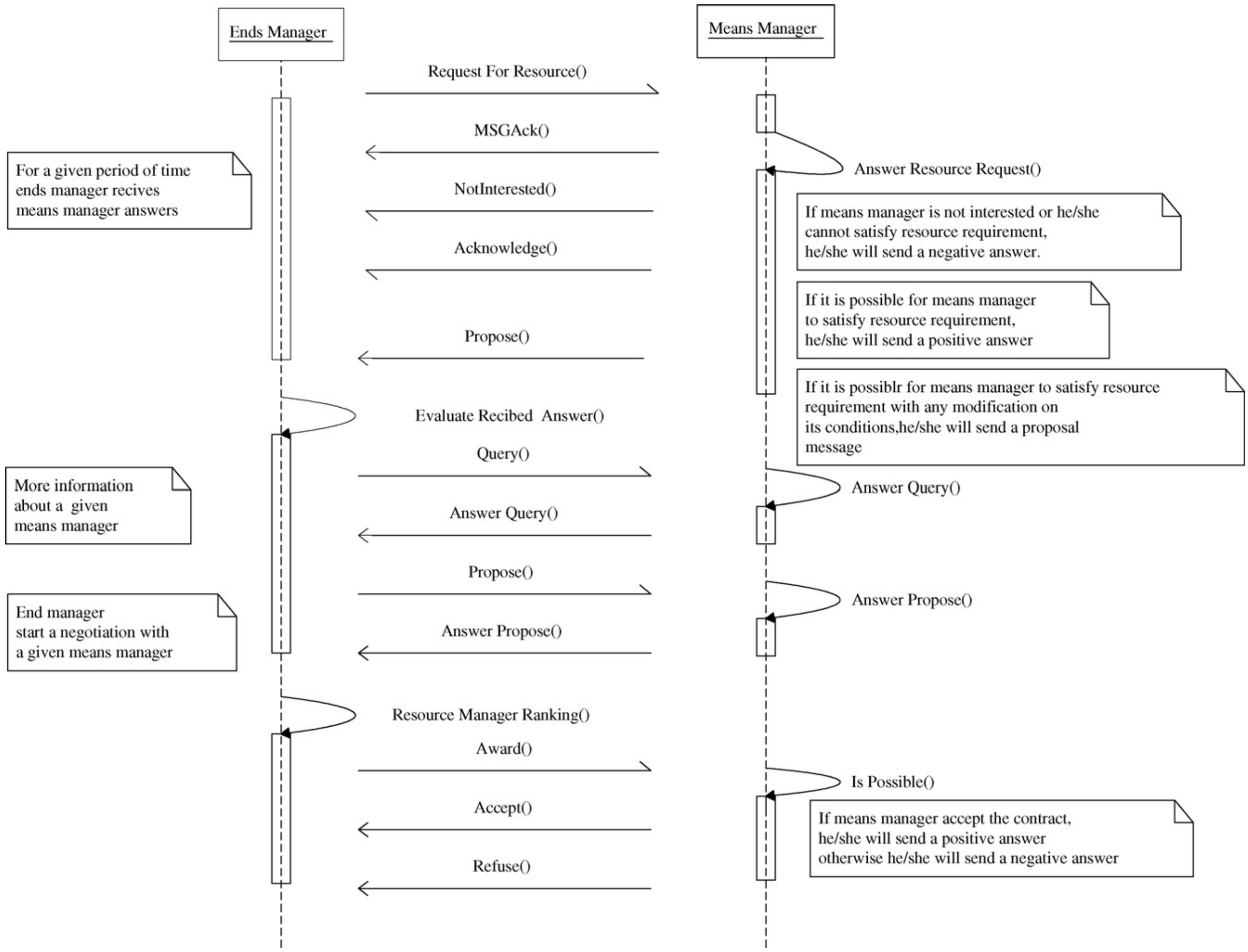


Fig. 11. Sequence diagram of the protocol to establish client–server relationships.

*2.2.3.2. Interactions after relationships among project managers are established.* After a relationship among project managers is established, project managers exchange messages until the relationship becomes extinct. In a delegation-to-do relationship, the involved ends managers exchange messages primarily when:

- A child project manager informs the father project manager about its project state.
- A father project manager asks information about sub-project situation from any of its child project managers.
- A father project manager sends to its sub-project managers information about coordination between sub-projects and owner tasks.

In the case of a client–server relationship, an ends-manager and a means-manager exchange messages mainly when:

- A means-manager informs to an ends-manager that the assigned resources are ready to be used or consumed by a given project task.
- An ends-manager notifies the corresponding means-manager that assigned resources are grabbed. Then, when the resource usage has finished, the ends manager notifies means manager that assigned resources are available.
- At the beginning or during the resource usage, a ends-manager can inform a given means-manager about the occurrence of an unexpected event, abnormal situation or failure in the assigned resource(s).
- A means-manager notifies ends managers involved in the assignments of out-service resources.
- An ends-manager can ask a means-manager about situation of the assigned resources. Also, the means-manager can ask the ends-manager about the resource situation which is been used by a given project task.

To illustrate the above-mentioned interactions, the following paragraph describes an example of messages exchanged among project managers from the moment an agreement is reached for using a resource in a project task until the resource is effectively relinquished for further use elsewhere. Fig. 12 shows the sequence diagram of this interaction.

