# Empowerment the IDEF0 Modeling Language

Loukas Tsironis (Corresponding author)
Technical University of Crete
Department of Production & Management Engineering
73100, Chania, Greece
Tel: +30-2821-037-361 E-mail: loukas@dpem.tuc.gr

Andreas Gentsos & Vassilis Moustakis
Technical University of Crete
Department of Production & Management Engineering
73100, Chania, Greece

## Abstract

In the present article we discuss, evaluate and improve the IDEF0 modeling language. In order to meet end user requirements, we suggest concrete improvements which empower the language to face real world problems such as: human errors, process delays, parallel processes and detail information description. The experimentation field comprised from a common production system. Improved IDEF0 seem to overcome several deficiencies and increase its modeling performance. Results showed that language deficiencies were clarified. Thus, the improvements made, constitute better modeling performance and the development of more reliable models.

**Keywords:** IDEF0, Human error, Process delays, Parallel processes

## 1. Introduction

The term Business Process (BP) is widely used among Business individuals and researchers. It includes an activity or set of activities that supplied with one or more kinds of inputs and transforms them into an output which adds value to the potential customer (Hammer and Champy, 1993). Each activity composites of components such as tasks, information, materials, operators and machines. Examples of a BP can be a production line, the functions within organizational departments, a decision making process and so on.

The study of BPs based on their analysis through the representation and description of their component information, which flows among activities. The study and analysis of BPs can be transacted through the consideration of certain Business Models (BM). BP organizational modeling implies the utilization of specific Business Process Modeling Languages (BPMLs), corresponding software tools for the graphical description of observed behavior of processes in a business system. The result of the modeling process is a well structured and well mapped business model, which represents, in depth, the actual process components and its course. Models enable end users to face the complex and dynamic nature of modern BPs. Models are very useful in the entire BP life cycle course supporting its identification, description, redesign and continual improvement.

A very popular BPML is the IDEF0 (Ross, 1985; Feldmann, 1998; Kalnis et al., 2000; Kalnis, 2004; Recker et al., 2006; Liu and Fang, 2006;). IDEF0 applies to all kinds of organizational systems independently of their size or complexity (Godwin et al, 1989).

BPML makes use of objects for the graphic description of information flow. These objects are called components. The type of information that contains a process component (e.g. document, file) is called component information and it is steered by BPML semantics. Semantics of a BPML refers to the logical identification, of a syntax component and the appropriate linkages among them. Thus, a model that imitates the real world system, according to the principles and methodology of BPML, is created.

A BP is decomposed in functions, where every function captures components and semantics. It is a set of one or more functions, integrated in diagrams. A model composed from a number of diagrams develops in hierarchical fashion. For the modeling task and the application of the BPMLs, we used specific software tools.

The selection of a BPML for the organization constitutes a strategic goal for the establishment of effective communication among people who take part in a BP, in order to assure the quality of its outputs. Thus, the evaluation and comparison of BPMLs constitutes an organizational need.

In the present article we cope with several concrete deficiencies of IDEF0. Furthermore, we recommend specific syntax components and semantics that are targeted to improve the selected BPML in order to cover more organizational needs.

To document our findings we used the modeling case of a real production system.

The purpose of this article is to reveal several deficiencies of IDEF0 during the modeling phase, to improve these deficiencies and increase the modeling performance of the IDEF0. Some of the revealed deficiencies are:

(1) Errors. An error occurs after the influence of several factors resulting in an undesirable outcome. The error can be caused by an individual or a machine. Thus the errors can be categorized to human errors and machine faults.

(2) According to (1) remark, we propose new design capabilities for the detection of errors.

(3) Process delays. Process delays can be described as the several reasons that cause a delay to the process course.

(4) Parallel processes. Processes that are flow together.

(5) The transferred information. It concerns the exact descriptions of data or objects related to functions.

Scientific literature identifies a few attempts to assess IDEF0 and present its shortcomings (Tsironis et al 2007). However, a particular problem in this context is the lack of research regarding what is to be considered good design (Mendling, 2007). For instance several aspects or characteristics of BPML evaluation can be read at (Godwin, et al., 1989; Hernandez-Matias et al., 2006; Jansen-Vullers and Netjes, 2006; Janssen et al., 1997; Kalnis et al., 1998; Killich et al., 1999; Kudrass et al., 1996; Shen et al., 2004; Wang et al., 2006).

However, in all studies the need for a thorough evaluation of BPML tools is emphasized. In all articles it is reported that this need arises from the specific weaknesses in the resulting models, which create several problems within organizations (e.g. miscommunication among people and/or departments). Moreover, this need is empowered by the fact that BP modeling is a continuous and time demanding procedure, which has no pre-defined outcome (Tsalgatidou, 1998).

In some of the previously cited articles, the authors assign the need for improvements in BPMLs in order to describe BPs completely. In others authors, noticed the need for a better synergy between BPMLs and the corresponding software tools.

