INTERNATIONAL JOURNAL OF ACADEMIC RESEARCH IN BUSINESS & SOCIAL SCIENCES

Vol 3, Issue 9, (2013) E-ISSN: 2222-6990

The Systemic Modeling via Military Practice at the Service of any Operational Planning

Evangelos Papakitsos

Secondary Education Directorate of West Attica, Hellenic Ministry of Education

Department of Linguistics, Faculty of Philology, National and Kapodistrian University of Athens

School of Electrical and Computer Engineering, National Technical University of Athens

E-mail: papakitsev@sch.gr

DOI Link: http://dx.doi.org/10.6007/IJARBSS/v3-i9/200

Published Date: 15 September 2013

Abstract

The presented article demonstrates the value of systemic modeling in operational planning. The systemic modeling facilitates the description, depiction and computerized simulation of complex systems, with their interacting components and the relations between them. This ability is a crucial asset in strategic or operational planning and management of any kind. The article includes a brief presentation of popular systemic modeling tools, as they are applied in information systems and organizational management, along with a recently developed one: the Organizational Method for Analyzing Systems. The latter modeling tool consists of a series of notational facilities, along with the proposed methodology for using these provided facilities. It is briefly used to exemplify two typical applications, which describe the structure and activities of an organization and those of a project. These descriptions have been designed according to the relevant ones of the standard military practice, aiming at noting the significance of the robust but overlooked military practice in civilian, commercial or governmental administrative applications.

Keywords

Systems engineering; information systems; strategic planning; project management; organizational studies

JEL Code: M15

1. Introduction.

Comparatively to the short life of Informatics as a Science, the relevant applications for business and management have a long history. One of the first such applications have been the programming language COBOL (Common Business-Oriented Language, the name says it

all!), developed in the sixties. Another such notorious and widespread application is the spreadsheet software, originally developed as an electronic simulation of the equivalent hard-copies of the accountants (Garidis and Deligiannakis 1993). Nowadays, the conduct of business and management without some kind of software support is unthinkable. Specialized systems have been developed for this purpose, namely Management Information Systems (MIS), Decision Support Software (DSS), etc.

The contribution of Informatics though is not limited to the development of supportive software. The urgent need for software products, world-wide, led also to the development of engineering methodologies. It would be fair here to say that such methodologies existed before, either implicitly or explicitly. The former is evident at least by the monuments of the human culture, globally (e.g., the Great Pyramids; see: URL 1), while the latter has been documented since the early 20th century (e.g., see: Gantt 1910; Gilbreth 1924). In the era of Informatics though, with the usage of computational mathematics, these methodologies acquired sound formalisms. They are also conducted and presented through diagrammatic notations that excessively facilitate their understandability and utility. In fact, each methodology has been so much identical with the accompanying notation that it is named accordingly, as a modeling language, method or technique (e.g., see: URL 2). Some of these methodologies had been, at least originally, software-oriented. Later on, with the development of Systems Engineering (see: URL 3), they became of more general value, in depicting and/or simulating all kinds of structures, processes, functions or activities. A rough comparison of the Systems Engineering terminology (see: SEFGuide 2001) to the Software Engineering equivalent (see: Pressman 1987) will immediately reveal their similarity, if not identification. The rest of this presentation will be focused on the more general systemic modeling, aiming to exemplify its value as supportive business/management tool.

2. Systemic Modeling.

The systemic analysis aims at discovering and describing the components of complex systems, as well as their mutual relations and interaction. The systemic modeling support analysis by providing informative and understandable depictions of the studied systems. Depending on the application, these depictions (models) may be further simulated by a computerized system.

A milestone in the development of systemic modeling had been the creation of the Structured Analysis and Design Technique (see: 2.1. SADT). It has been the raw model for the successive equivalents of the Integration Definition for Function Modeling family (see: 2.2. IDEF) and of the Organizational Method for Analyzing System (see: 2.3. OMAS). Another model is also the Unified Modeling Language which seems more of a collection of various existent modeling techniques (e.g., see: URL 4). This style of modeling (i.e., the SADT-oriented) will be the subject of the present work.

2.1. SADT.

This is a commercial *design language*, created by D.T. Ross (1977) and Softech Inc, in 1973. It is a tool of structured design in a large scale for the description of problems in a hierarchical manner. The basic notation contains rectangles (activities or databases) and arrows denoting flow of data (Input or Output). The functions of the system are bounded by rules (Control) and the parts are conducted by a Mechanism of predefined entities (Fig. 1). Every major

function (or process) can be analyzed at a lower level in three to six minor functions, which can be described in the same manner. One reason of this limitation is the clarity of depiction, in order to avoid large number of intersecting lines, wherever possible.

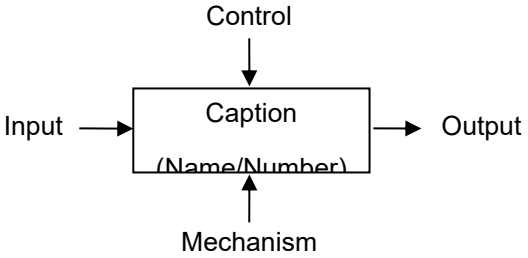


Figure 1: The basic block diagram of SADT/IDEF.

