# Business Process Models

**Aphrodite Tsalgatidou and Maria Anastasiou**
Department of Informatics and Telecommunications
University of Athens,
TYPA Buildings, Panepistimiopolis
GR-157 71 Ilisia Greece
afrodite@di.uoa.gr

In many fields of science, a situation or a system is not studied directly but indirectly through a model of the situation or the system. Many definitions have been given about what a model is. Beer defines a model as "a representation of something else, designed for a special purpose". According to Wilson a model "is the explicit interpretation of one's understanding of a situation, or merely of one's ideas about a situation. It can be expressed in mathematics, symbols, or words, but it is essentially a description of entities, processes or attributes and the relations between them. It may be prescriptive or illustrative, but above all, it must be useful". A model is useful when it facilitates the description and the understanding of a complex situation.

There are various reasons why we use models. We model an existing system in order to understand and study it without preventing its operation. Models are also used to examine a risky situation is to be examined something that cannot happen in real. Finally, a model could be used to describe and analyse a future system before its implementation.

In business field, business process modelling refers to a collection of techniques that are used to model the behaviour of business systems, which is expressed in terms of processes. Managers and system analysts use business models in order to deal with the complex and dynamic nature of modern organisations. Models are used throughout the lifecycle of a process, supporting its definition, (re)design, implementation and continuous improvement. Therefore, modelling a business process should be considered as a continuous, endless activity similar to monitoring it, rather than an activity with a predefined end point.

Although the development and maintenance of business process' models is a difficult, time consuming, and costly project, there are very important reasons justifying the necessity of a modelling process, and great benefits to be obtained through it. The most important of them are following presented:

- A well-defined model facilitates understanding of the process by every process's participant.
- When a process model permits the decomposition of a complex business process at different level of detail, it enables focusing in the required level of interest, excluding details of lower levels.

- A model can show where processes cross organisation's boundaries, extend outside organisation, participate in product distribution or in providing services to customers.
- Given the right notation, a model can be used to simulate the process in order to analyse it and to examine "what-if" scenarios.
- Models with formal syntax and well-defined semantics can be used to support process enactment.

Existing business process modelling approaches mainly originate from the software-engineering field. Several classifications for business process models have already been suggested in modelling bibliography. In following we briefly describe business process models that fall into the following categories:

1. Activity oriented models that describe a process as a set of ordered activities (e.g. IDEF0, DFDs, EPC).
2. Petri nets that also describe the activities of a process, but are generally treated differently than other activity oriented models.
3. Agent-oriented (or role oriented) approaches that specify and analyse the role of the agents that participate in the process (e.g. Role Activity Diagrams).
4. Goal-oriented models that support goal-based work (e.g. Action workflow model, the i* framework)

Using a model or not is not that important as the way it is used. The kind of model that will be used in order to describe a business process is also important. The decision depends on the kind of process under study as well as on the phase of the process lifecycle that the model will support.

The functionality of process models varies throughout the different stages of a BPR effort. At the beginning they should facilitate process participants and BPR team's understanding as well as enable the communication between them. Following, they help analysing current processes and revealing existing problems. During the design phase, they describe the future process and should provide the basis for the evaluation of the alternative design solutions. Finally, they could be used to implement the new process.

In this chapter we examine the requirements that a process model should fulfil in order to be successfully used in a BPR effort as well as a set of criteria for the model selection process. The applicability of a model in a BPR effort is evaluated against the following criteria:

- *ease of use* - how easily a modeller can learn to use the model and how easily a process can be modelled
- *comprehensibility* - whether the model enhances process understanding by the process participants and the BPR team, as well as whether it enables the communication between them
- *originality* - whether the model was developed for business process modelling or not
- *formal syntax and semantics*- whether the model has a formal syntax and well defined semantics

- *expressiveness* - whether the model can represent the relationships between the activities of the process and efficiently express issues such as iteration, time relationships, path selection, conditions e.t.c.
- *hierarchical decomposition* - whether the model enables the decomposition of process models and the construction of model hierarchies
- *analytical capabilities* - Whether the model can be used to analyse the process and what kind of analysis it supports.

## *1. Activity – oriented Models*

Activity - oriented models focus on representing the way the work is done in a process or what activities take place, rather than why things are done in this particular way and what goals should be achieved. The main modelling entities of activity-oriented models are:

- activity: a step executed by an individual, or a group of people, or a machine
- control flow: the execution order of activities
- resource: a necessary object for the execution of an activity
- resource flow: the path of resources between activities
- role: a set of responsibilities for a person or organisational unit
- organisational structure: organisational units, roles, people etc.

The IDEF0 and EPC models are representative examples of activity models and are next described.

### 1.1. IDEF0 Models

The IDEF0 Function model is designed to model the decisions, actions, and activities of an organisation or system. It was derived from SADT (Structured Analysis and Design Technique) which was developed by SoftTech Corporation in the 1970's. The Air Force commissioned SADT developers to develop a function modelling method for analysing and communicating the functional perspective of a system. The IDEF0's objective is to efficiently describe complex business processes and systems as well as to promote effective communication between the modellers and the process' participants.

The IDEF0 model provides structured format and syntax to facilitate user communication. An activity is described by a box labelled with a verb. Each activity may have inputs, controls, outputs and mechanisms (ICOMs). Figure 1 shows the basic syntax of an IDEF0 model. Several activity boxes and related Inputs, Controls, Outputs, and Mechanisms construct an IDEF0 model. Thus, an IDEF0 model describes the individual activities of a process as well as the interelationships between these activities: for example the output of one activity may be the input, control, or even mechanism of another activity within the same model.

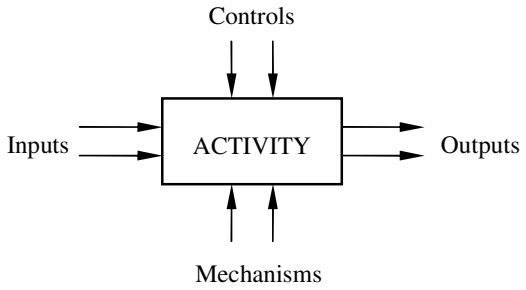



***Figure 1***. IDEFo Syntax

The IDEFo model supports the hierarchical decomposition of activities. A box in an IDEFo model represents the boundaries of an activity. Inside the box there is the decomposition of this activity in sub activities. This hierarchical structure supports increasing level of detail by hiding unnecessary complexity and revealing it when is needed to enable a better and deeper understanding.

In this way the first step in constructing an IDEFo model is to create a Context Diagram. A Context Diagram limits the scope of the modelling effort. It represents the function, inputs, outputs, controls and mechanism at the highest level. Then a functional decomposition of the context diagram follows.

It should be mentioned that an IDEFo model just depicts what an organisation does, but it does not support the specification of how the work is done in terms of specific logic or the timing associated with the activities. The abstraction away from timing, sequencing and decision logic allows concision in an IDEFo model. In fact, the model's lack of reliance on time is the most powerful feature of it, since it provides an objective assessment of what actually occurs in the process.

IDEFo's standard notation, it's hierarchical structuring and the effective rules for decomposition facilitates learning and understanding of the modelled process. One problem with the IDEFo models is that sometimes they are so concise that it is difficult for someone non-expert to understand them. Moreover, the abstraction away from timing, sequencing, and decision logic, also contributes to comprehension difficulties among readers who tend to interpret IDEFo models as representing a sequence of activities.