Janssen et al., (1997) and Wang et al., (2006) compared BMPLs. They argue that there is no BPML able to completely satisfy end user' needs. They noted that there are significant deficiencies in languages and that they can be if syntax components and semantics can be simply exchanged with one another. Similarly Jansen-Vullers and Netjes (2006) work underlines the incompleteness in specific capabilities of business simulation tools.

The static nature of IDEF0 and the absence of an adequate dynamic representation of the process is also reported by Godwin et al., (1989). Shen et al., (2004) compared IDEF0, IDEF3 and DFD (Data Flow Diagram). They present the deficiencies of each language. Particularly for IDEF0, one of the main drawbacks they report is the absence of time representation in a process. The absence of time representation for the description of a BP is supported by Killich et al., (1999). The need for a new innovative and complete BPM tool was reported in Hernandez-Matias et al., (2006).

Apart from reported in the literature findings, the need for better BPMLs created during the modeling steps on two real world's BPs in Tsironis et al, (2007). They detected a communication gap among end users or departments when trying to interpret BP diagrams. Furthermore, they observed that the available syntax components confused modelers and end users. The objectives were not obvious. What seemed to be the exact outputs of the process and of which type (e.g. document, a component which is using to update the database) was not clear. The former observation was also noted in Shen et al., (2004).

All the above articles identify the need for a thorough research on the modeling performance of IDEF0. On the other hand, there is a lack on discussing the modeling of factors such as the variable nature of BPs due to the inclusion of human operators for example.

Another aspect that enforces the need of this research is the cost enclosed during poor designed BP models. Mendling (2007) reports that the costs of errors increase exponentially over the development life cycle it is of paramount importance to discover errors as early as possible.

As an experimentation field we select a traditional production process. Specifically we used IDEF0 to model the machine part production process as it described in SADT, (year). For the modeling phase we used Workflow Modeler software (ref). After the creation of IDEF0 model and diagrams we transformed the process by the insertion of errors, delays and parallel processes.

The remainder of the article is organized as follows. In section that follows we present the basics of IDEF0 and tool used herein. In the following section we describe the application field based on which we applied our proposed improvements. The following section we discuss the proposed syntax components and methodology approaches for the IDEF0 language. Finally we conclude with some remarks on our findings concerning its applicability and the quality of results.

## 2. IDEF0

The IDEF0 language is an updated version of the Structured Analysis and Design Technique (SADT) (Marca and McGowan, 1988) proposed by D. Ross in 1976 for structured analysis of systems. It is accepted in the USA as a federal standard (IDEF0, 1993).

IDEF0 is a BPML used to model decisions, actions, and activities of an organization or system (IDEF0, 1993). The resulting model expresses knowledge about how a system, process, or organization works (Wang et al., 2006).

IDEF0 describes the specific steps of a process course and the relationships developed. It also records the information flows, resulting from these relationships.

Finally, IDEF0 model includes a set of syntax components essential for BP integration. The syntax components include boxes, arrows and diagrams. Boxes represent functions, defined as activities, processes or transformations. Arrows represent data or objects related to functions. The format also provides the basis for model configuration management (IDEFO, 1993).

For the application of IDEF0 on modeling tasks we choose Workflow modeler (Meta Software, 2003) due to its ease of use and its ability to provide the whole set of syntax components of IDEF0 language. Workflow modeler is a standalone software due to its features. It supports modeling capabilities based on IDEF0, IDEF3 and IDEF1X BPMLs. It is also supports exporting the model to Workflow Simulator, a very useful module for simulation purposes of the model.

During the modeling stage we faced several differences that the software has with the basic notion of IDEF0 language. These can be summarized to the following:

(1) The software demands only one input arrow.

(2) Mechanism arrows that originate from the output of a specific process does not allow to the diagram

(3) Tunneled arrows are not recognizable

(4) By default in Workflow Modeler there is an attention point to the synchronization of feedback procedures.

## 3. Case study: model representation

We adapt a model taken from Marca and McGowan (1988). This model describes the manufacturing process of a specific part in a machine workshop. The model consists of six diagrams. Inputs in the model constitute raw materials, mechanical parts and the appropriate information that enclosed in the process (i.e Job completion request, quality standard manual). Process output is the outgoing part. An overview of the manufacturing process is presenting in A0 diagram (figure 1).

In the specific manufacturing process (figure 1) are implemented three basic operations, which can be seen in A0 diagram. The first operation includes information assignments about what specific tasks should implement and how. This operation involves the insertion of plans, activities order and time duration and sequences. In the output is generated the appropriate manufacturing time and specific plans that are sending to the main manufacturing process.

The second operation represents the main manufacturing process of each part. In this operation perform activities and tasks such as: machine preparation, part processing, and tool selection. Process inputs include raw materials, machines and tools, while outputs include finished and unfinished job.