After the set up task on a given resource has finished, the means-manager informs the corresponding ends-manager that the assigned resource is ready to be used or consumed by the project. Based on this information, the ends-manager updates

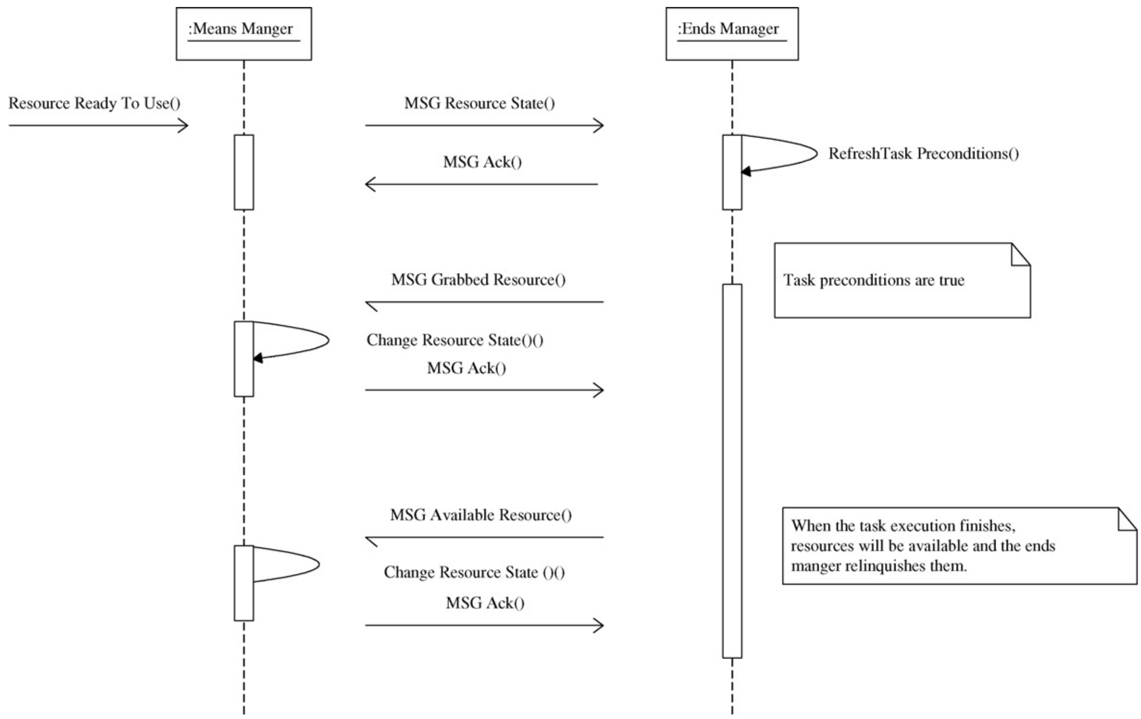


Fig. 12. Actor interaction in a client–server relationship.

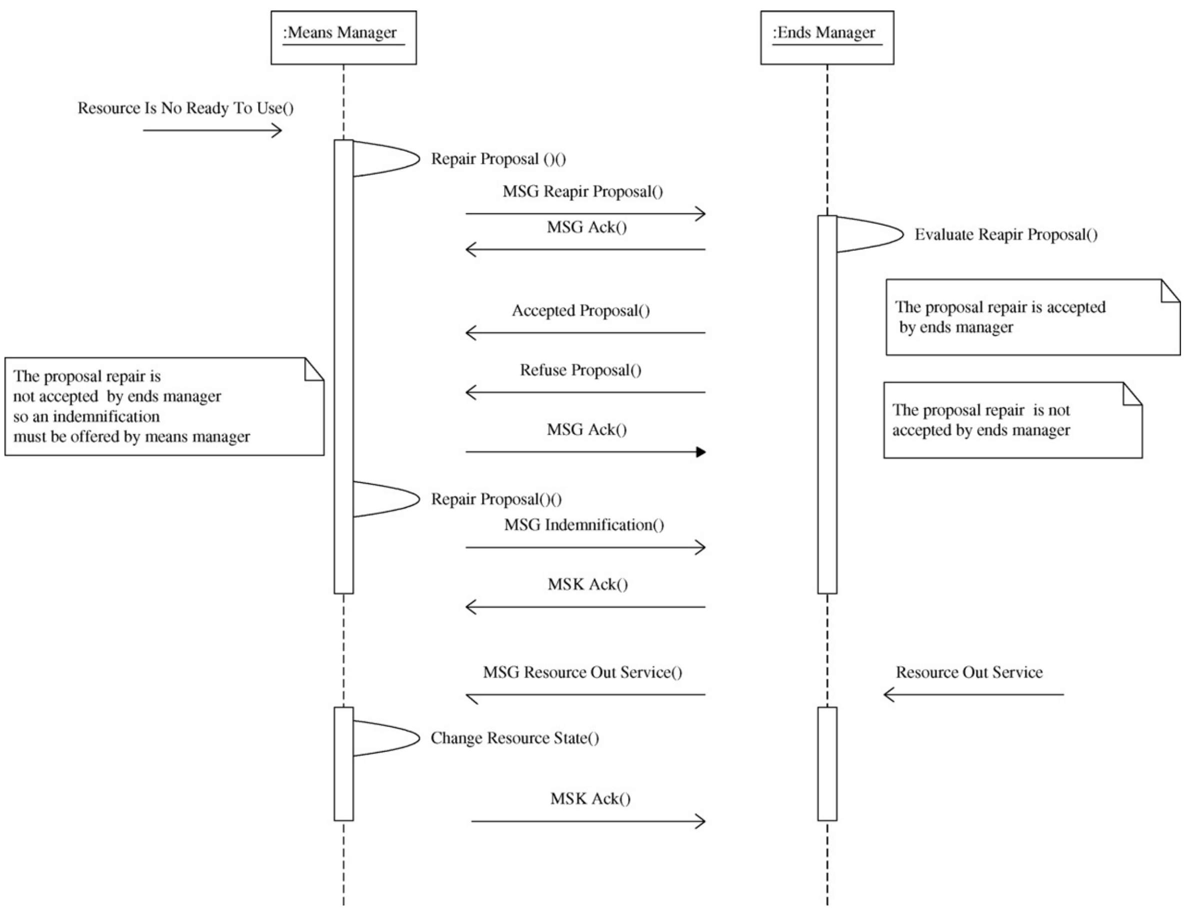


Fig. 13. Actor interaction during a client–server relationship in abnormal resource situations.