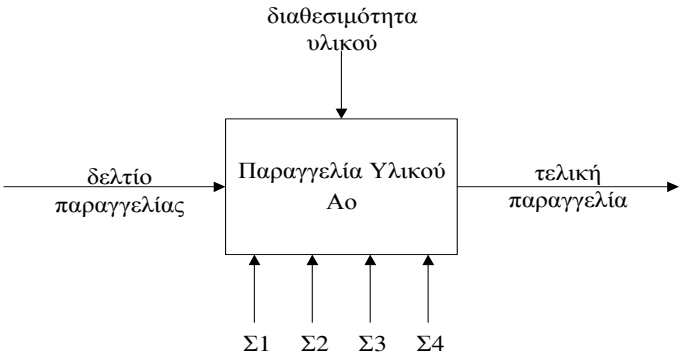
IDEFo allows the modeller to express clearly factors which are required by an activity (inputs, controls, mechanisms). Therefore, it provides the information necessary to generate activity-based costing scenarios. By analysing the "as-is" situation of the process and focusing on high-cost activities, the methodology identifies prime candidates for reengineering.

However, IDEFo models are imprecise about details at the primitive process element stage and vague about details of concurrency, resource conflict, timing, and state-oriented behavior. As a result they should be extended to allow execution or enactment. Most models of this category cannot be simulated or enacted due to the necessity to add more information than the notation supports.

Moreover, since the model describes what an organisation does, it is good in identifying the core activities and secondary functions of the organisation. It provides the modeler and process participants with the means to understand

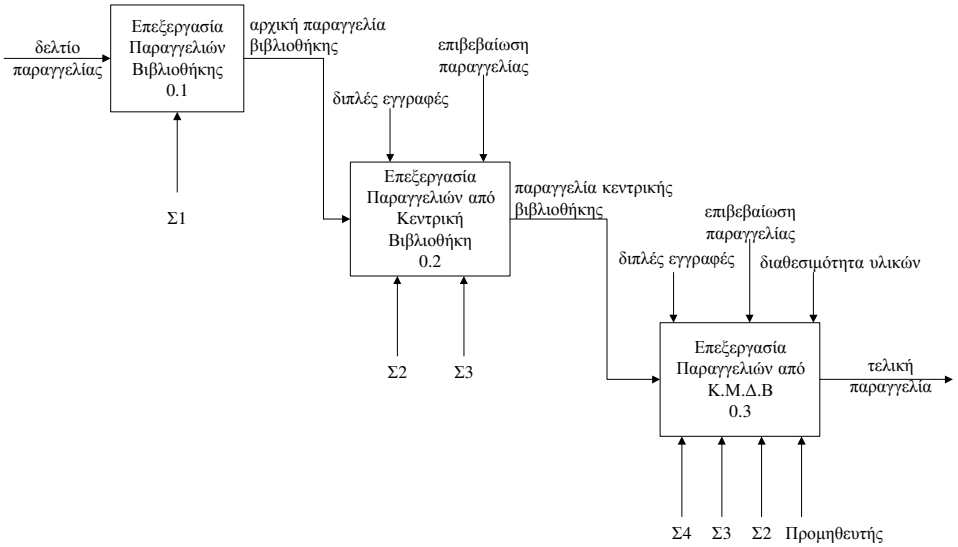
the structure of activities within a process, it is a useful device in radical change BPR efforts, where the concern is to understand a process as a whole and in particular what is done rather who does it and how.

Having the above in mind we can conclude that the IDEF0 model could facilitate the communication between the modellers and the process participant's as well as their understanding of the process by being used in high level modelling of the current situation. It could also be used to create high level models of the "to-be" process.



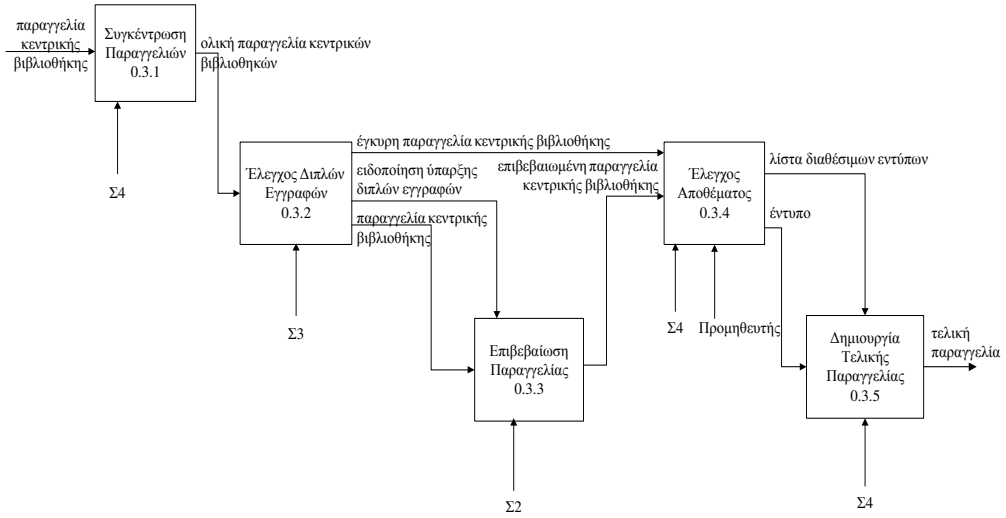

Σ1: Στέλεχος του Γραφείου Διαχείρισης Υλικού και Εξοπλισμού της Βιβλιοθήκης
Σ2: Υπεύθυνος Γραφείου Διαχείρισης Υλικού και Εξοπλισμού της Βιβλιοθήκης
Σ3: Προϊστάμενος της Κεντρικής Βιβλιοθήκης
Σ4: Γραφείο Διαχείρισης της Κ.Μ.Δ.Β

***Figure 2***. The Context Diagram of the *Order of Material* process




Σ1: Στέλεχος του Γραφείου Διαχείρισης Υλικού και Εξοπλισμού της Βιβλιοθήκης
Σ2: Υπεύθυνος Γραφείου Διαχείρισης Υλικού και Εξοπλισμού της Βιβλιοθήκης
Σ3: Προϊστάμενος της Κεντρικής Βιβλιοθήκης
Σ4: Γραφείο Διαχείρισης της Κ.Μ.Δ.Β

***Figure 3***. First level model of the *Order of Material* process




Σ1: Στέλεχος του Γραφείου Διαχείρισης Υλικού και Εξοπλισμού της Βιβλιοθήκης
Σ2: Υπεύθυνος Γραφείου Διαχείρισης Υλικού και Εξοπλισμού της Βιβλιοθήκης
Σ3: Προϊστάμενος της Κεντρικής Βιβλιοθήκης
Σ4: Γραφείο Διαχείρισης της Κ.Μ.Δ.Β