2.2. IDEF.

The IDEF family of models followed the development of SADT, after a requirement of the US Air Force for a function modeling methodology, able of depicting a large number of activities not limited to software engineering analysis, like engineering and reengineering; system control; data flow and others, in details (Grover and Kettinger 2000). The IDEF family was created by the same developers of the SADT in 1981, nowadays consisting of IDEF0 to IDEF5 members ("sub-models"). Each one of these "sub-models" is better suited for a different activity, as a "function" model, an "information" model, a "dynamics" model, a "semantic data" model, etc. In 1993, the IDEF0 model was adopted as a software Standard Modeling Technique by the National Institute of Standards and Technology (of USA) and consequently as a Federal Information Processing Standard (FIPS), until 2002. The IDEF family is a robust and well-tested systemic model, since it is proven after many years of usage by both governmental agencies and the private industrial and commercial enterprises, in a wide variety of applications. The main initial diagrammatic notation is the same as that of SADT (Fig. 1), supplemented by different kinds of arrows and charts, denoting different semantic properties. The initial block diagram is further decomposed in subsequent levels, as previously. There are many commercial computerized tools that facilitate the development, analysis and design of models via the IDEF diagrammatic notation (CASE tools).

2.3. OMAS.

OMAS is another systemic modeling method, also based on SADT. It was originally developed in 2010 (Papakitsos 2010), in order to deal with requirements for clarity, flexibility and descriptiveness through simplicity. The second revised version (OMAS-II) was released in 2011 and tested at the National and Kapodistrian University of Athens, Greece (Papakitsos 2011). It was observed there that the postgraduate engineers (working professionals as well) tended to avoid the usage of large and complex notations, even in favor of improvised diagrams. Unlike other engineering disciplines (mechanical; electric; electronic), in software engineering, the drawing standards (e.g., see: ISO 10303) are neither indisputable nor (even worst) imposed. Hence nothing prevents individually a software engineer from not using a good but complex model, which is described in a manual of hundreds of pages (e.g., see: Idef5 1994), requiring a lot of time to master. The notational simplicity and flexibility can be

achieved through the software Object-Oriented-Design technique of *polymorphism*, where the same sign has a different meaning in a different level of representation, provided there in no ambiguity. For the enhancement of the description power, the information is packed in the initial block diagram through avoiding large numbers of intersecting arrows. The goal is to achieve a balance between simplicity and descriptiveness through polymorphism and information packing.

The last modification towards increased clarity was the usage of all the common, universal journalist-questions in describing the entities of the model, unlike other relevant representations (see: URL 5; NIH 2011) that make a partial use. Natural languages, being the most sophisticated communicational tools in our disposal, contain already the means (words) for revealing information, through questions. Finally in the notation, there is also an emphasis given to explicitly depict the role of the leader, which is crucial in organizational theory (e.g., see: Jones 2003). The last improved version of OMAS is briefly described in the next section (OMAS-III), along with two examples.

3. OMAS-III.

OMAS-III is a systemic modeling technique. It combines a process, designed to achieve a clear definition of the structure and organization of a system, along with a simple yet powerful diagrammatic representation of the relevant meanings. The purpose of its development was the creation of a tool for systemic analysis and design (information; software; business; governmental; non-profitable or natural systems) that provides an integrated and flexible way of describing the system's semantics, according to the application of fundamental principles that are adapted to the particular conditions.

The overall approach is based on the following essential definitions (mainly form Theodorakatos and Tsevrenis 1983):

i) System: a set of elements which are interconnected in order to accomplish a particular task. It is described by relations of causality among its dynamic elements that their nature depends on the type of the system (Cybernetics).
ii) Analysis: The decomposition of an object or a phenomenon to its constituent parts. It can be conducted either by deduction or by induction, depending on the nature of the system (natural or artificial, existent or novel).
iii) Method: A designed process for achieving a particular goal.
iv) Organizational (technique): The application of basic rules, adapted to particular conditions, which aims to achieve the best possible result by using the provided means (resources).
v) Structure: The composition of an entity.
vi) Organization: The arrangement of relations between entities or meanings. These relations can be of any kind, namely qualitative (supportive, obstructive, reinforcing, retarding, balancing, etc.), quantitative (one-to-one, one-to-many, many-to-many), structural (hierarchical, "is-a", "has") or directional (unidirectional, bidirectional).

Consequently to the essential definitions, a *methodology* is required in order to discover the parts of the system and the factors that influence its functionality (system's elements). This methodology is implemented according to the *basic hypothesis*.