The third operation represents the control of the final products produced in the second operation. The assessment of the work done until this point and the measurement of final product's dimensions are performed in this operation. If the final product does not meet the standards then it will forward back to the appropriate process stage.

These three operations are visualizing in A1, A2 and A3 diagrams respectively and every one of them disaggregated in child diagrams which demonstrate the course of sub-operations.

## 4. Improvements

In the paragraphs that follows we present modeling attempts for the cases of errors and delays. Our attempts lie, basically, in the insertion of new syntax components and semantics to the IDEF0 language. We propose the use of specific logical operators in order to model errors and delays within the BP. The following table 1 summarizes and describes the proposed logical operators. A lengthy analysis and explanation on the use of logical operators can be found in the forthcoming paragraphs.

### *4.1 Errors*

An error can occur when any factor affects the process course. The outcome of error affection is always an undesirable or unexpected result. Errors can be predictable or not. Consequently the time that an error occurs can be deterministic or stochastic.

When the error generated by individual's interference then is called human error. Human error's occurrence time cannot be predicted. On the contrary machine errors is sometimes predictable and depends on its life cycle.

One of the disadvantages that burdens IDEF0, is the lack of modeling notations in any type of errors. This happens because the existence of IDEF0 based only to the description of the process course and not to the identification or prediction of errors. Furthermore, is not obvious to any kind of end user the effects that an error might have. In the present section we are trying to enhance and empower IDEF0 with proposed modeling notations in order to cope with possible errors in the process.

Modeling of errors can be achieved by inserting as modeling notation specific logical operators. Logical operators, used in some BPMLs for connecting several Functions and Events. We, herein, propose the use of logical operators in order to describe and represent several aspects of activities (e.g. human errors) as continuity and try with this way to visualize them. Logical operators used for instance to EPC language. We were modeling the process with the use of “AND” operator and “XOR” operator. The “AND” is used for parallel execution of Functions. The “OR” is used for decisions, when one or more choices are possible. The “XOR” is used for decisions, when only one of several choices is possible. Logical operators can be used as a general modeling notation and not only for the identification of errors.

The use of logical operators in the output of the process enables the modeler to present the possible directions when a possible error can occur. In other words the modeler design new output directions in the face of an error. For instance logical operator “XOR” can be locate in process output and provide three discreet options, desirable or undesirable result and no result.

In figure 2 we present an example that helps to be comprehensive our modeling rationale.

Figure 2 presents the A3 diagram of our case. Logical operators “AND” and “XOR” inducted to model all the possible aspects of the process. In activity A31, for instance, there are two aspects in the output Seal and Assess Job. The “AND” operator can be used in order the two aspects can go along. Assess job task, might have 3 aspects on the output: Good, wrong or no assessment. Logical operator “XOR” can cover those three aspects.

### *4.2 Delays*

Process delays can occur either after the affection of an error or after a mismanagement of an activity. The insertion of the appropriate logical operator can resolve this problem. We propose a logical operator which means that if the corresponding arrow is activated a process delay will occur. In other words the delays operator is an optical signal which points the possibility of an unwanted event.

Activity A235 (figure 3) can be described as follows (Marca and McGowan, 1988): Place bins to easily load raw materials and catch scrap. Take into account material properties and the direction of cuts to determine where scrap will be thrown. The activity has two possible outcomes: correct or erroneous placement of bins. In order to cover those two aspects we place the “XOR” operator at the outcome of the activity. Observe that when an erroneous placement takes place then the process might have a delay of the fulfillment of its objectives.

### *4.3 Parallel processes*

Two arrows generating from two activities might further join in one. That could be an indication that the first two activities might act in parallel. Parallel processes are not visible to an IDEF0 diagram. Parallel processes are those that progress at the same time. They can begin together but not finished at the same time or they could start at different times and ending together or one of them can start before the other and ends after the other and so on. The main characteristic of parallel processes is that in order for a new process to start they must have been finished.

The case of parallel processes handled by the insertion of a specific logical operator which is named as “parallel operator”. This parallel process logical operator characterizes the arrows which join from the leading processes come from parallel processes. The non parallel processes operator identifies that from two arrows the predecessor is the one that arrives first. That is that the event with the smaller finishing time will first pass from the operator and the other one will queuing.

Figure 4(a) sets off the concept of modeling the parallel and non-parallel processes in IDEF0. Activities A233, A234 and A235 are parallel processes and this fact is indicated with the appropriate logical operator.

On the opposite, figure 4(b) visualizes a confusion point in IDEF0. Activities A241 and A242 although they seem to act in parallel they are not and that is why the non-parallel operator placed in the appropriate point on the diagram.

## 5. Proposed design capabilities

In the present paragraph we describe a proposed way of the development of an internal control procedure which enables the BPML to conduct control for errors.