***Figure 4***. Second level model of the *Order of Material* process

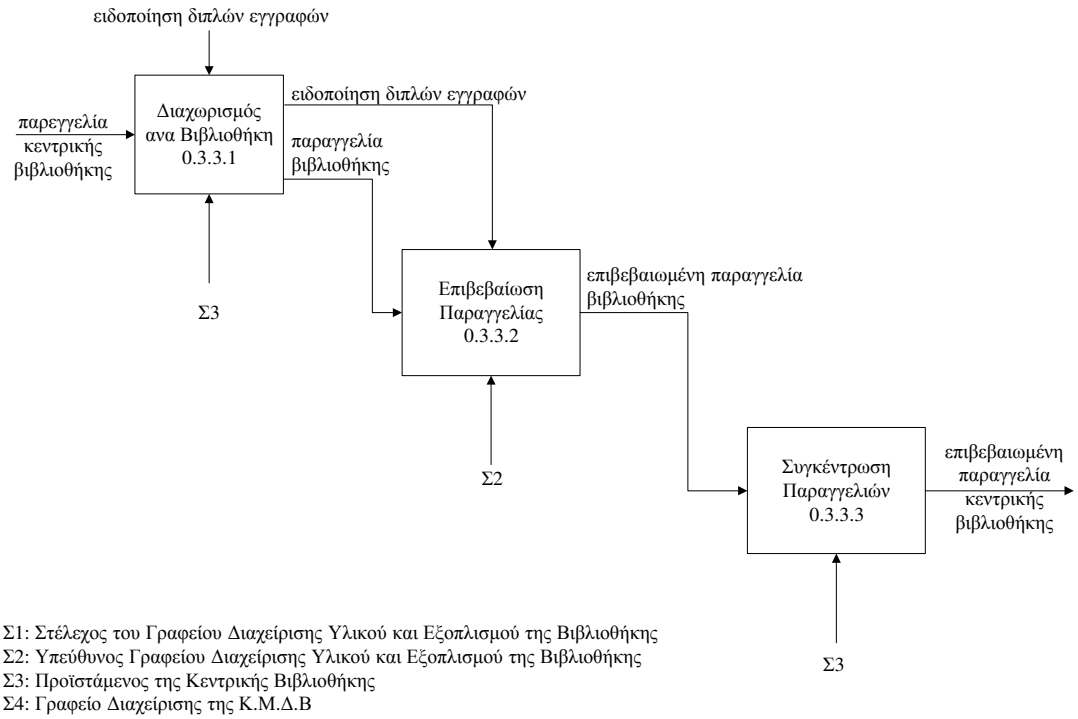


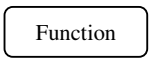

***Figure 5.*** Forth Level of the *Order of Material* process

## 1.2. EPC Model (Event-Driven Process Chain Model)

The Event-Driven Process Chain Model is a widely known and used notation for the description of business processes. In fact, it is an advanced form of flow chart diagrams.

The architecture of integrated information systems [Scheer ή SAP] splits the total view of an organisation into four different views: data view, function view, organisation view, and process view. The process view describes the dynamics of an organisation using information from the static views and as a result it integrates parts of the other three views. EPC was developed in order to reflect the organisational, functional, data oriented and dynamic aspects in the process view. It is used by SAP R3 Reference Model to describe all business processes supported by R3 System of SAP. The most important graphical entities of EPC are:


: a (business) event which reflects a signal in a business environment which triggers the execution of a function.




: a (business) activity that is executed either by a person or automatically. After the execution of a function an event is produced.

: conjunctive, disjunctive, and adjunctive split and join. Splits and joins are used to describe the logic of control flow.

An AND-split executes parallel paths that are synchronized by

an AND-join.

An XOR-split defines that exactly one of the following activity should start. So there is no need for XOR-join.

An OR-split indicates that all combinations of the following paths are possible. An OR-join is sometimes needed to synchronize the incoming paths.

: control flow

process

: process paths. Process paths are connections to other EPCs to divide business processes into various EPCs.

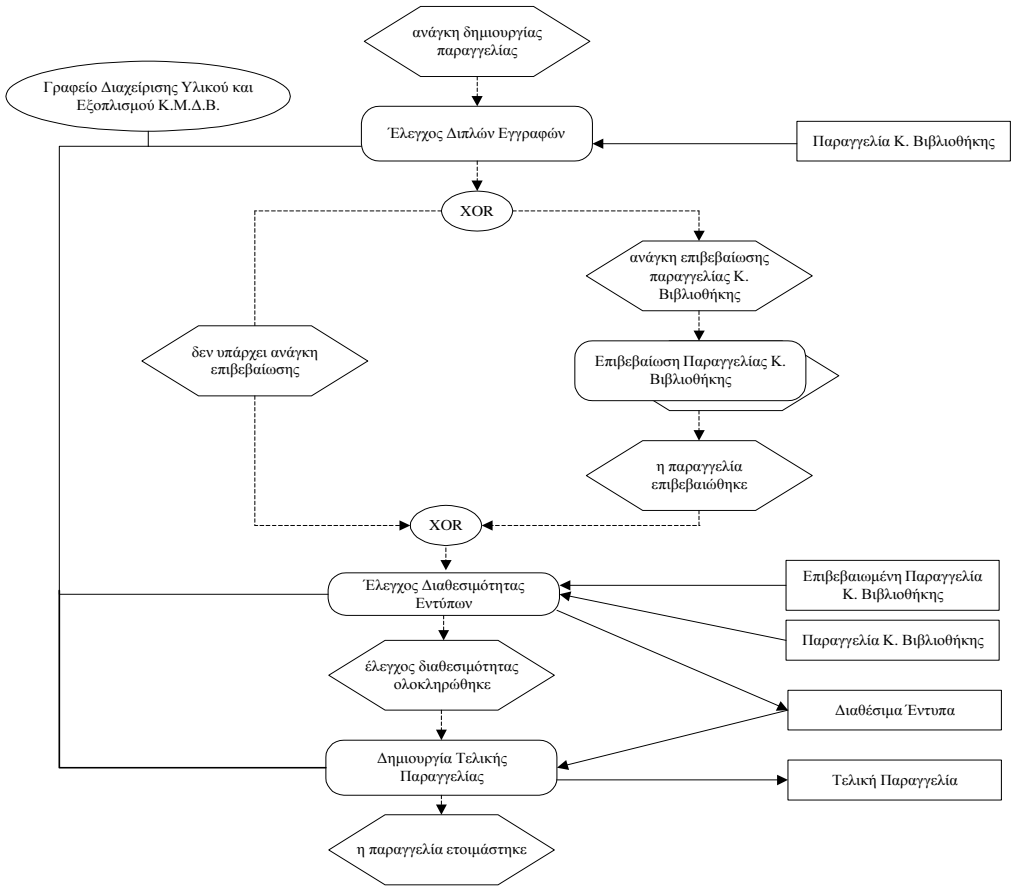
org. unit

: organisational unit. Functions can be connected to an organisational unit to define responsibility for the execution of the activity.

information object

: information object. Information objects describe what information is needed for the execution of an activity (function inputs) and what information is produced by an activity (function output).

Despite its popularity, EPC has some drawbacks mainly because the initial goal of its use was to create a simple documentation of business processes. Documenting a business process does not require either a formal syntax or formal semantics. Therefore, EPC lacks formal syntax and well defined semantics, which are essential elements for model simulation and enactment. Furthermore, EPC does not offer a possibility to specify in more detail the data



flow, the conditions of the control flow and the functions - activities of a business process.