3.1. Basic Hypothesis.

The basic hypothesis is that for the entire understanding of a system, answers must be given to each one of the seven (and only) relevant and fundamental questions, concerning the system. These questions are explained with the help of Fig. 2:

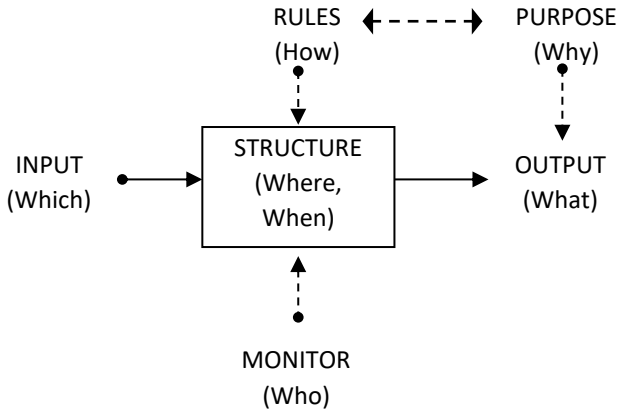


Figure 2: The basic block diagram of OMAS.

i) *Why* does the system exist (PURPOSE)?
ii) *What* is the result of its function (OUTPUT)?
iii) *Which* are the necessary resources (INPUT)?
iv) *How* does it work (RULES)?
v) *Who* operates the process (MONITOR)?
vi) *Where* is it activated (natural or virtual space)?
vii) *When* is it activated (relevant or absolute time)?

Thus, the essential Elements of a system are defined by the answers to the above questions, concerning its *parts* (STRUCTURE) by answering to questions [3.1vi]-[3.1vii], while concerning its *factors* that define the organization (the relations between the parts) by answering to the rest of the questions (3.1i-3.1v). In this way, the description leads to a robust perception of the system.

The main Factors of the system, as mentioned above (3.1i-3.1v), can be further explained respectively as follows:

a) The Purpose of the system is the first-one to be defined (for artificial systems) or discovered as a result of its understanding (for natural systems).
b) The Output of the system includes the results of its function, in terms of measurable (quantitative outcome) and/or describable (qualitative outcome) deliverables or behavior.
c) The Input consists of the resources (means; equipment/technology; raw materials; hardware & software; human resources; know-how and information; capital; energy; infrastructure) that are required for the Output.
d) The Rules consist of the relevant legislation, professional ethics, internal regulations (for a corporation/agency), natural or social laws, constraints, conditions and restrictions of any kind that have a regulatory influence on the system.
e) Monitoring is applied by individuals or groups of them (Boards, Committees, staff, etc) that have the leading roles of administration, management, supervision or crucial operation of the system's Parts. Here, the observers of a natural system (i.e., in

academic research) are also included, since the degree of their personal objectivity and precision is important for the understanding of the system.

Factor [3.1iii] constitutes the main data of the system. The main data are transformed to the Output by the Parts (STRUCTURE) of the system, through its activation, with the influence of factors [3.1iv] and [3.1v]. The Parts allocate the system's functions in specific *levels* (space element: [3.1vi]) and appropriate *phases* (time element: [3.1vii]). The combination of Levels and Phases composes the Structure of the system. Every Part of the system, either static or dynamic, constitutes a sub-system, which is a system of its own that can be recursively described in the same manner.

3.2. Semantic Notation.

OMAS-III is accompanied by a diagrammatic notation that graphically depicts the semantics of the methodology. The notation is based on that of SADT (as the IDEF family does), but it is modified according to the manner of Nassi-Schneiderman Charts (N-S charts). The N-S charts (Nassi and Schneiderman 1973) use a notion similar to Venn diagrams (Set Theory of Mathematics), where the shape of a larger set encloses the shapes of its sub-sets. In this way, a complex figure becomes simpler by reducing the number of intersected arrows required. Additionally, such a modification allows the depiction of two simple hierarchical levels of representation in a single figure, instead of two remote ones. The difference is exemplified in Figures 3-4:

Both processes (A , B) have common RULES, common MONITOR, a common input element (INPUT-C) and a single final OUTPUT, since the output of PROCESS-A (Outcome-A) is directed only to PROCESS-B (Fig. 3). It is noticeable at this point that there is no clear indication whether the intersecting lines are branching or crossing. By enclosing processes {A} and {B}, as sub-processes to a larger one (an enclosing rectangle), the depiction becomes obviously simplified and less space-demanding. All the common elements of the two sub-processes (A , B) are directed only to the enclosing rectangle (Fig. 4). Processes {A} and {B} can be arranged either vertically or horizontally, according to the needs of the description and to the actual surface available on paper.

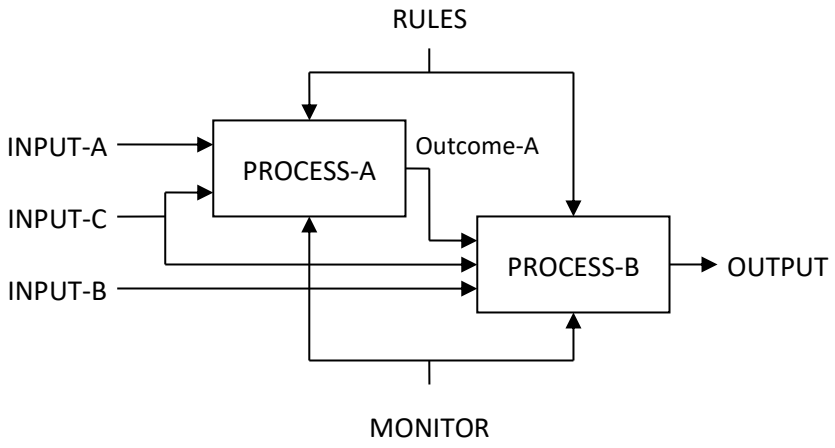


Figure 3: SADT style of depiction.