As it is mentioned before errors is very possible to appear in a production process, sometimes depending on known reasons and sometimes on unknown. In the current stage we propose the development of an internal error control procedure in every diagram of the IDEF0 model. This procedure based on a three step notion: Control, Execution and Control-Decision.

We can design a BPML that can develop diagrams with embodied abilities of sending information on the appropriate activities that should be accomplished in order to have a standard based process.

The first step controls the status of the process and reports about the necessary actions should be done for the fulfillment of the process. Afterwards it sends information to the diagram boxes through control arrows on the way that should be acted. Usually control step comprises of one box.

Second step concerns the development of the child diagram based on the standards that have been instituted. This step based on more than one boxes. Box output constitutes the input for the following one.

Third and final step controls and verifies that the pre-set standards were met. If control action identifies an error then a feedback procedure supplies the first step's control action. The main difference between first and last control action is that in the first check what and how things should perform, while in the last check if standards had been met and deciding what else missing that should have been done.

As it is known any IDEF0 diagram comprises from three until six boxes. If the completion of the three steps requires more than six boxes then new child diagrams should form. However, the new child diagram should underlie the notions of the IDEF0 language.

The proposed rational, described earlier, can be seen in figure 5. For instance from A1 box first step, Control, is conducted. From A1 box the green arrows provide with the appropriate description of what exactly should be done at the later stages of the process. The A2 box represents the second step, Execution, of the overall process. A2 box receive information on the way it should acts in order to fulfill process requirements and standards. The third step, Control-Decision, consists of the boxes A3 and A4. In the final stage the produced product is controlled in A3 box, which decided whether it should rejects and returns to A2 box or continue to A4 box in order to comes out of the system. This is the reason why A2 box sends the blueprints in order to help the process on the control and decisions should make.

### *5.1 Number of arrow components*

The number of elements that are enclosed in an arrow can be varied across the diagram components. It would be very helpful for the end user to see the number of arrow components during the modeling phase. Many times it is possible that the number or input components in an activity to differ from the output ones. This could be happened if the activity selects some components and sending them in a lower activity.

Workflow modeler provides the option to denote the number of components carrying by the arrows. This option represent with the "sz" symbol on the starting point of the arrow.

Figure 6 depicts the way that we propose the formation and the appearance of the carrying information on the arrows. For instance let's assume that the carrying information of the arrow "tool crib" concerns six parts. The activity A22 "pick tool" will select three of them that should be used for processing, according to the information provided by activity A21 "evaluate job progress". Consequently, activity's A22 output arrow will tagged by the index 'sz=3", meaning that its size is equal to three parts (figure 6).

However, it would be more helpful, the number of components to appear at the starting and ending points of the arrow. Due to the big length of IDEF0 arrows that frequently have. The information about the number of components could be useful for arrows that are represent parts, process elements or tools. Finally, the use of enclosed number of components should constitute a very useful piece of information concerning production, time, and cost, which are very useful elements during simulation phase.

## 6. Discussion

Top down hierarchical design fashion, ICOM and component semantics are the most strong points of IDEF0 language. These three design components lead to the establishment of good communication infrastructure among design components and within diagrams of the IDEF0 model. However, the design of complex and tight diagrams is one of its critical disadvantages.

During practicing with IDEF0, we noticed several shortcomings, from the human involvement point of view. Our first thoughts fluctuate upon questions such as: What happens if a human error takes place? Is it designable? Is it predictable? Can the model inform the modeler somehow?

Furthermore, some additional questions that can not be solved with the current version of IDEF0 occupied our minds and created unsolved problems. These questions were: What happens if a justified or unjustified delay occurs in the process? How it can be modeled, or how it can alert the modeler of the possible occurrence?

The answers of these questions along with some additional interference on the parallel processes and enhancements of information annotations of the exact arrow carrying information discussed and investigated herein.

As evidenced in the former paragraphs of the article, the proposed work solve many practical real world problems.

Results shown that the developed version of IDEF0 performs better and helps the modeler to produce comprehend and easy to read diagrams. The produced diagrams reveal cases where the decision maker has to predict them with the use of additional tools and methodologies.

The objective of the article is to enhance the IDEF0 in order to increase modeling power and model BPs more thoroughly and exhaustively. The produced models cover all the cases and events and finally all the unwanted situations that governing BPs because of their uncertainty nature. Uncertainty lies on the participation of human, machines and external factors within the BP.

## References

Bosil, J., Giaglis, G., & Hlupic, V. (2000). IDEF diagrams and petri nets for business process modelling. In Proceedings of the 2000 winter simulation conference (pp. 1676–1683).

Feldmann C. G. (1998). *The practical guide to business process reengineering using IDEF0.* Dorset House Publishing, New York.

Godwin, A., Gleeson, J. and Gwillian, D. (1989). An assessment of the IDEF notations as descriptive tools. *Information Systems*, 14 (1), 13-28.