the task precondition states and immediately she/he sends a *MSGGrabResource*($R_{resource}$, $PM_{project\text{-}manager}$) message to the concerned means-manager. Thus, the means-manager knows that the resource is being used by the project. The resource usage is assigned to a given ends-manager for a specific period of time. At the end of it, the ends-manager has to relinquish the resource by means of a *MSGAvailableResource*($R_{resource}$, $PM_{project\text{-}manager}$) message. Thus, the resource is available for allocation to other tasks. In all interactions the recipient ends-manager sends a MSGAck($M_{msg}$, $PM_{projectManager}$) to notify the message sender on its reception.

If the resource usage period is over and a given ends-manager has not sent a MSGAvailableResource($R_{resource}$, $PM_{project\text{-}manager}$) message to the concerned means-manager, the latter can request information about resource situation from the ends-manager. In this case different resource situations are possible. For example, the resource may be available and the ends-manager only has to send a *MSGAvailable*($R_{resource}$, $PM_{project\text{-}manager}$) message to the corresponding means-manager and the abnormal situation is immediately repaired. There may be the case the resource is still being used by a given project task, thereby the ends-manager begins a negotiation with the corresponding resource manager in order to extend resource usage period sending a new *RequestForResource*($R_{resource}$, $PM_{project\text{-}manager}$) message. The means-manager must repair its resource assignment plan in order to satisfy this new requirement. If it is possible for the means-manager to agree the extension request a *MSAceptedRequest*($R_{resource}$, $MM_{means\text{-}manager}$) is sent. Otherwise, the means manager has to assess the impact of delaying next resource assignments and she/he can penalize the ends manager for non-compliance with the agreement.

Finally, while a resource is been used by a given project task, there may occur an event that may cause the resource is temporally out of service. In this case, the ends-manager has to inform the means-manager about the unplanned resource state and negotiate additional resources with the same skills and capabilities. At the same time, if it is not possible for the means-manager to satisfy future resource assignment without modification, she/he has to notify all other actors involved about the resource situation and propose a new resource assignment or negotiate the terms for non-compliance agreement. Fig. 13 shows actor interaction in the abnormal situations described previously.

## 3. Case study

The project-based fractal model is now exemplified by a hypothetical case study related to process development of new pharmaceutical products in a specialty chemical enterprise network. First, a brief description of a drug development process is presented. Later on, roles and