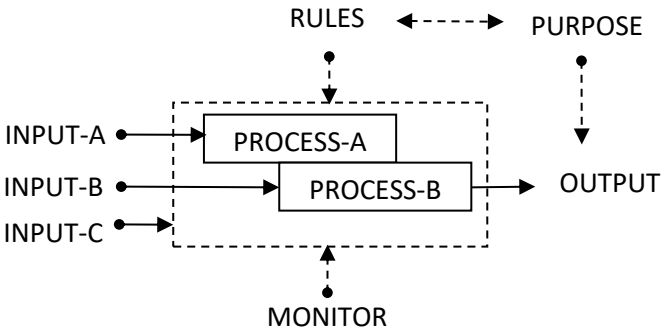


Figure 4: OMAS-III style of depiction.

The next important modification is the selective usage of shapes and arrows. The different shapes are not all equally suitable, having standard word-processing and desktop software tools. An important goal of OMAS's development was to be easily made by the standard designing facilities of the common inexpensive desktop-processing tools (e.g., unlike IDEF). In this respect a labeled circle, like those predominantly used in Data Flow Diagrams (DFDs), occupies too much space on paper compared to a labeled rectangle. A rhombus, used for branching in Flowcharts, is equally space consuming to a circle. The comparison is visible in Figure 5, where the useless space is shadowed. The larger the label the more useless space is occupied by a circle or a rhombus.

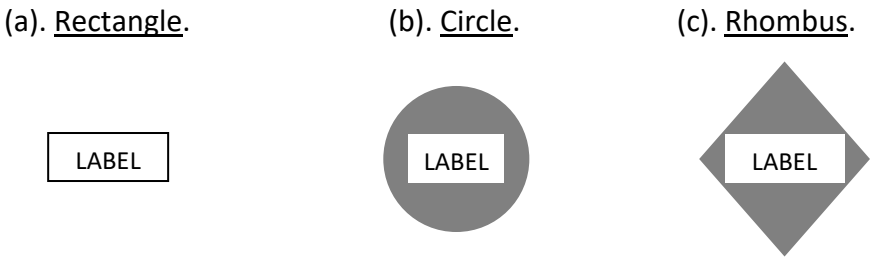


Figure 5: Labeled shapes.

Hence in OMAS-III, space consuming shapes are generally avoided. Circles are used only for denoting a node of conditional branching, labeled just with a couple of symbols or a number. This designing decision allows not only the compression of more information in a single sheet of paper but also to avoid splitting a diagram in more than one page. The main notation (11 out of 30 symbols) is presented in Fig. 6.

Finally, the last modification to be addressed concerns the number of shapes to be used semantically. OMAS-III aims (just like the IDEF family) at describing system models, function/structure/process models, information/software/data models, etc. Because of the wide-spread computerized decision support tools nowadays, it is usual that the representation of a system may contain all the above mentioned modeling. There are modeling languages (like UML, see: URL 4) that use different or altered notation for every

different such modeling, although these are descriptions (like process versus data modeling) that may not appear together at the same level of representation. To contain many different notations makes the modeling tool more difficult to memorize and thus more prone to errors. This is why OMAS-III was developed according to the technique of polymorphism, as mentioned already (see: 2.3. OMAS).