***Figure 6.*** The EPC model

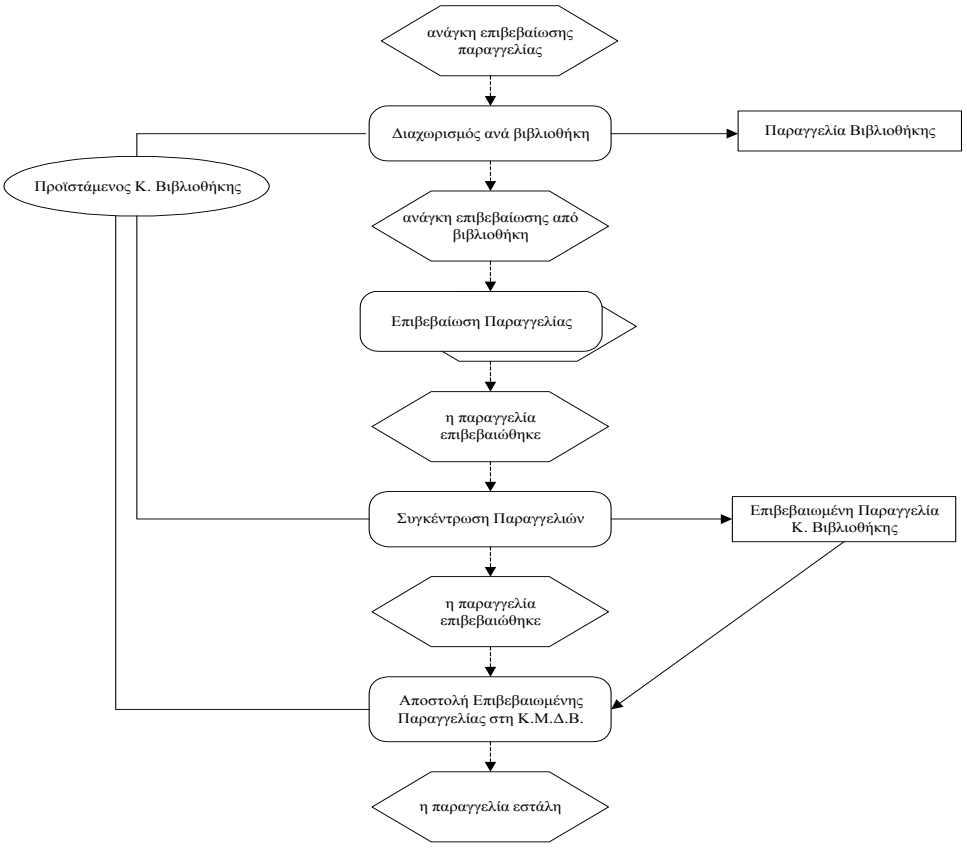



***Figure 7.*** The EPC model

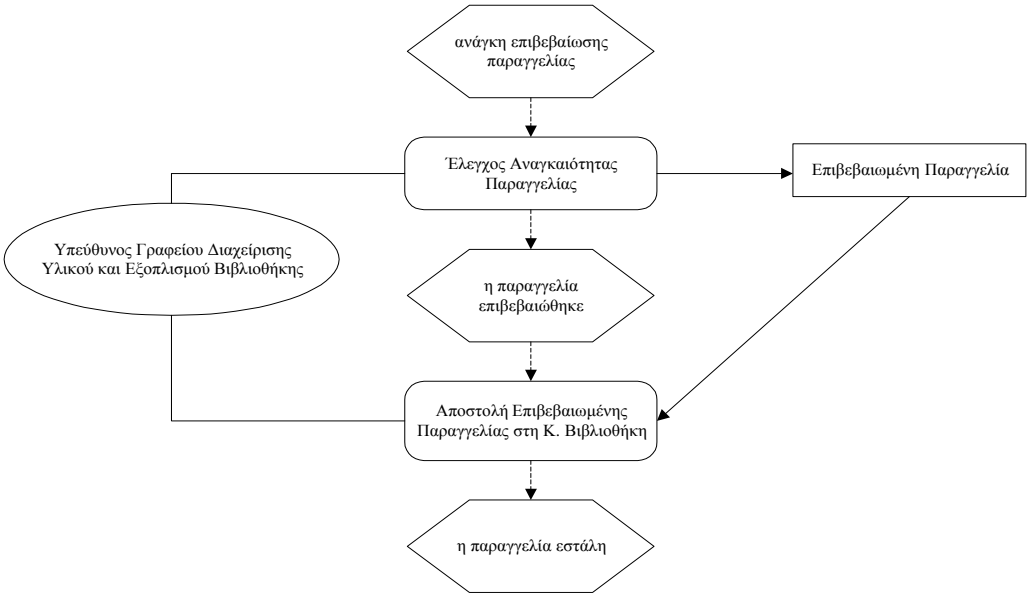



***Figure 8.*** The EPC model

## *2. Petri Nets*

Petri nets is a well-known formalism designed to model systems that comprises interacting concurrent components. They originated from the doctoral dissertation of Carl Adam Petri.

A Petri net is composed of four parts:

1. a set of places P,
2. a set of transitions T,
3. an input function I, and
4. an output function O.

Its places, transitions, input function, and output function define the structure of a Petri net. Transitions and places are related through input and output functions. The input function I is a mapping from the transition $t_j$ to a collection of places $I(t_j)$, known as the input places of the transition. The output function O maps a transition $t_j$ to a collection of places $O(t_j)$, known as the output places of the transition.

A token is another primitive concept of Petri nets. Tokens are assigned to, and can be thought to reside in the places of a Petri net. They are used to define the execution of a Petri net. The number and position of tokens may change during the execution of a Petri net. An assignment of tokens to the places of a Petri net is called a marking $\mu$.

The graphical representation of a Petri net is a Petri net graph, which has two types of nodes:

- a circle ○ that represents a place,
- a bar | that represents a transition,

Tokens are represented by small dots • in the places of the Petri net.

Directed arcs connect places and transitions. An arc directed from a place $p_j$ to a transition $t_j$ defines the place to be an input to the transition. An arc directed from a transition $t_j$ to a place $p_j$ indicates an output place for this transition.

The number and distribution of tokens in the Petri net control the execution of it. A Petri net executes by firing transitions. A transition fires by removing tokens from its input places and creating new tokens that are distributed to its output places. In order a transition to fire, it should be enabled. A transition is enabled if each of its input places has at least as many tokens in it as arcs from the place to the transition. Firing a transition will in general change the marking $\mu$ of the Petri net to a new marking, $\mu'$.

Petri nets were soon recognised as one of the most adequate model for the description and analysis of synchronisation, communication and resource sharing between concurrent processes. But, this primitive form of Petri nets had two important drawbacks: First, they often became very large because there were no data concepts and all data manipulation had to be represented directly into the Petri net. Secondly, they do not allow modelling in various level of abstraction. Many modifications of the original Petri nets have been developed trying to overcome these shortcomings, such as: Coloured Petri Nets. Here we describe Multilevel Petri Nets, a form of Petri nets specially developed for business process modelling.

### *Multilevel Petri Nets*

Multi - Level Petri Nets (MPNs) is a modified form of Petri Nets introduced by Tsalgatidou that facilitates business process modelling. The main goal was the development of a model able to provide the essential symbols to model organisation's architecture and dynamics, and to derive executable models that can be validated either through static analysis techniques or through simulation.

MPNs provide constructs for accommodating all kinds of business process related information. Business process' activities are modelled as MPN transitions, while objects required for the execution of an activity are modelled as objects on MPNs.

These objects are:

- Control information - signals representing messages among activities or events representing occurrences of incidents,
- Resources - data objects used by the process, and
- Actors - a set of duties and responsibilities in the organisation or an external participant. Actors are required in the input places of a given transition where specific human participants must be present for its enactment.

Figure 9 shows the basic modelling symbols of MPNs.

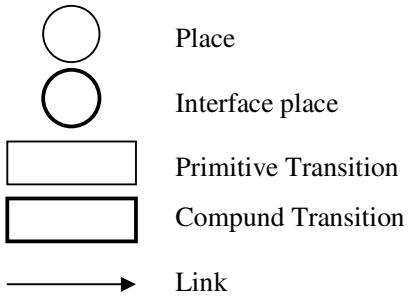



***Figure 9***. The basic modelling symbols of MPNs.

Multi-level Petri Nets inherit the formal syntax and semantics of the pure petri nets. A formal definition of MPN is next provided. A modified Petri Net MPN is defined as a tuple:

$$\text{MPN} = < P, T, F, N, S, script, struct, SubMPNs, M_o >$$

where:

- *P* is a set of places,
- *T* is a set of transitions,
- *F* is a set of arcs connecting places to transitions and transitions to places,
- *N* is the multiplicity of the arc,
- *S* is the underlying structure of an MPN,

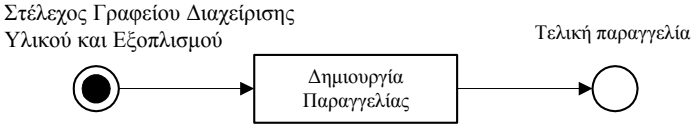
- *Script* is a mapping from transitions to scripts where a script is the set of steps to be carried out during the execution of a given activity,
- *Struct* is a function that maps places to the underlying structure of the MPN, and specifically to object types and actor classes,
- *SubMPNs* is a set of MPNs which may be empty. For each compound transition in MPN there is a member in SubMPNs.
- $M_o$ is the initial marking of the Petri Net.