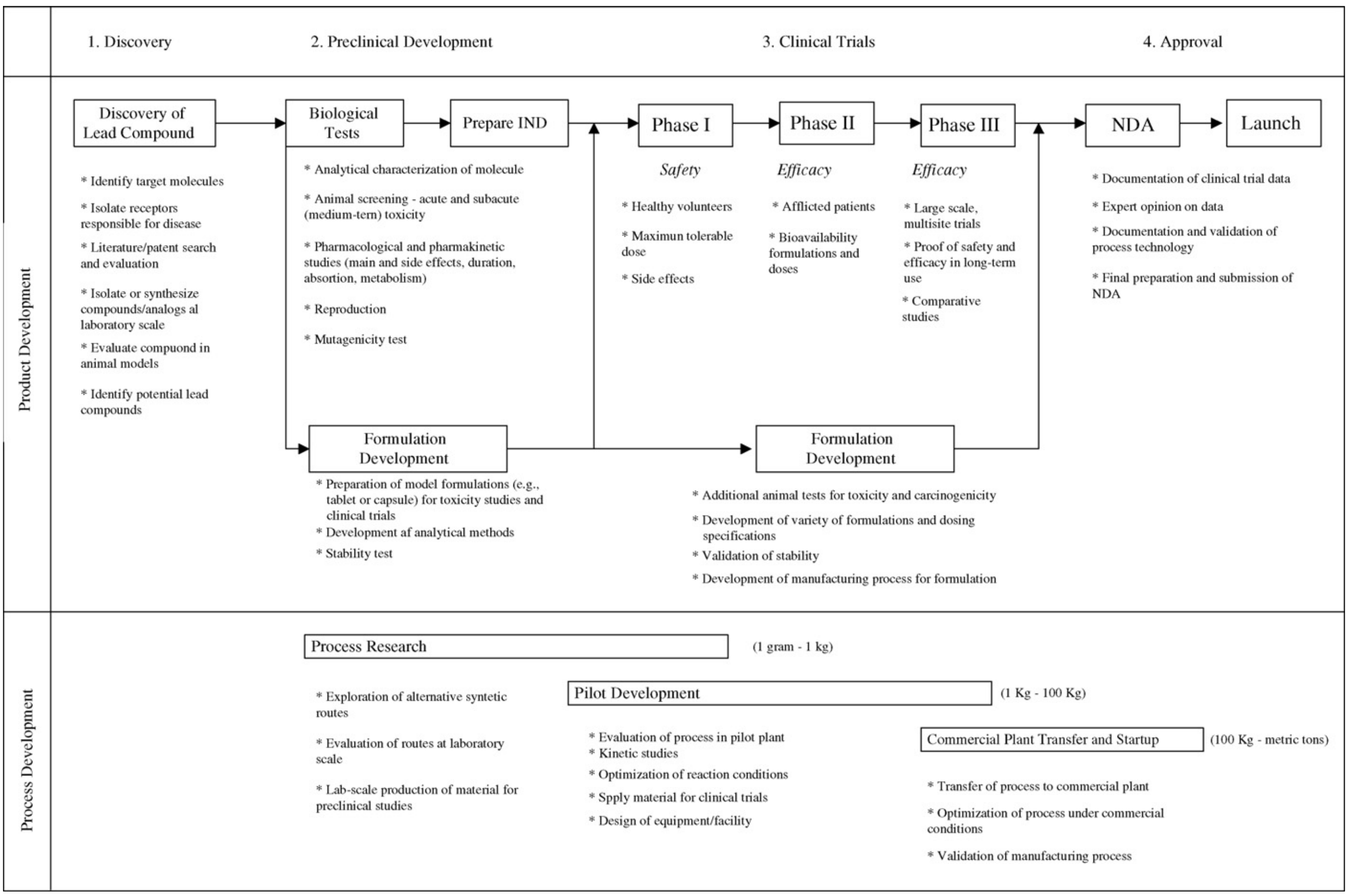


Fig. 14. Discovery and development of a new chemical entity [26].

relationships defined in the proposed fractal company model are discussed.

In the pharmaceutical industry the product development cycle can typically extend for about 10 years and it can be divided into four stages [26]: discovery, preclinical development, clinical trials, and regulatory approval (Fig. 14). During the discovery stage, molecules that may prove safe and effective in the treatment of a given disease are identified. Once a molecule is discovered or synthesized, the preclinical research phase begins, with the goal of gaining information about the molecule safety and therapeutic properties. To this aim, preclinical analysis in both test tubes and laboratory animals are performed. The next major stage of drug development is human clinical trials. Once a company has reasonable confidence in a compound's safety and therapeutic benefits, it can begin test on human patients. During the Phase I clinical trials the new drug is administered to a small sample of healthy volunteers using an experimental design aimed at determining side effects. Phase II trials are designed to determine the appropriate dosage regime and form (e.g. crystal shape) for the drug. In general, the end of Phase II trials marks an important project milestone. At this time, the company should be relatively confident that the drug has no serious side effects and know which doses are most effective. So the company makes a ''go or no-go'' decision onto Phase III. Phase III trials, which involve head-to-head comparisons of the drug against placebos or existing drugs in a large sample of patients, are by far the most costly phase of human clinical trials. After Phase III trials are completed, the company must submit its clinical data for review to the regulatory entity. With the preclinical development stage begins process development which involves three phases: (1) *process research*, in which the basic process chemistry is explored and chosen; (2) *pilot development*, in which the process is run, evaluated, and refined in an intermediate-scale pilot plant; and (3) *technology transfer and startup*, in which the process is transferred and adapted to the commercial manufacturing site.

The case study involves small and medium chemical, pharmaceutical, and biotech companies that participate as managers of the goal achievement or server providers in different projects. Resources involved in these projects are batch plants, pilot plants, laboratories, scientists of different discipline (chemists, formulation analyst, biologist, etc.).

It is often the case that several projects are active simultaneously with different degrees of advance. Several companies are linked temporarily to undertake the discovery and development of new drugs. At any time, new projects can begin so new enterprise networks will be defined for each of them.

In order to reduce project complexity and allow the concerned manager to control its progress, projects can be broken into smaller projects associated with synthesis research, process development and the production of small batches to satisfy preclinical or clinical trials. In turn, these projects can be broken into sub-projects associated, for instance, with setting up a pilot plant. Each new sub-project is delegated to and managed by other project managers. Then, active projects