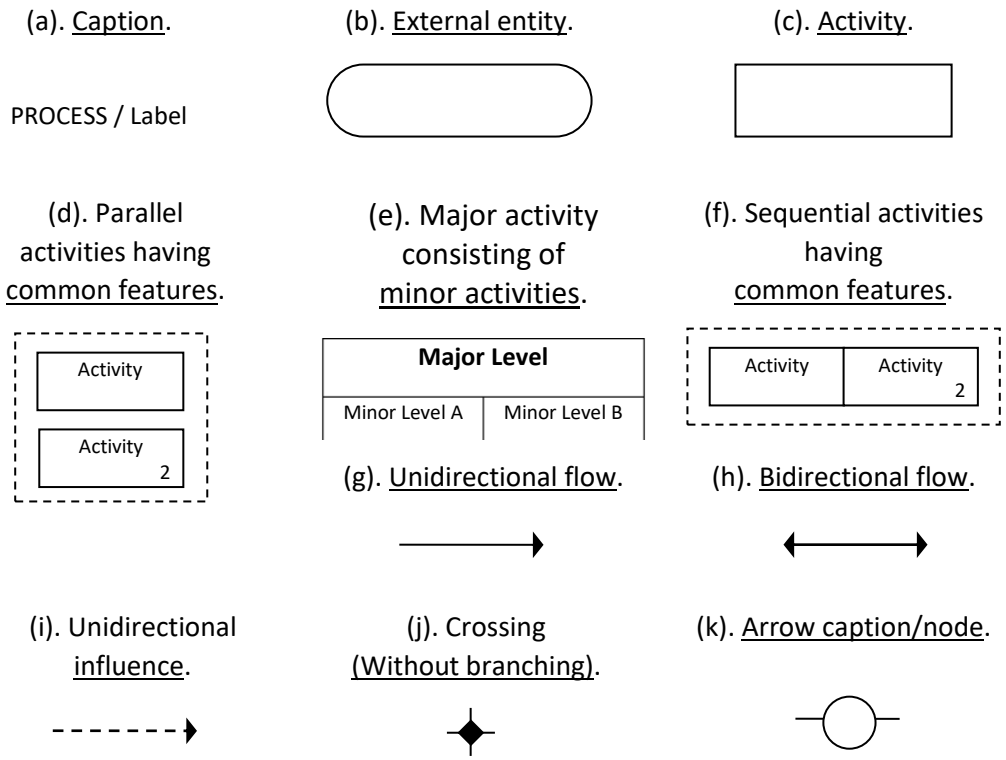


Figure 6: Main notation.

3.3. Implementation.

The analysis is conducted by the definition of all the Elements of a system (Parts and Factors). The Factors are distinct but interactive. So, every change of one's features or status may affect the others. Consequently, their study must include all the possible relations between them. These relations can be qualitative or quantitative of any kind, while the influence between two Factors can be unidirectional or bidirectional. Other types of relations (see: 3.vi) can be depicted either as captions or as circled symbols (see: Fig. 6.k) attached to the arrows. Their definition can be conducted through the following process (Figure 7):

- Definition of the PURPOSE of the system.
- The required OUTPUT is defined according to the PURPOSE.
- All the RULES concerning the system's functions are collected. Their relations to the PURPOSE and the OUTPUT are thoroughly examined. Since the purpose of an artificial system has to be achievable, any condition that hinders the achievability of the system must be discovered, as soon as possible, in order to redefine the OUTPUT before it is too late for changes.

- The INPUT that will potentially provide the required OUTPUT is defined and collected or secured.
- The desired and necessary qualifications of the persons who will manage the process (MONITOR) are defined, before the proper persons are sought and appointed to the task.

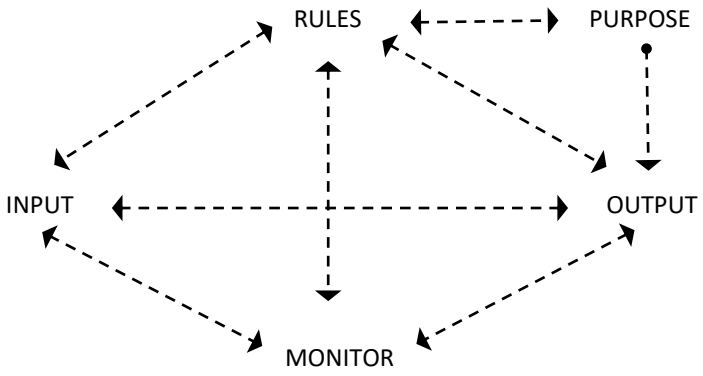


Figure 7: Interaction between Factors.

After the definition of the Factors, the Structure of the system is to be defined or discovered, through the next steps:

- The OUTPUT is gradually decomposed in a number of partial outcomes and their corresponding functions, regarding space (Levels) and time (Phases), until the INPUT is reached. The most important criterion of this decomposition is the greatest possible independence of the above partial functions. Every partial outcome/function is allocated to a distinct Part of the system. The process follows a step-wise and recursive manner that is applied to every major part, in order to define the minor parts (sub-parts), until the problem (structure's definition) is solved.
- At every stage of the decomposition, the partial outcomes are examined, regarding their relations to the other Factors of the system. If necessary, the partial outcomes are revised.
- The Structure of the system is finalized through the definition of its Parts.

As the system starts operating, the outcomes are constantly monitored and evaluated regarding the PURPOSE and the RULES. The evaluation leads to *conclusions*, which are a distinct deliverable of the OUTPUT. The way that these Conclusions influence the rest of the Factors is carefully examined, since they constitute the *feed-back* of the system that may cause the future revision of other Factors. Any similarity of the entire above process to the standard software engineering analysis one is not accidental!

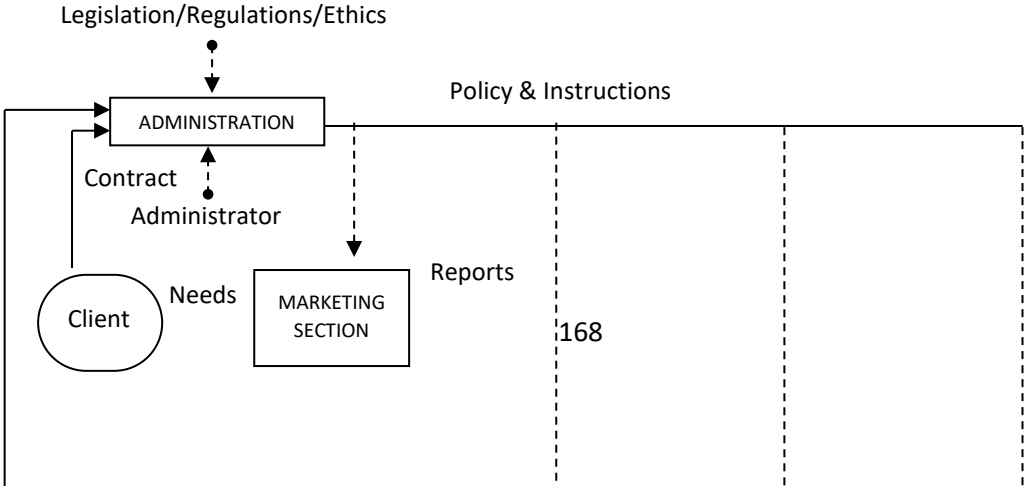
4. Applications.