An important feature of MPNs is that they allow modelling business processes in various levels of abstraction each one corresponding to the activities of the organisational units of one or more levels of the organisational hierarchy involved in the modelled process. In a MPN there are compound transitions, which are decomposable activities, and primitive transitions that do not require decomposition. The highest abstraction level of a MPN (called the top level MPN) contains just one transition that represents the behaviour of the organisational unit that is at the highest level of the part of the organisational hierarchy involved in the modelled process. The top-level transition is decomposed into a number of other transitions and places. These resulting transitions may be further decomposed into a number of transitions and places of lower level MPNs. Sub-activities of a given activity are performed either by organisational units of the same level as those performing the parent activity or by organisational units of lower levels, if they are carried out by more specific organisational entities.

MPNs are executable and can be simulated and validated in different abstraction levels. Another important feature of MPNs is the Timestamp property owned by all objects (control and resource objects). Timestamp property contains object's creation time and enables temporal modelling using MPNs. A transition may introduce delays by appropriate handling the Timestamp of its output tokens, allowing in this way several kinds of statistical analysis of temporal aspects of MPNs. Both static and dynamic validation is facilitated by the formal and executable nature of the model combined with the formal decomposition semantics.

Although the formality of the model enables its static and dynamic analysis as well as it execution, it can be considered as a disadvantage of the approach, because it makes it difficult for modellers to use it easily and for process participants to understand it.

Subsequently, we can conclude that MPNs are not suitable to be used in the early stages of a BPR effort during which comprehensibility and ease of use are of greater importance. On the contrary they can efficiently support the modelling of a business process in a lower level, where details are needed in order to describe how a process is performed, as well as the dynamic analysis of the model in order either to identify possible drawback in the current process or to evaluate the performance of alternative re-design scenarios.



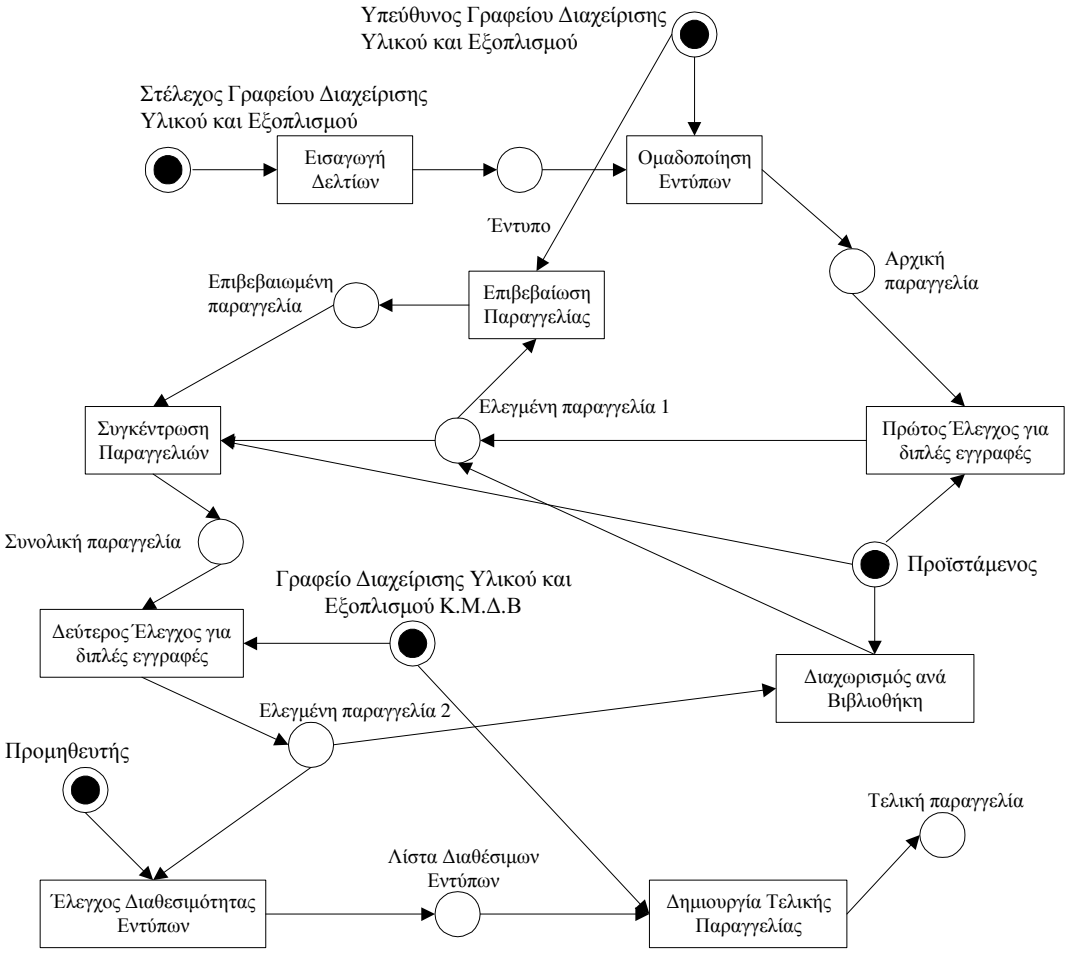



***Figure 10***. The first level of the MPN model of the *Order of Material* process

## *3. Agent oriented (or Role oriented) Models*

Agent-oriented (or role oriented) approaches specify and analyse the role of the agents that participate in the process. A representative example of the agent oriented approaches is the Role Activity Diagrams.

### 3.1 Role-Activity Diagrams

Role-based models group process activities into 'roles', which describe the behavior of an individual, a group of people, or a system. According to Ould "a role involves a set of activities which, taken together, carry out a particular responsibility or a set of responsibilities". Role Activity Diagrams (RADs) originated from the study of coordination by Anatol Holt. Although they are easily readable and simple, RADs have very powerful modeling features. In the following section we describe the most important RAD concepts: roles, roles' state, activities, control and iteration.

***Roles***

It comprises a set of **activities**, which are carried out by a **group**, an **individual** or a **system**. So a role reflects **some unit of responsibility** or an **effort** to accomplish a goal. A role is **independent** of other roles, but communicates with others through interactions. An individual may have the responsibility of multiple roles.

A RAD role should be considered like a **class** in object-oriented design: it describes behavior, but when the process is enacted there are many instances of it. Therefore, there may be many instances of a particular role (e.g. customers are instances of customer role).

In a RAD, roles are represented as rounded rectangles surrounding activities. These rectangles are typically white but sometimes are shaded. A **vertical line** in a role represents a **thread of control**. Control threads indicate *sequential or parallel activities*, or the possibility to *choose* between alternatives.

*State*

A vertical line in a RAD role indicates a state of this role. A role moves from state to state by carrying out activities. Typically role's states are not labeled. The vertical line above the activity represents role's previous state and the line below its new one.

*Activities*

There are two kinds of activities in a RAD: **actions** and **interactions**. An action is an activity that the role carries out without interacting with others, and is represented by a small square inside the role. After an action, a role moves from its present state to its next state.