Hernandez-Matias, J., Vizan, A., Hidalgo, A., Rios, J. (2006). Evaluation of techniques for manufacturing process analysis. *Journal of Intelligent Manufacturing*, 17 (5), 571-583

IDEF0 (1993.) Integration Definition for Function Modelling. FIPS Publication 183. National Institute of Standards and Technology.

J. Mendling (2007). Detection and Prediction of Errors in EPC Business Process Models. Doctoral Thesis, Vienna University of Economics and Business Administration. Vienna, Austria.

Kalnins, A. (2004). Business modeling, Languages and tools. - Progress in Industrial Mathematics. *Proceedings of the 13th European Conference on Mathematics for Industry (ECMI)*. Springer: Berlin, 41-52.

Kalnins, A., Barzdins, J. and Podnieks, K. (2000). Modeling Languages and Tools: State of the Art. *Proceedings of the 2nd International Conference on Simulation, Gaming, Training and Business Process Reengineering.* Riga. 211-214.

Karpe, Y. S. (2006). Weave-Room Performance Decision-Making Process In Textiles: Mapping An Information Engineering Methodology, Doctor of Philosophy dissertation, North Carolina State University

Liu, W. and Fang, K. (2006). Using IDEF0/Petri net for Ontology-Based Task Knowledge Analysis. The Case of Emergency Response for Debris-Flow. *Proceedings of the 39th Hawaii International Conference on System Sciences.*

Ma, Z., Zhang,W. and Ma, W. (2002). Extending IDEF1X to model fuzzy data. *Journal of Intelligent Manufacturing*. 13, 295–307.

Marca, D. A. and McGowan, C. L. (1987). *SADT: Structured Analysis and Design Technique*. McGraw-Hill, Inc.

Mayer, R.J. (1992). IDEF0 function modeling. A reconstruction of the original Air Force Wright Aeronautical Laboratory Technical Report.

Metasoftware. (2007). Workflow Modeler Users Guide. [Online] Available: http:// www.metasoftware.com (May 2007).

Presley, A. and D. Liles (1995). The Use of IDEF0 for the Design and Specification of Methodologies. *Proceedings of the 4th Industrial Engineering Research Conference*, Nashville.

Recker, J., Rosemann, M., Indulska, M., Green, P. (2006). Business Process Modeling: A Maturing Discipline? BPM Center Report, No. BPM-06-20. [Online] Available http://www.BPMCenter.org (May, 2007).

Roger, K.J., Whitman, L., Underdown, R. (1998). The Enterprise Integration issues Encountered with Agile Process Introduction. [Online] Available http://arri.uta.edu/eif/rogersfaim98.pdf (May 2007)

Ross, D. T. (1985). Applications and extensions of SADT. *Computer*. 4, 25-34

Tsironis, L., Anastasiou, K. and Moustakis, V. (2007). A framework for BPML assessment and improvement: a case study using IDEF0 and eEPC, *Business Process Management Journal* (Submitted).