It is time to exemplify two kinds of simplified trivial applications: a long-terms one and a short-terms one, to see in which manner OMAS-III can be used to describe these applications and facilitate the understanding of a systemic approach. A long-terms application may describe the generic structure of an organization (public or private enterprise), which is a trivial

example. A short-terms example may describe the initiation of a project, hence contributing to the management of the whole process.

4.1. Generic Enterprise.

In Fig. 8, the simplified generic structure of an enterprise is described via the notation previously presented. The major standard sections or departments of any organization are depicted by rectangles (ADMINISTRATION, MARKETING, FINANCIAL, PERSONNEL and OPERATIONAL). Every section is monitored by a Director or a Manager accordingly, in a fairly standard way, depicted below each rectangle. Above the rectangles, the principles that guide their activities are visible (e.g., Policy & Instructions). Every section receives an input on its left and produces and output on its right. The output of a section can be the input of another. An external entity (Client) appears to have a bidirectional influence (dashed line) with an output of the OPERATIONAL SECTION (Products, Services), to the extent that the needs of the client dictates the nature of the final output of the enterprise (Requirements & Sales). From every section of the enterprise, reports are sent to the Administration, as the feedback of their activities. Notably, the depicted structure of the enterprise is a direct modification of the equivalent structure of a battalion headquarters (Papakitsos 2010).



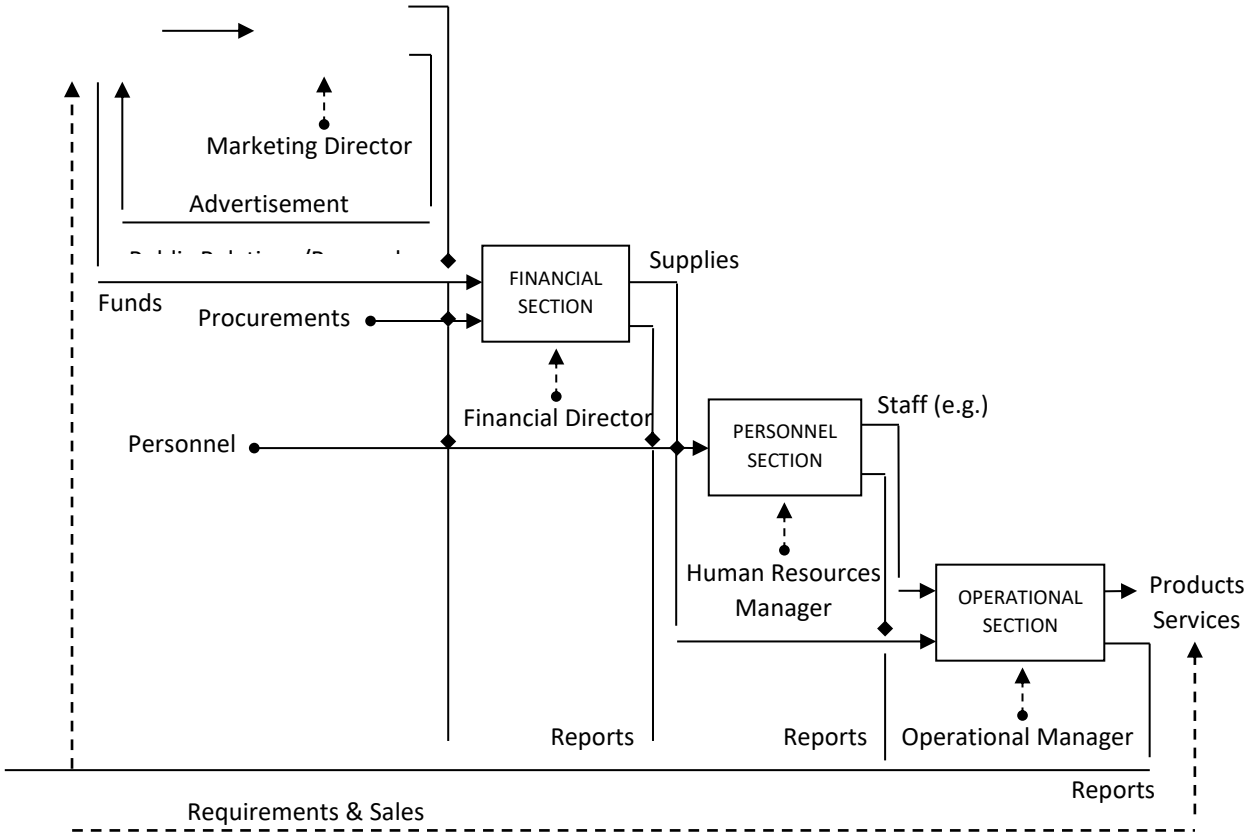
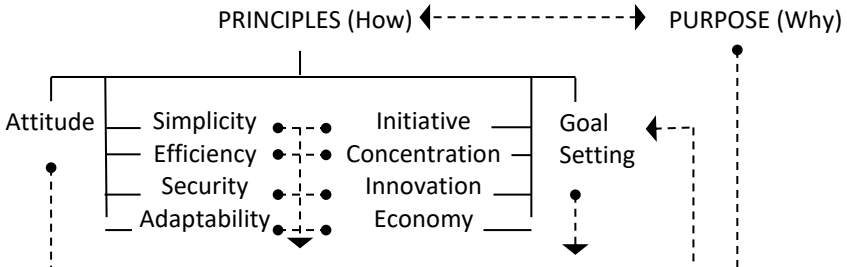