An interaction indicates a role's activity that is carried out in sequence with another activity or some other activities in another role (or roles). After an interaction, all the involving roles move to their next state. An interaction

involves two or more roles, but is always **driven by one of them**. Interactions are represented by small squares joined with horizontal lines.

*Control*

RADs allow the representation of both **alternate** and **concurrent** activity paths. Alternate paths indicate choice, while concurrent path indicates parallel execution.

Choice is also called **case-refinement**. The control thread being split into two parts represents a case refinement. The top of each part is marked with a downward pointing triangle or a circle. Only one of the alternative paths may be chosen.

**Parallel execution** is also called **part-refinement** and parallel threads (parts of the path) represent it. The points where the path is divided are marked with an **upward pointing triangle**. All threads joined together again after the split denote completion of all parts.

*Iteration*

Iteration is used when there is some control mechanism within the role. It is represented by drawing a loop back to a previous point on the role indicating that this particular state may be revisited.

RADs can be efficiently used to describe the execution of an existing process. Part refinement is a very powerful feature of RADs, because it allows concurrent execution of particular activities to be represented. Usage of part refinement leads to fatter roles that capture the flexibility of work more effectively. Moreover, some case studies have shown that this RADs' feature is very helpful in avoiding development of very prescriptive and constraining process support.

It is obvious that RADs can be used to describe organisational roles (job titles). RADs built this way can be very helpful, since interactions are closely related to the real process performance. But if RAD roles are viewed as "a set of activities which, taken together, achieve some particular goal", then we can detach them from the organisational structures of the particular company. Then RAD can represent" cohesive and decoupled modules of work", showing what is done and not who does it. When RADs are built under this perspective, an interaction does not show people or agents communicating with each other, but is a point of synchronisation between two or more roles.

RADs have proven to be a very powerful modelling technique, because they are simple, describe in sufficient detail how work is done, can express work modularity and can contribute to process redesigning efforts at execution and support level. Case studies have shown that RADs convey less information about what triggers the execution of a particular activity than an IDEF0 model does. RADs are not proper for the representation of the overall flow and structure of an end to end process. Moreover, constructing a RAD is a time consuming process.

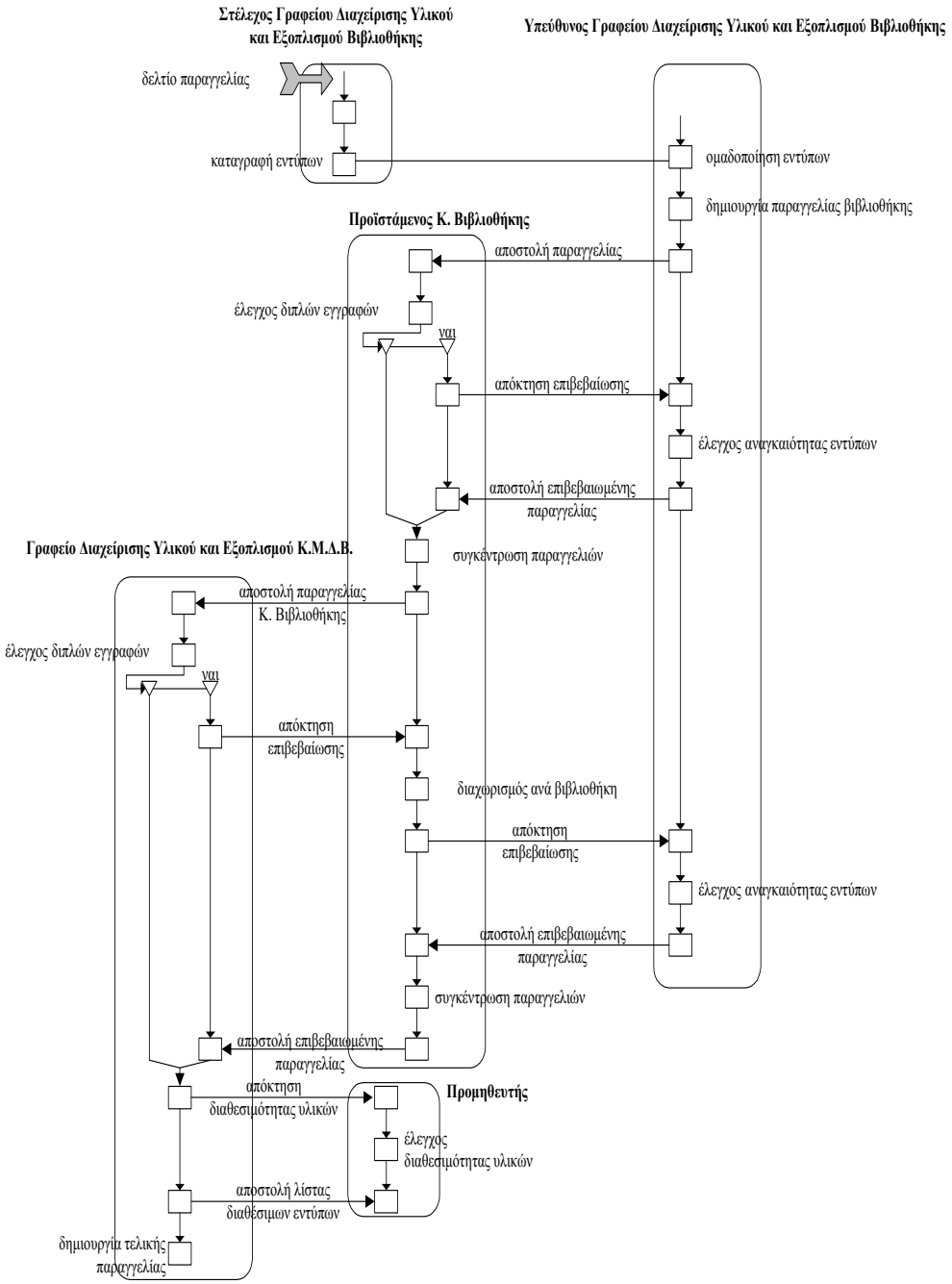



***Figure 12***. The RAD of the *Order of Material* process

## *4. Goal-oriented models*

Human participation is one of the most important features of business processes. Human action is goal-driven, meaning that human act in order to achieve their goals and fulfil their expectations. So each business process exists in order to accomplish some goals. Modelling business processes and business process reengineering aim at a more efficient and effective organisation. According to some reearchers the explicit use of goals during the design of new processes and its evaluation can significantly contribute to the desired improvement.

Inability to understand and evaluate business goals and especially the goals of a business process may conclude to a disoriented modeling effort. This is the reason why many researchers suggest that a modeling effort should evolve the representation of goals, the understanding of the way that these goals shape processes, and a goal-based evaluation of new process' designs. In this case, the modeler should:

1. capture the different kind of goals from the business process participants,
2. assess the compatibility of these goals,
3. manage inconsistencies, and
4. create business processes that contribute to the fulfillment of these goals.

Here we give some examples of goal-oriented process models. We describe i* framework and Action Workflow model.

### 4.1 I* framework

*[Yu, Mylopoulos, From E-R to "A-R" – Modelling Strategic Actor Relationships for Business Process Reengineering, in Entity-Relationship Approach (ER'94) – Business Modelling and Re-Engineering, (Proc. 13th Int. Conf. on the Entity-Relationship Approach, Manchester, U.K., December 1994) Springer-Verlag, LNCS-889, pp. 548-565]*

In order to understand a business process, despite knowing which are its activities and how they relate to each other, it is also necessary to see **why** these activities and their inter-relationships exist.