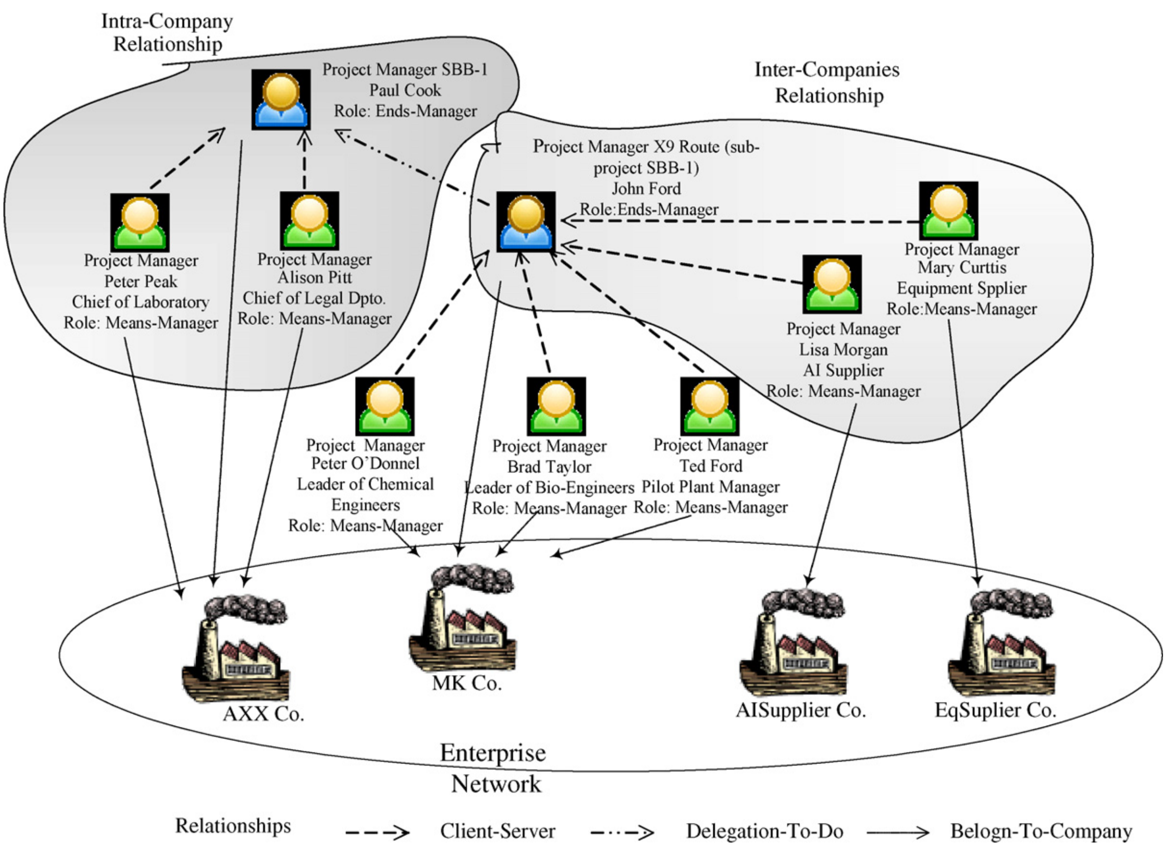


Fig. 15. Delegation-to-do, client–server, intra- and inter-companies relationships among project managers.

require resources in order to perform each project task plan. The required resources can be internal or external to the companies. For this case study resources can be laboratories required to make preclinical tests, pharmacological and pharmakinetic studies, small chemical or multipurpose plants that allow producing batches for clinical trials, high-tech start up companies, etc. All of them run by different project managers playing the means-manager role. As mentioned before, a drug development project may extend for about 10 years and has several important milestones when the project owner must decide if the project continues or not, so the resource deals and sub-project delegations between project managers are conditioned by successful project evolution over time.

Briefly the following paragraphs describes project SBB-1 life-cycle and discusses delegation-to-do and client–server relationships established among project managers of different small and medium companies of chemical, pharmaceutical, and biotech sectors. The project goal is to perform pre-clinical, clinical trials of SBB-1 (an anti-infective compound) and research process development of the drug in order to achieve official approval to launch the new drug to market. This project is managed by Paul Cook who belongs to AXX Co. The project includes a sub-project called "*Development and scale-up X9 route*" that was delegated to MK Co., a medium size biotech company. Also, Paul Cook has established client–server relationships with different resource servers, for instance, Peter Peak, laboratory chief of AXX Co, where analytical test

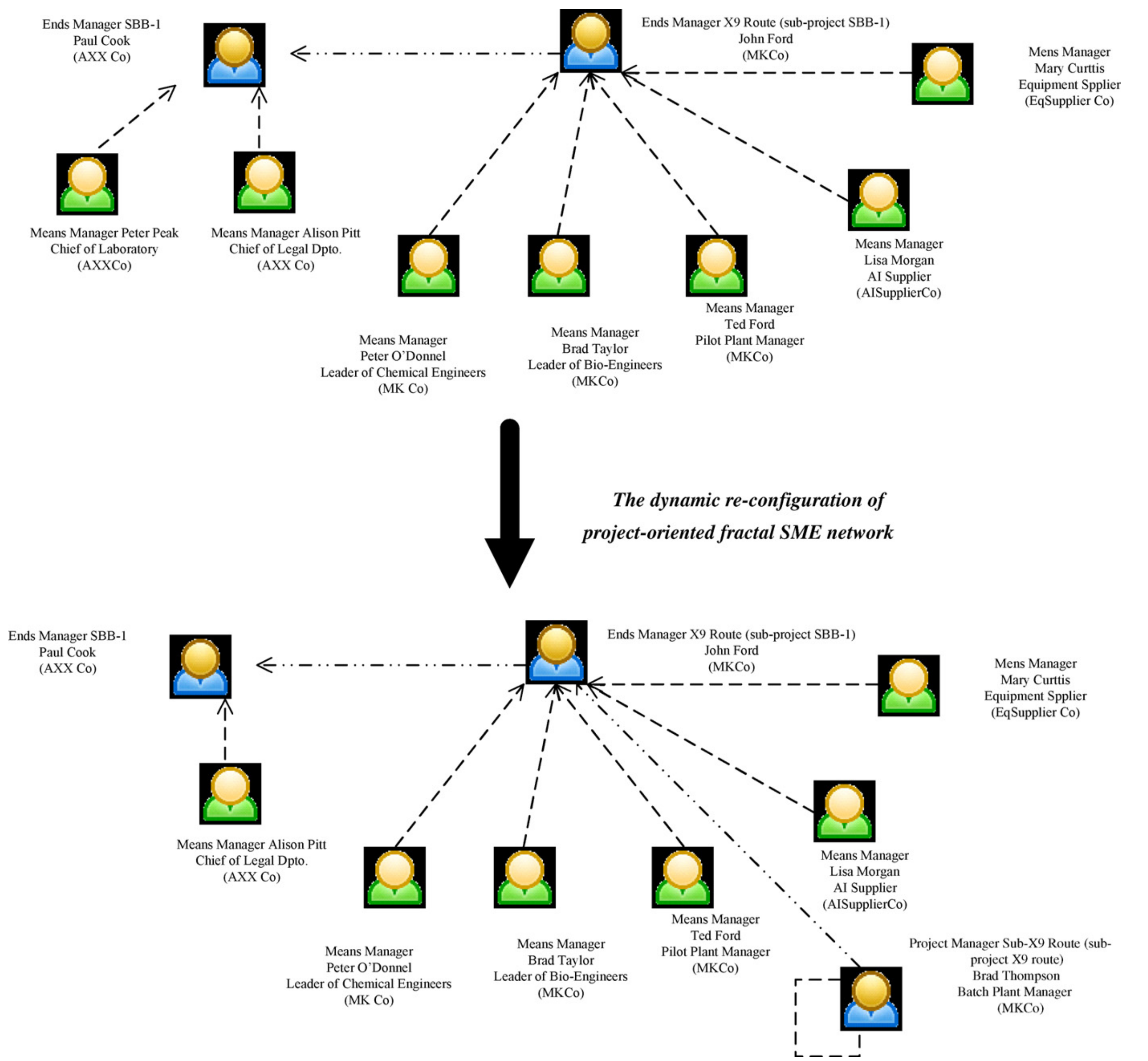


Fig. 16. SME network re-configured after the SBB-1 project plan was repair and new relationships among actors were established.

and animal experiments for initial evaluation of SBB-1 and the preclinical development will be performed and Alison Pitt, Chief of Legal Department of AXX Co who will staff to "*Submission Official Approval*" task.