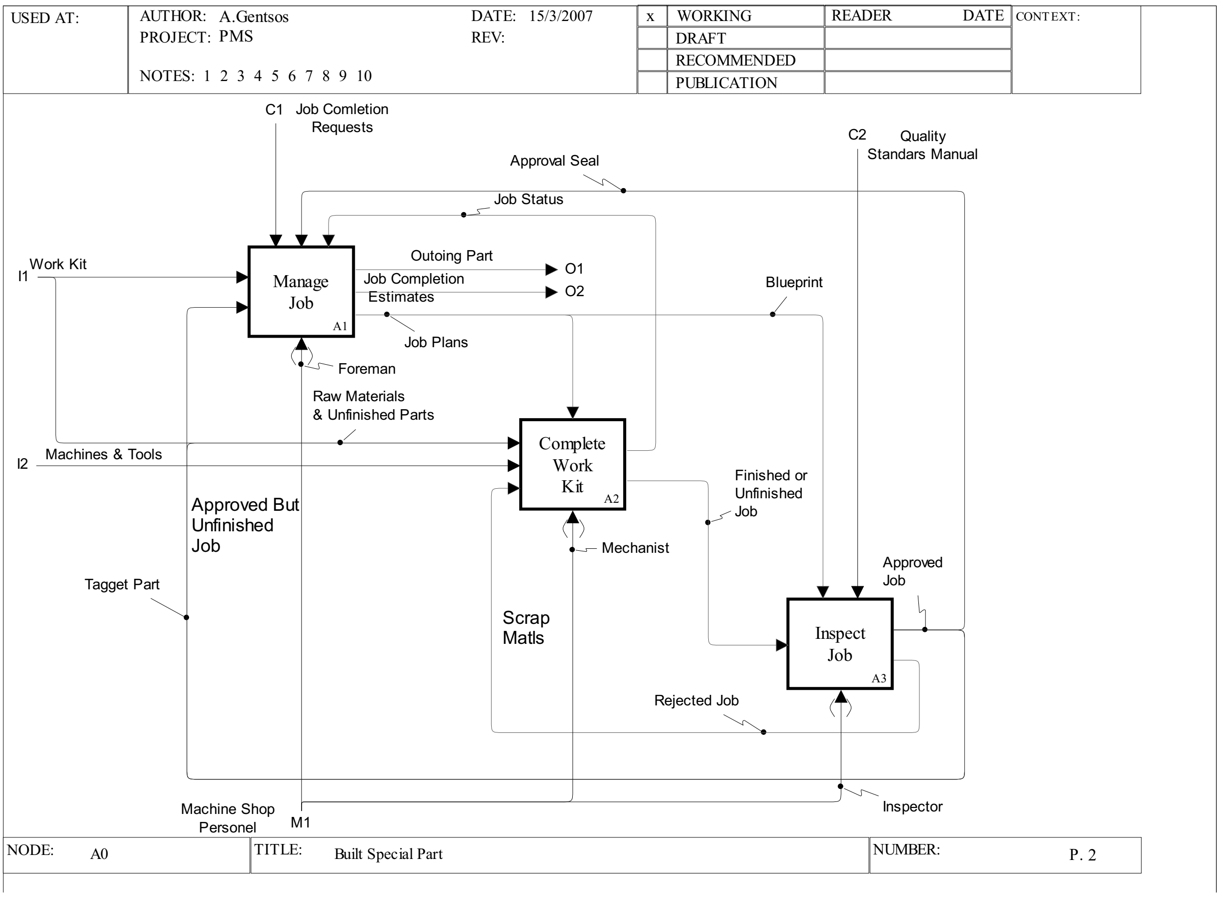


Figure 1. The A0 diagram of the model

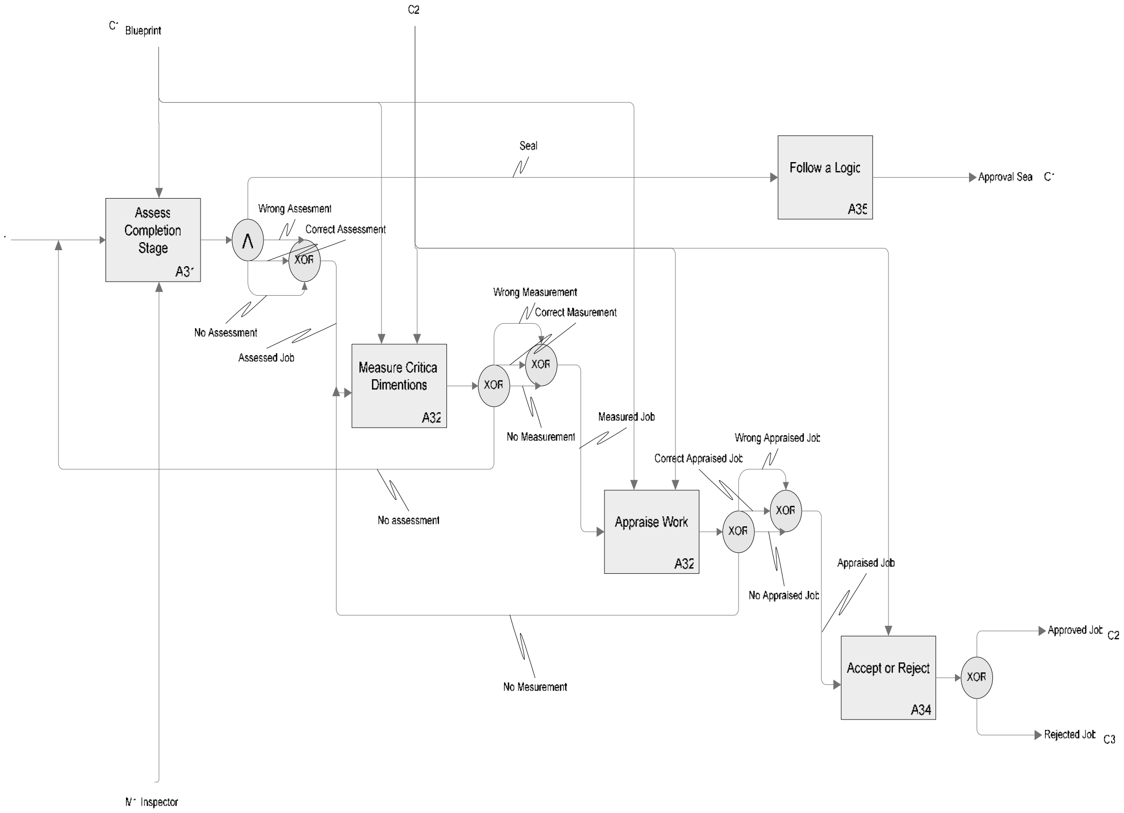


Figure 2. Transformation of A3 diagram

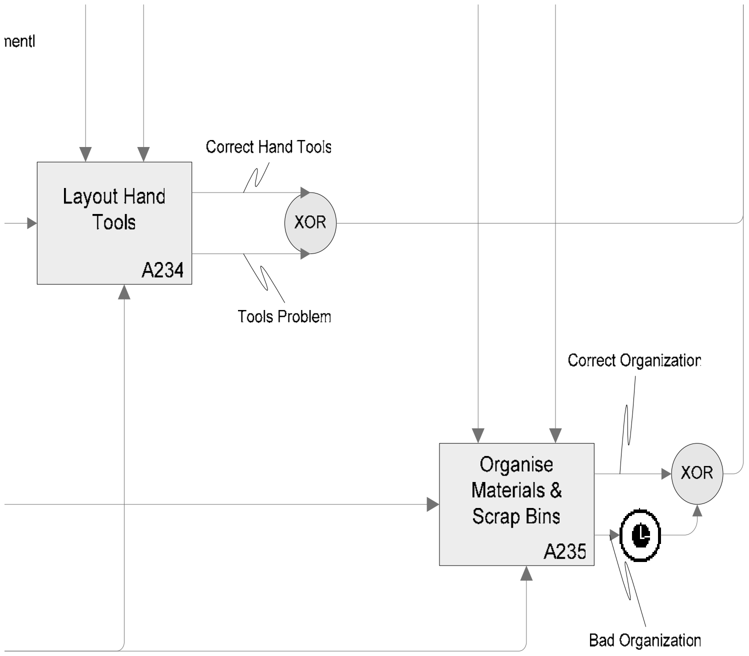