Since business processes exist in a social environment, humans participate in their executions. Humans have goals and interests and act respectively in order to achieve and realise these goals.

Yu and Mylopoulos note that a model for business processes should describe not only the activities and the relations between them but also how the actors performing these activities relate to each other intentionally, that is in terms of concepts such as **goal, belief, ability,** and **commitment**. Moreover when an organisation tries to reorganise the way work is performed, actors are likely to evaluate these proposals strategically, e.g. in terms of potential opportunities and threats. So actors relate to each other by intentional and strategic relationships. Yu and Mylopoulos present the i* framework for

modelling intentional, strategic relationships. It consists of the Strategic Dependency (SD) model and the Strategic Rationale (SR) model.

The Strategic Dependency (SD) model describes an organisation in terms of the dependencies that actors have on each other in accomplishing their work. A SD model is a graph, where nodes represent actors, and a link between two actors indicates a dependency between the two of them. An actor depends on another for an entity in order that the former may attain some goal. The depending actor is called the depender, while the actor who is depended upon is called the dependee. The object around the dependency is called the dependum. The existence of a dependency between two actors makes the depender able to achieve a goal, but also vulnerable to the dependee failure to deliver the dependum.

The model uses four types of relationships, based on the type of dependum.

- **Goal dependency:** in a goal dependency the depender depends on the dependee to bring about a condition in the world. The dependee is free to find a way to achieve the goal since the depender is interested in the outcome. Through a goal dependency, the depender can achieve a goal even if it does not have the knowledge or the resources to do so, but becomes vulnerable since the dependee may fail to bring about the condition or the state of the world.
- **Task dependency:** in a task dependency the dependee has to carry out an activity for the depender. The dependee performs the task in a way that is defined by the activity specifications. Through a task dependency, the depender is able to have performed an activity, but becomes vulnerable to the dependee since it may fail to carry it out.
- **Resource dependency:** in a resource dependency the depender depends on another for the availability of an entity. Entities are objects in the world and they can be physical or informational. Through a resource dependency, the depender can use the entity, but becomes vulnerable if the entity is not available.
- **Softgoal dependency:** a softgoal dependency is a *hybrid of goal and task dependency*. A depender depends on the dependee to perform some task that meets a softgoal. A softgoal is typically a quality or non-functional attribute on one of the other intentional elements.

A dependency can be open, committed, or critical. An open dependency means that is the dependency fails, the depender would be affected but not significantly. A committed dependency means that the depender would be significantly affected if the dependency fails. A critical dependency indicates that if it fails some goal of the depender could not be achieved.

SD model uses **three different kinds of actors: roles, positions,** and **agents**. A role is an abstract actor. Agents are concrete physical actors such as humans or software agents. Agents play roles. An agent typically plays a collection of roles, which make up a position.

The SD model is used in order to describe a particular design for a business process.

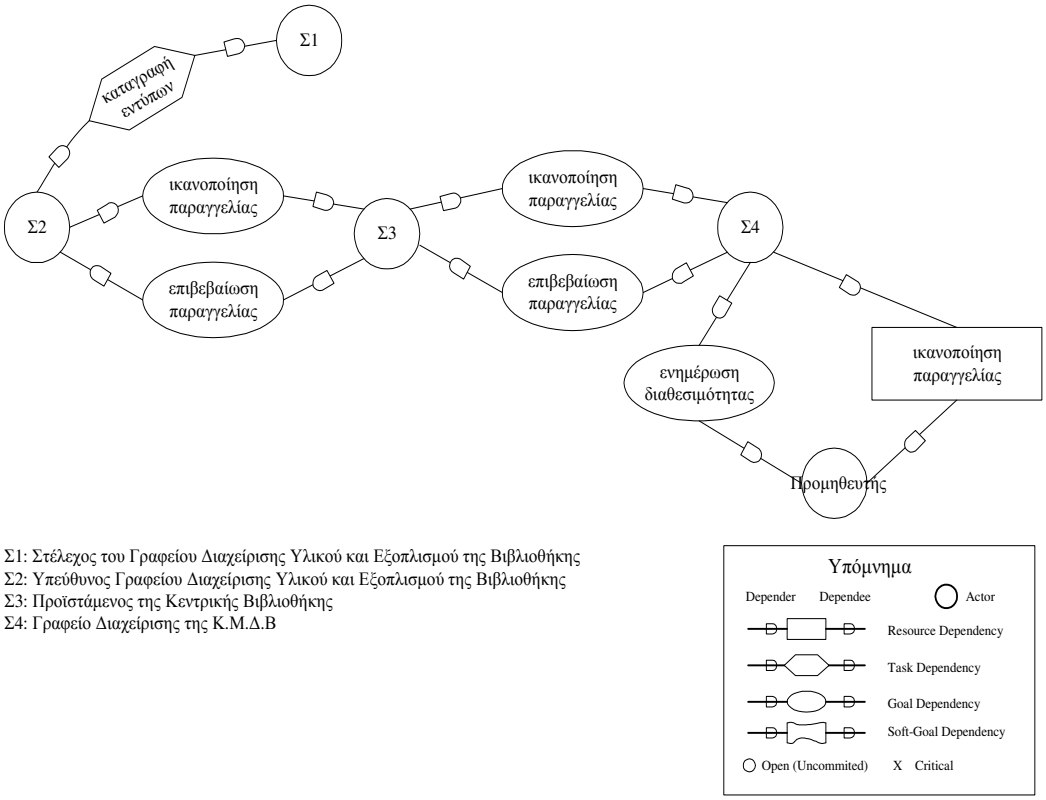



***Figure 13.*** The SD model of the *Order of Material* process

The Strategic Rational (SR) model describes what actors think about the different possible ways of organising work, which are represented by different configurations of SD networks. In fact a SR model shows how incoming dependencies (for which the actor is dependee) are related to outgoing dependencies (actor is depender). The way an actor meets its incoming dependencies or internal goals and desires is called task. A task is broken into its components, which are broken into sub-components, and so forth. Each component is an intentional element that can be a goal, a task, a resource, or a softgoal. There are more than one way to achieve a goal, to perform a task, to produce a resource, or to satisfy a softgoal. So between an element (the end) and each way (the means) of decomposing it into sub-elements there is an intervening means-end link.

A SR model is a graph. There are **four types of nodes: goal, task, resource, and softgoal**, and **two kinds of links: means-ends links and task decomposition links**. A task decomposition link can be a subgoal, subtask, resource, or softgoal link. A task decomposition link can be open or committed.

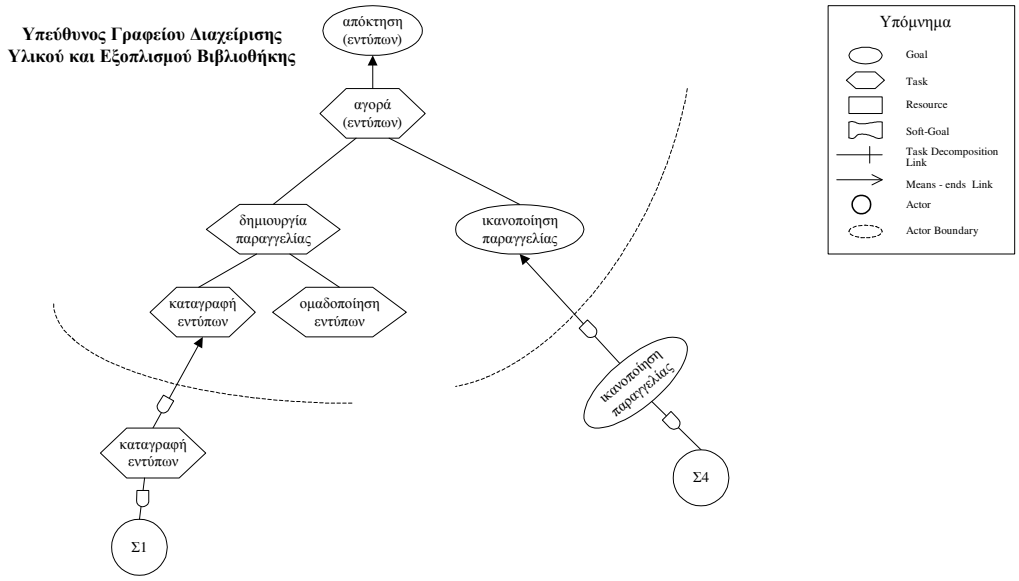



***Figure 14.*** The SR model of the *Order of Material* process

Although the I* framework was developed to express and visualise the inter-relationships and dependencies between the actors of an activity, it is not easily understood. Moreover, it does not represent the sequence of activities within a process and therefore in a BPR project its usage is a complementary one.