Figure 8: The generic description of an enterprise.

4.2. Generic Project.

In Fig. 9, the elements of a project are arranged in a way that their relations are visible. The PROJECT is composed of two major phases: PREPARATION and REALIZATION. At the phase of PREPARATION, the two crucial activities are the Training of the personnel that will conduct the realization of the project and the Analysis and Planning of the realization. The phase of REALIZATION can be divided in two parts, the Application of its main activity (e.g., for a business enterprise, the creation of a product) and the Exploitation of the Application's outcome (respectively say, the promotion of the product). The output of the project (RESULTS) can be either successful or not. Actually, there can be a wide spectrum of results between a total Success and a total Failure. The project team (PARTNERS) monitors the process and manages the flow of recourses (unidirectional dashed line). The resources (Capital and Materials, among others) are interconnected, while consideration has to be given to potential competitors (if applicable: "{}").



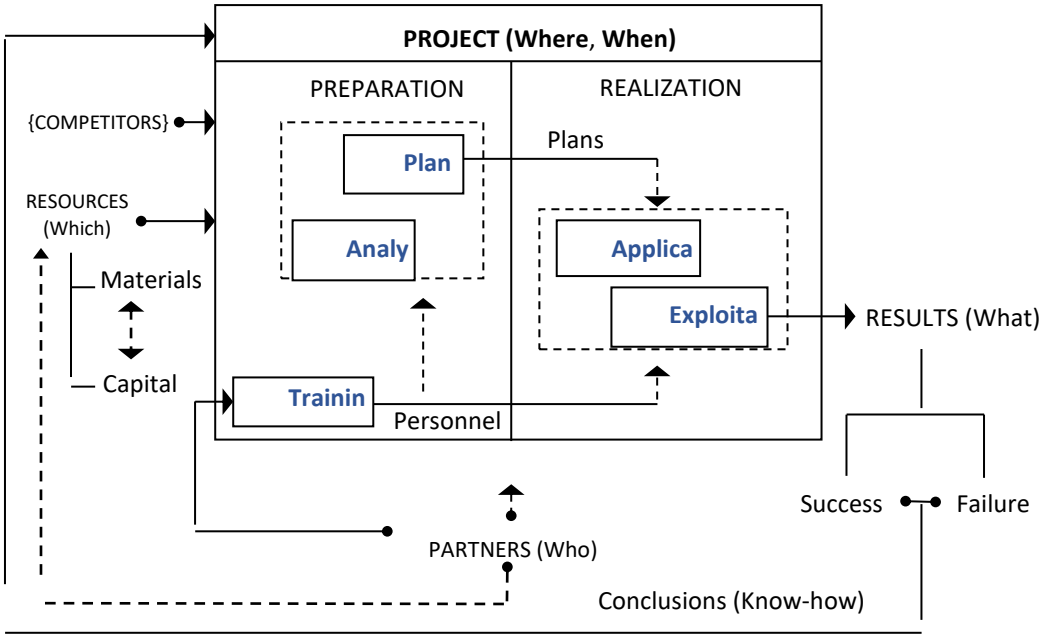


Figure 9: The generic description of a project.

The major rules for guiding the process can be described by ten principles, being positioned above the Project's area (rectangle). These principles may be presented as follows:

- The positive Attitude towards success that has to be shared by the personnel of the project. It depends on the ability of the leader to motivate the personnel, to instruct, to inspire and to provide empowerment whenever necessary.
- The Simplicity of the design, in order to achieve understandability (and comprehension) of the activities.
- The Efficiency of every activity, being evaluated by measurable criteria, for a rational usage of the resources.
- The protection of the crucial aspects of the process (Security), like confidential information, data integrity, personnel's safety, equipment's readiness.
- The Adaptability of the design in order to cope with unexpected conditions that will almost inevitably appear during the execution of the project.
- The Initiative of such actions that will prevent delays, excessive cost or application errors (to mention a few).
- The Concentration of the resources and effort to those activities that have the greatest impact to the outcome.
- The Innovation, necessary to cope with an ever changing and demanding environment.
- The Economy of the resources-consumption to the appropriate predefined level for achieving each task.
- The Goal Setting has to determine the series of intermediate goals towards the final outcome, that they will be measurable, achievable and monitored throughout the project.