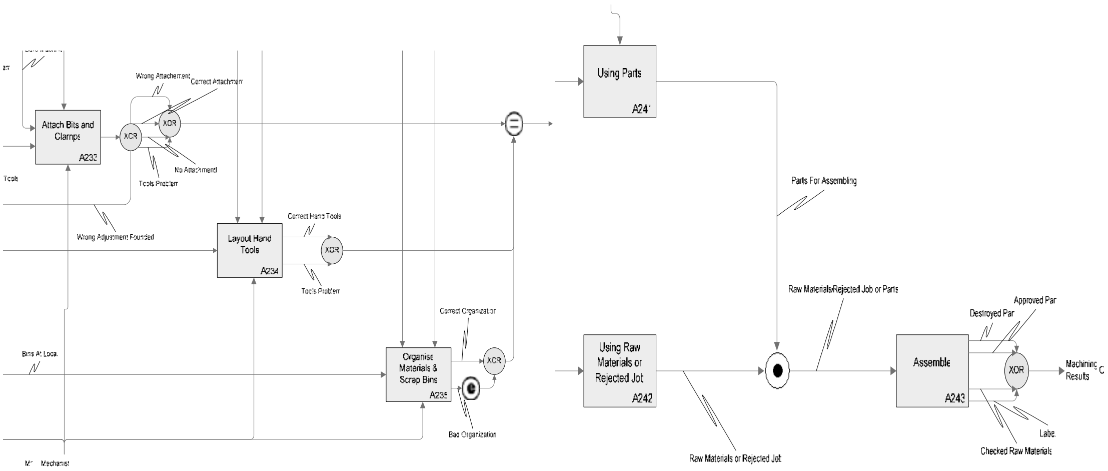


Figure 3. Part of the A23 diagram, depicted the delays operator



4a 4b

Figure 4. Parts of A24 and A23 diagrams presenting the discrimination of parallel and non-parallel processes