An MK Co. personnel, named John Ford, is the manager of the project "*Development and scale-up X9 route*". The goal of this project is to carry out process development, scale it up and provides the drug samples for preclinical and clinical trials. Thus, a delegation-to-do relationship is established from Paul Cook to Peter Peak (Fig. 14). In turn, Peter Peak defines a project plan without sub-projects but he establishes several relationships with different resource servers that belong to different companies. This project needs resources, such as specialized personnel (chemical engineers, bio-engineers, etc.) pilot plants, batch plants, row material, active ingredient suppliers, and equipment suppliers. The temporal network of relationships among project managers is shown in Fig. 15. Also, this figure shows intra-company relationships defined between actors Cook, Peak and Pit who belong to AXX Co in the SME Network and inter-companies relationships between actors Ford, Morgan and Curtis who belong to MK Co., AISupplier Co. and EqSuplier Co., respectively.

While the SBB-1 project progresses, several problems occur and the manager must make decisions in order to achieve the project goal. For instance, if clinical trials were very successful so the clinical trial team and formulation team demand increasing amounts of the drug in order to progress their work further, but the development project is facing difficulties in satisfying such demands. Therefore Peak, the development project manager, has to repair the project plan in an attempt to solve the present problem. In order to satisfy demand of trial and formulation groups, batches of SBB-1 compound are

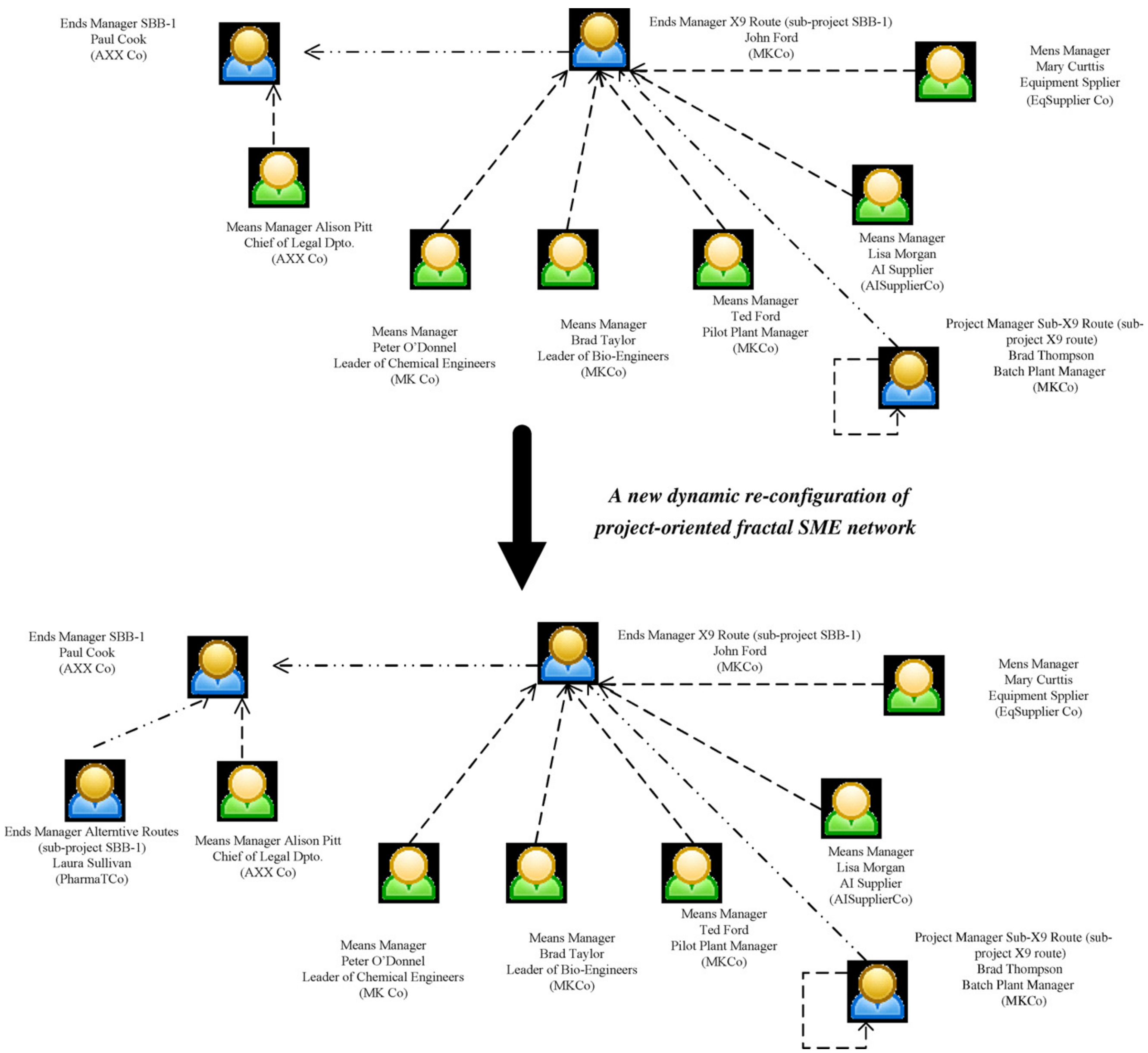


Fig. 17. SME network re-configured after the project to research alternative synthesis route was created.

manufactured following a set of steps called synthesis route. The synthesis route of SBB-1 compound involved 23 steps and it is called X9. When the project manager repairs this synthesis plan decides to split the manufacturing task of product batches for trials. Thus, the last three steps of X9 route remain in the same pilot plant and the first 20 steps of the X9 route are transferred to the MK Co.'s large-scale pilot facility. Also, the chemical engineering group of this pilot plant assumes responsibility for improving this portion of the production process. Thus, a new sub-project is included in the SBB-1 project that is managed by the chief engineer Brad Thompson who belongs to MK Co. Engineering Division. In this way, a new delegation-to-do relationship is established between Peter Peak and Brad Thompson who in turn, defines a plan for the new project and allocates plant resources accordingly. Fig. 16 shows the fractal company reconfiguration.