## 4.2 Action Workflow Model

Typical examples of goal-oriented models are the Communication-based models. These models assume that the goal of a business process is to improve customer satisfaction.

This methodology assumes that the objective of business process reengineering is to improve customer satisfaction. The action model of Action Technologies belongs to this category. It is the outcome of an effort aiming at the combination of co-ordination theory and workflow systems and comes from the Winograd/Flores Conversation for Action Model.

The action model considers a process as a sequence of interactions between people who have a common goal, that is to satisfy a specific customer's proposal. In fact a process is viewed as an effort to co-ordinate human actions. When this co-ordination is successful customers are satisfied. It reduces every action in a process to four phases on communication between a customer and a performer:

1. *Proposal*

The customer requests (or the performer offers) the completion of a particular action according to some conditions for satisfaction.

*2. Agreement*

The two parties mutually agree on the action to be performed and the conditions of satisfaction.

*3. Performance*

The performer performs the requested action and declares the completion to the customer.

*4. Satisfaction*

The customer reports to the performer either satisfaction or dissatisfaction.

Figure 15 illustrates these four phases.

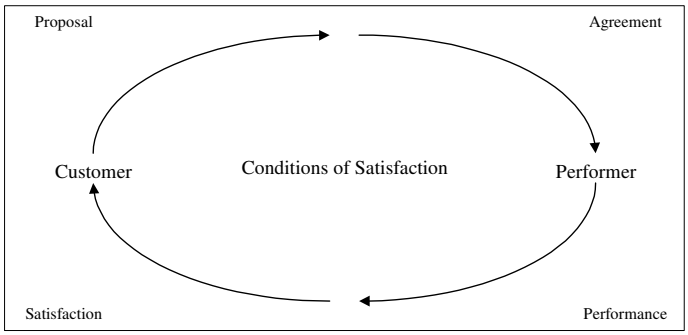



***Figure 15***. The four phases of the Action Workflow Loop.

At the heart of the Action Workflow model is a loop, which represents a unit of work within the process. It involves two participants: a customer and a performer. Linking several workflow loops specifies the flow of the process. As other workflows are specified and linked, graphical map of the process emerges as a network of workflows.

There may be some more actions, such as clarifications, further negotiations on the conditions of satisfaction, or changes in participants' commitments. The structure is based on language acts through that people co-ordinate their activities, but not on the actions they perform in order to fulfil the conditions of satisfaction. We can notice a shift from a task structure to co-ordination structure. In Action workflow requests and the commitments that are expressed in the workflow loop define tasks.

According to this approach a business process is designed or redesigned as a collection of related workflow loops. Each workflow loop has each one probability to be completed successfully. One loop's performer may be the customer of another loop and vice versa. Relations between workflow loops in a process map are used to trigger the right sets of following actions. These relations are of various kind:

- *Subordinate workflow loops*

In order a part of a workflow to be completed, it is necessary a subordinate one to be initiated and completed.

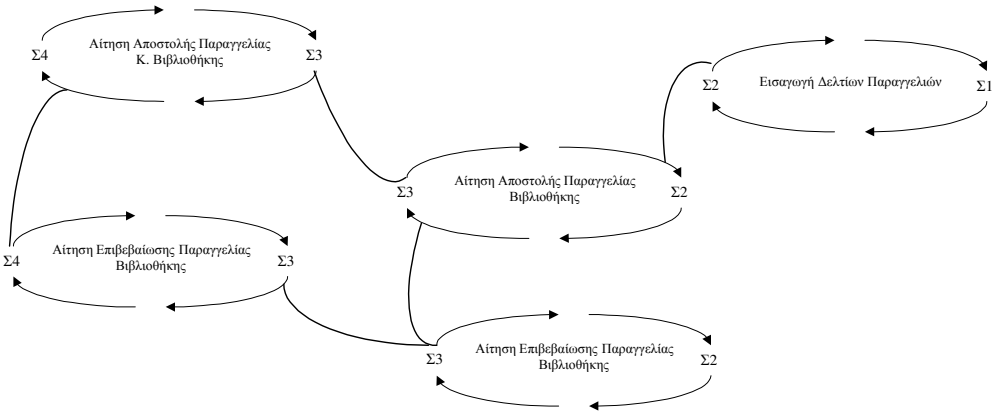
- *Independent triggered workflow loops*

  A loop's action triggers the initiation of another, which is independent.

- *Resolving workflow loops*

  The decision on which action to be performed in a workflow loop requires the initiation of another.

Workflow loops can proceed concurrently.




Σ1: Στέλεχος του Γραφείου Διαχείρισης Υλικού και Εξοπλισμού της Βιβλιοθήκης
Σ2: Υπεύθυνος Γραφείου Διαχείρισης Υλικού και Εξοπλισμού της Βιβλιοθήκης
Σ3: Προϊστάμενος της Κεντρικής Βιβλιοθήκης
Σ4: Γραφείο Διαχείρισης της Κ.Μ.Δ.Β

***Figure 16.*** The Action Workflow Model of the *Order of Material* process

The Action Workflow model efficiently represents administrative processes. Although, it is based on the communication between the customer and performer of an activity, it does not explicitly express the values on which the fulfilment of an agreement depends. Moreover, in practice the conditions under which an agreement will be fulfilled cannot be defined from the beginning.

## *5. Conclusions*

In the previous sections seven representative modelling notations were briefly presented. In the following table the main features of them are summarised.

| **Feature** | **IDEF0** | **IDEF3** | **EPC** | **RAD** | **MPN** | **I* framework** | **Action workflow** |
|---|---|---|---|---|---|---|---|
| *ease of use* | High | Moderate | Moderate | Low | Low | Low | High |
| *Comprehensibility* | Very good in high level | Quite good | Expresses what triggers an activity | Easily understood<br>But unable to give an overview | Difficult | difficult | Quite good |
| *Originality* | no | yes | For documented processes | yes | no | yes | Yes |
| *formal syntax and semantics* | no | yes | no | No | yes | no | no |
| *expressiveness* | Only in forms of ICOMs | Selection, iteration | yes | Yes | Implicitly yes | no | Not explicitly |
| *hierarchical decomposition* | yes | ? | no | no | yes | no | In a way |

What model is going to be used depends on two things:

- on the kind of the process under study,

- on the phase of the BPR effort.

In a BPR effort a combination of models should be used.

Models of goals form the basis for defining the objectives of the study as well as the monitoring and analysis of the current and to be – processes.

Activity oriented models can be used for high level modelling while for more low level models RADs or IDEF3 models could be used. The first facilitates understanding and support the selection of processes for reengineering the second facilitates the analysis (simulation) and the redesign of the process.