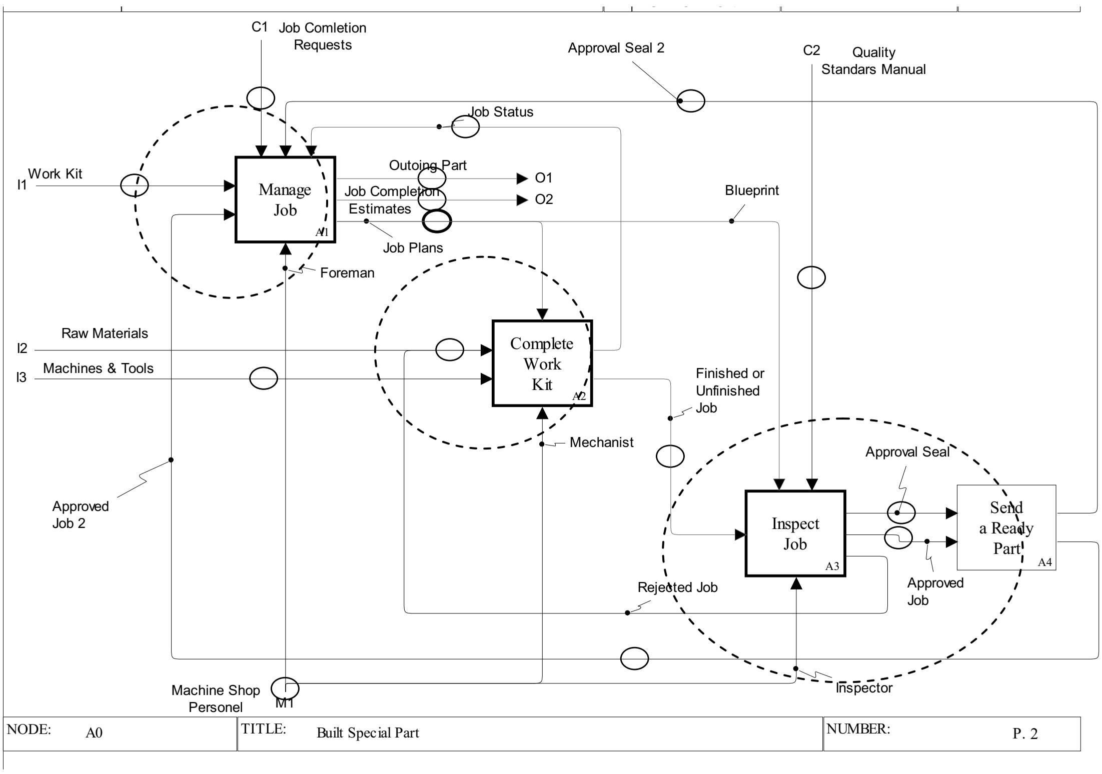


Figure 5. The A0 diagram remarks the 3 design stages

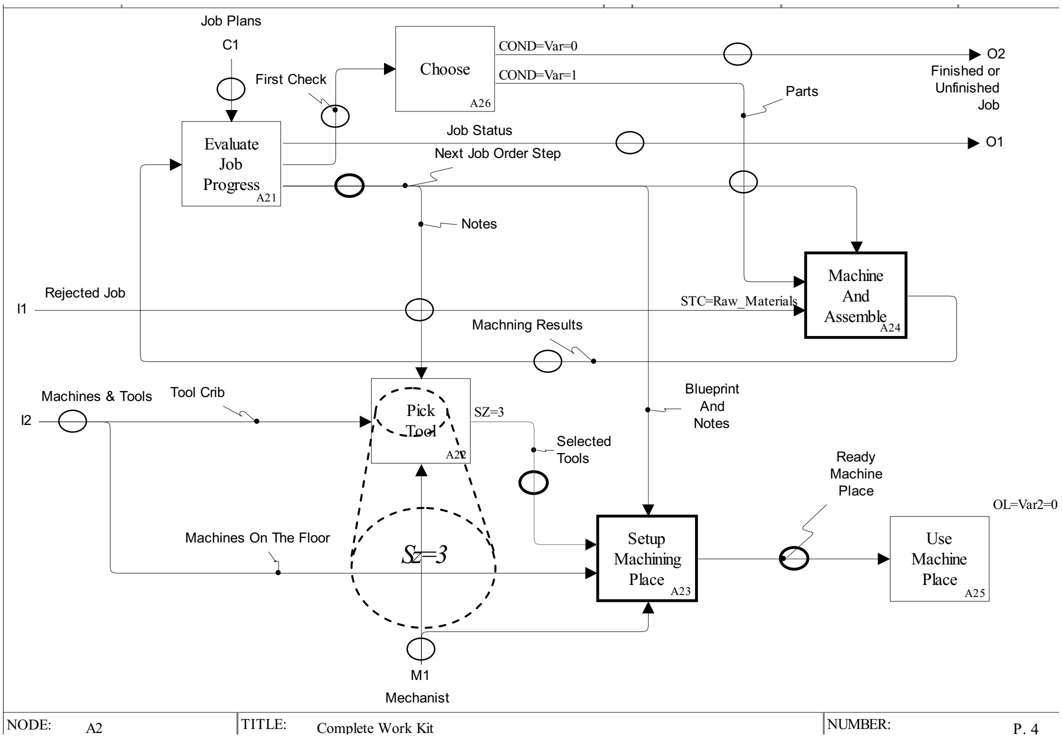


Figure 6. The A2 diagram depicts the appearance of the carrying information on the arrows

Table 1. The proposed logical operators and their notion

| Logical Operator | Label | Description |
|---|---|---|
| ∧ | AND | Both of preceding and following arrows, triggered. |
| XOR | XOR | One of the preceding or following arrows will trigger. |
| 🕒 | DELAY | If the particular arrow triggered, then the process will concern under delay. |
| ⊜ | PARALLEL | The arrows that gathered to the operator came from parallel processes. |
| ⊙ | NON-PARALLEL | The information that comes from the arrow that reach the operator first is admitted to pass first. |