After a while, the project moves into Phase III trials. Clinical results are excellent but the manufacturing process has still serious problems, so Cook decides to research alternative synthesis routes. Since most of the technical resources for the project has been consumed attempting to optimize the X9 route, Cook seeks to form an alliance with other small or medium pharmaceuticals interested in developing alternative synthesis routes. Therefore, AXX Co. and PharmaT Co. cut a deal to research and develop alternative synthesis routes for SBB-1. Thus, the developing of alternative synthesis routes is a new sub-project in the Paul Cook's project plan and a new project for PharmaT Co. Laura Sullivan is named as its project manager. Again, the temporal relationship network is modified by new delegation-to-do and client–server relationships and a new enterprise is added to the enterprise network (Fig. 17).

About a year earlier, some preliminary research has been started on alternative synthesis routes. The 19-step X10 route looks promising. The X9 route continues with unsolved problems and Paul Cook knows that this route would never be commercially feasible so, although X10 route has been run only at small scale and he knows it has problems that will be able to solve, Cook decides to adopt X10 route, aborts all task related to X9 route and re-assigns economic and technical resources from the aborted sub-project to sub-project X10 route. Also, Cook gets additional funding that allows increasing resources for X10 route tasks and launching SBB-1 at the scheduled date. Thus, new relationships are established and the project plans are repaired again in order to achieve the project goals.

## 4. Concluding remarks

An enterprise model of a fractal management system for SMEs networking using projects has been proposed. In this model, each project is an autonomic, self-similar, self-optimized and self-organized unit within the SMEs network. The key to the project-based fractal company is establishing client–server relationships between ends-managers and means-managers. The fractal management unit is made up of concepts like: goals, plans, roles, project managers and their relationships.

The project-oriented fractal company makes important contributions to the development of the application of the fractal company idea for SMEs networking. First, the model proposes a fractal unit of management for SMEs networking. Also, the model defines the internal structure (project manager and managed object), the behavior and interactions among management fractal units or projects. Furthermore, the novel concept of client–server and delegation-to-do relationships between actors allows a more effective device for collaborating in a competitive environment.

In order to describe and represent rigorously and accurately the main structures and relationships, interaction, roles, goals, behavior of actors and constraints within project-oriented fractal company for SMEs networking, the proposed enterprise model has been formalized using a logical language [31,32]. The resulting set of axioms representing the network dynamics is implemented in the logical programming language $ECL^{i}PS^{e}$ Prolog [33]. The execution of model simulation allows us to describe and analyze emergent behaviors and constraints as well as elaborated queries to the fractal company knowledge base.

A hypothetical case study related to process development of new pharmaceutical products in a SME network of specialty chemical sector was used to illustrate the proposed project-based fractal model. Also, this case study exhibits fractal characteristics of each project (self-similary, self-optimization, self-organization), highlighting the possibility of dynamic re-configuration in order to exploit business opportunities and event handling.

As the proposed enterprise model can be used to specify a detailed design for information systems in a network of SMEs, our present research work is to develop a prototype of the project-based fractal company for SMEs network using the Enterprise Project Management Solution$^{TM}$ of Microsoft Project 2003. This commercial set of project management tools is being tailored in order to describe the interactions between project managers according to the proposed model for enterprise networking.

Finally, the project-oriented fractal company can be seen as a multi-agent system, where each project manager is an agent who has capacity to perceive his environment, to make decisions and to act with the objective to achieve his goals. Accordingly, our research work is now driven towards the design and implementation of algorithms of reinforcement learning [34] that will specify the detailed behavior of each project manager. On this basis, the collective performance resulting from the interactions of multiple project-manager agents will allow the discovery of emergent behaviors (both desired and undesired) during the learning process of a SME network.

## References

[1] I. Mezgar, G. Kovacs, P. Paganelli, Co-operative production planning for small- and medium-sized enterprise, International Journal of Production Economics 64 (1–3) (2000) 37–48.

[2] E. Westkamper, M. Huser, R. von Briel, Transition processes: a survey of German industry, Journal of Materials Processing Technology 106 (2000) 141–151.