Equivalently to the previous example (see: 4.1. Generic Enterprise), this one is also a direct modification of a military operation, as depicted via OMAS (Papakitsos 2010). All of the above

presented principles are modified strategic standards, as recognized by relevant military experts/regulations (see: Ayalon 1987; Paraskevas 1987).

5. Conclusions.

The key-features of a user-friendly systemic model are the simplicity and flexibility of the notation, the descriptive power and the clarity of the semantics. The goal is to relieve planning experts (leaders, designers, analysts, etc.) from the burden of memorizing rigid and complex formalities, and to let them focus on the problem in hand, without compromising the essential utility of the modeling tool. In this respect, common/standard and inexpensive desktop-software can be utilized to assist the modeling process.

Hopefully, the value of using systemic modeling in management, to support planning and administrative activities, has been demonstrated in this presentation through OMAS-III. Besides the included examples, the particular systemic modeling technique has been used to describe various types of activities, namely:

- the structure and actions of a shipping escort agency (anti-piracy operations);
- the designing of martial arts training curricula;
- the depiction of the elements of a crime (Criminology);
- the description of a standard educational/training activity;
- for academic writing guidance;
- in perceiving language as a system (ongoing).

The possible applications of such a modeling are not limited to a particular field, being compatible to the Systemic paradigm in scientific/academic and operational research.

References

Ayalon A. (1987). Advantages and limitations of the War Principles. *MILITARY REVIEW*, July Issue.

Gantt H.L. (1910). Work, Wages and Profit (published by The Engineering Magazine, New York, 1910). Republished as: Work, Wages and Profits. Easton, Pennsylvania: Hive Publishing Company, ISBN 0879600489.

Garidis P.K., Deligiannakis E.N. (1993). *Dictionary of Computing* (in Greek). Athens: Diavlos Publications, 6th edition.

Gilbreth F. & L. (1924). *The Quest of the One Best Way*. Purdue University, Frank and Lillian Gilbreth Papers, 1924.

Grover V., Kettinger W.J. (2000). Process Think: Winning Perspectives for Business Change in the Information Age. IGI Global, DOI: 10.4018/978-1-878289-68-1.

Idef5 M.T.D. (1994). Information Integration for Concurrence Engineering (IICE): IDEF5 Method Report. Texas: Knowledge Based Systems, Inc. (September 1994), USA.

ISO 10303: *Automation systems and integration - Product data representation and exchange*. TC184/SC4.

Jones G.R. (2003). *Organizational Theory, Design and Change, 4 edition*. Pearson Education, ISBN: 0131227017.

Nassi I., Schneiderman B. (1973). *Flowchart Techniques for Structured Programming*. SIGPLAN Notices, ACM, August Issue.

NIH (2011). *Business Architecture Modeling Methodology*. National Institutes of Health, Enterprise Architecture, USA.gov. Retrieved from:

http://enterprisearchitecture.nih.gov/BusinessModelingMethodology.aspx (accessed: August 2013) .

Papakitsos E. (2011). *Basic meanings of Linguistic Engineering (*in Greek). Course Manual: "Technoglossia VI" MSc Programme in Linguistic Engineering. University of Athens, Faculty of Philology - National Technical University of Athens, School of Electrical & Computer Engineering. PDSA 673: E. K. Thessalou, TRN 062079143, Athens.

Papakitsos E. (2010). *Organizational Method of Analysing Systems* (in Greek). Diagrammatical technique of Systemic Theory along with application examples. PDSA 626: E. K. Thessalou, TRN 062079143, Athens.

Paraskevas G. (1987). The Battle of Gavgamila. Application of War Principles (in Greek). *ARMY REVIEW*, Issue of February, Publications Directorate of the Hellenic Army General Staff.

Pressman R. (1987). *SOFTWARE ENGINEERING: A Practitioner's Approach*. McGraw-Hill.

Ross D.T. (1977). *Structured Analysis: A Language for Communicating Ideas*. IEEE Trans. Software Engineering (January 1977), pp.16-34.

SEFGuide (2001). *Systems Engineering Fundamentals*. Department of Defence, Systems Management College, Defense Acquisition University Press (January 2001), Virginia, USA.

Theodorakatos D., Tsevrenis I., (Eds), (1983). *C. Giovanis New Dictionary – Thesaurus of the entire Greek Language* (in Greek). Athens: C. Giovanis Publications.

URL 1: http://en.wikipedia.org/wiki/Management. Wikipedia, the free encyclopedia (retrieved August 2013).

URL 2: http://en.wikipedia.org/wiki/Function_model. Wikipedia, the free encyclopedia (retrieved August 2013).

URL 3: http://en.wikipedia.org/wiki/System_engineering. Wikipedia, the free encyclopedia (retrieved August 2013).

URL 4: http://en.wikipedia.org/wiki/Unified Modeling Language. Wikipedia, the free encyclopedia (retrieved August 2013).

URL 5: http://en.wikipedia.org/wiki/Business_architecture. Wikipedia, the free encyclopedia (retrieved August 2013).