[3] H. Gou, B. Houang, W. Liu, X. Li, A framework for virtual enterprise operation management, Computers in Industry 50 (2003) 333–352.
[4] K. Mallidi, A.T. Paraskevopoulos, P. Paganelli, Process modeling in small-medium enterprise networks, Computers in Industry 38 (1999) 149–158.
[5] N. Okino, Bionic manufacturing systems—Modelon-based approach, in: Proceeding of the CAM-I 18th Annual International Conference, New Orleans, Louisiana, (1989), pp. 485–492.
[6] N. Okino, A prototyping of bionic manufacturing systems, in: Proceeding of the International Conference on Object-Oriented Manufacturing Systems, Calgary, Alberta, (1992), pp. 297–302.
[7] H. Van Brussel, J. Wyns, P. Valckenaers, L. Bongaerts, P. Peeters, Reference architecture for holonic manufacturing system PROSA, Computers in Industry 37 (1998) 255–274.
[8] P. Valckenaers, H. Van Brussel, J. Wyns, L. Bongaerts, P. Peeters, Designing holonic manufacturing systems, Robotic and Computer-Integrated Manufacturing 14 (1998) 455–465.
[9] K. Fischer, Agent-based design of holonic manufacturing systems, Robotics and Autonomous Systems (27) (1999) 3–13.
[10] B. Huang, H. Gou, W. Liu, Y. Li, M. Xie, A framework for virtual enterprise control with the holonic manufacturing paradigm, Computers in Industry (43) (2002) 299–310.
[11] H.J. Warnecke, The Fractal Company: A Revolution in Corporate Culture, Springer-Verlag, Berlin, 1993.
[12] A. Tharumarajah, A.J. Wells, L. Nemes, Comparison of the bionic, fractal and holonic manufacturing concepts, International Journal of Computer Integrated Manufacturing 9 (1996) 217–266.
[13] T. Tirpak, S. Daniel, J. LaLond, W. Devis, A note on a fractal architecture for modeling and controlling flexible manufacturing systems, IEEE Transactions on Systems, Man, and Cybernetics 22 (3) (1992) 564–567.
[14] E.C. Martinez, Project-oriented management of batch plants. Information system design, Computers and Chemical Engineering Supplement (1999) S711–S714.
[15] E.C. Martinez, Systems thinking and functional modeling of batch process management using projects, Computers and Chemical Engineering 24 (2000) 1657–1663.
[16] M.S. Fox, M. Gruninger, Enterprise Modelling, AI Magazine, AAAI Press, 1998 pp. 109–121, Available at http://www.eil.utoronto.ca/EIL/publiv/aimag98.ps.
[17] M. Koubaraskis, D. Plexousakis, A formal framework for business process modeling and design, Information Systems 27 (5) (2002) 229–319.
[18] B. Mandelbrot, Fractal: Form, Chance, and Dimension, W.F. Freeman, New York, 1978.
[19] J. De Rosnay, The symbiotic man. A New Understanding of the Organization of Life and a Vision of the Future, McGraw Hill, 2000.
[20] K. Ryu, Y. Son, M. Jung, Modeling and specifications of dynamic agents in fractal manufacturing systems, Computers in Industry 52 (2003) 161–182.
[21] B. Schweyer, A. Haurat, Information system design using a project approach, Journal of Intelligent Manufacturing (8) (1997) 15–29.
[22] E.C. Martinez, G.A. Perez, A project-oriented model of batch plants, Computers and Chemical Engineering 22 (3) (1998) 391–441.
[23] R.G. Smith, The contract net protocol: high-level communication and control in a distributed problem solver, IEEE Transactions on Computers 29 (12) (1980) 1104–1113.
[24] J. Ferber, Multi-Agent Systems. An Introduction to Distributed Artificial Intelligence, Addison-Wesley, 1999.
[25] J. Rumbaugh, G. Booch, I. Jacobson, The Unified Modeling Language (UML) ver 1.0, Rational, CA, USA, 1997.
[26] G. Pisano, The Development Factory, Harvard Business School Press, 1997.
[27] W. Sihn, Fractal business in an e-business world, in: The Eighth International Conference on Concurrent Enterprising, Rome, Italy, 2002.
[28] P. Massote, R. Bataille, Future production systems: influence of self-organization on approaches to quality engineering, International Journal of Production Economics 64 (2000) 359–377.
[29] K. Ryu, Y. Son, M. Jung, Modeling and specifications of dynamic agents in fractal manufacturing systems, Computers in Industry 52 (2) (2003) 161–182.
[30] Walsh, P., Koumpis, A., 2000. Managing Information Supply Chains ACM's Ubiquity magazine, Issue 39.
[31] M. Canavesio, E. Martinez, A formal enterprise model for project-oriented management of batch plants, in: Proceeding of FOCAPO 2003 Foundations of Computers-Aided Process Operation 2003, Coral Sping Florida EE.UU, (2003), pp. 501–504, ISBN: 09654589110.
[32] M. Canavesio, E. Martinez, A fractal model for enterprise networking in the specialty chemical industry using projects, in: Proceeding of Second Mercosur Congress Chemical Engineering and Fourth Mercosur Congress on Process System Engineering, Río do Janeiro, Brazil, (2005), pp. 1–10.
[33] http://eclipse.crosscoreop.com.
[34] R. Sutton, A. Barto, Reinforcement Learning: An Introduction, MIT Press, Bradford Book, Boston, MA, 1998.
[35] P. Valckenaers, H. Van Brussel, Holonic manufacturing execution systems, CIRP Annals (2005) 427–432.

**M. Mercedes Canavesio** received an engineering degree in information systems in 1996 from the Technical University of Argentina (UTN). She is a now candidate for the PhD degree in information systems engineering at the same university. She is also a teaching assistant to the core course "Systems and organizations" for students of Systems Engineering. Her research interests are enterprise modeling, multi-agent systems, systems thinking and modeling of artificial societies.

**Dr Ernesto Martínez** is a Research Fellow of The National Research Council of Argentina (CONICET) and Chair Professor of Systems and Organizations in the Technical University of Argentina (UTN). He has published more than 40 journal papers and book chapters on systems modeling, control and optimization. He has been a visiting professor at Lehigh University (USA), Nottingham University (UK) and Universidad de Valladolid (Spain). Awards received include the Senior Research fellowship from the Fulbright Commission (USA), Repsol-YPF 2000 Innovation award and The Ramón y Cajal Distinguished Fellowship award by the Ministry of Science and Technology of Spain. His main interests are intelligent system optimization and control, learning algorithms, multi-agent systems, complex system supervision, enterprise modeling and generative modeling of artificial social systems.