Augmenting SADT with Respect to Timing Constraints, Formalization and Dependability Evaluation

Lakhoua M.N., Ben Salem J., El Amraoui L.
Research Unit Signals and Mechatronic Systems,
SMS, UR13ES49, National Engineering School of Carthage, ENICarthage, University of Carthage
MohamedNajeh.Lakhoua@ieee.org

Abstract—After a presentation of the Structured Analysis Design Technique (SADT) method used in designing computer integrated manufacturing systems, we present a review on this method and its applicability in the industrial and pedagogical fields. Thus, some applications of the SADT method that have been presented in various researches are presented. Previous researches showed that any kind of system can be modeled using SADT. So, this method is a confirmed way to model any kind of domain. Then, we present some researches to augment the SADT method with respect to timing constraints and formalization in one hand and to take into account the dependability evaluation, in other hand. Finally, we present how SADT augments software development methods.

Keywords—SADT method; timing constraints; dependability evaluation; domain modeling.

I. Introduction

Early in the system design process, a variety of a design method is usually dictated by what methods the designer has earlier used, not by an open selection process. In fact, particular interest in the use of graphical modelling methods and techniques to aid changes in system operations and the interactions of staff to effectively build and use modelling for analysis, design and communication of systems in the manufacturing industry.

Besides systems specification supposes two essential characteristics: temporal evolution of the system components and the system - environment interaction. Indeed, the complexity of relations between a system and its environment is especially verified in the domain of process conduct.

Among the techniques of system specification, we mention: (1) methods of analysis that permit to systematize and to canalize the various perceptions, (2) specification languages possessing syntax and very definite semantics, and (3) simulation languages.

Structured Analysis Design Technique (SADT), which was designed by Ross in the 1970s [1] [2], was originally designed for software engineering but quickly additional areas of application were found, such as aeronautic, production management, etc.

SADT is a standard tool used in designing computer integrated manufacturing systems [4] [5]. In fact, a significant complexity of automated manufacturing systems requires methods and tools which must allow preliminary safety analysis beginning right from the start of the design cycle [5].

In order to present how SADT is a proven design method, we present some researches in this paper: (1) the extended SADT method with respect to timing constraints and formalization, (2) the Safe-SADT method for dependability evaluation and (3) the augmentation approach for software development methods

This paper can be loosely divided into five parts: First, we present the SADT method and second, we present a review on SADT and its applicability in the industrial fields. In section three, we present some researches to augment the SADT method in order to take into account the timing constraints, the formalization and the dependability evaluation. Then, we present how SADT augments software development methods. Finally, the last section presents conclusion and future work.

II. Presentation of the SADT method

As the inventor of SADT, Ross was an early developer of structured analysis methods. Through the 1970s, along with other contributors from SofTech, Inc., Ross helped develop SADT into the IDEF0 (Icam DEFinition for Function Modeling) method for the Air Force's Integrated Computer-Aided Manufacturing (ICAM) program's IDEF group of analysis and design methods [6] [7].

Although SADT does not require any specific supporting tools, several computer programs implementing SADT methodology have been developed. In fact, IDEF0, a function modeling building on SADT, is designed to characterize the decisions, actions and activities of an existing or prospective organization or system [8].

IDEF0 graphics and accompanying texts are presented in an organized and systematic way to gain understanding, support analysis, provide logic for potential changes, specify requirements and support system-level design and integration activities. IDEF0 may be used to model a wide variety of systems, composed of people, machines, materials, computers and information of all varieties, and structured by the relationships among them, both automated and non-automated.

ISSN 2229-5518

For new systems, IDEF0 may be used first to describe requirements and to specify the functions to be carried out by the future system. As the basis of this architecture, IDEF0 may then be used to design an implementation that meets these requirements and performs these functions. For existing systems, IDEF0 can be used to analyze the functions that the system performs and to record the means by which these are done.

Figure 1 shows the Top-down, modular and hierarchical decomposition of SADT.

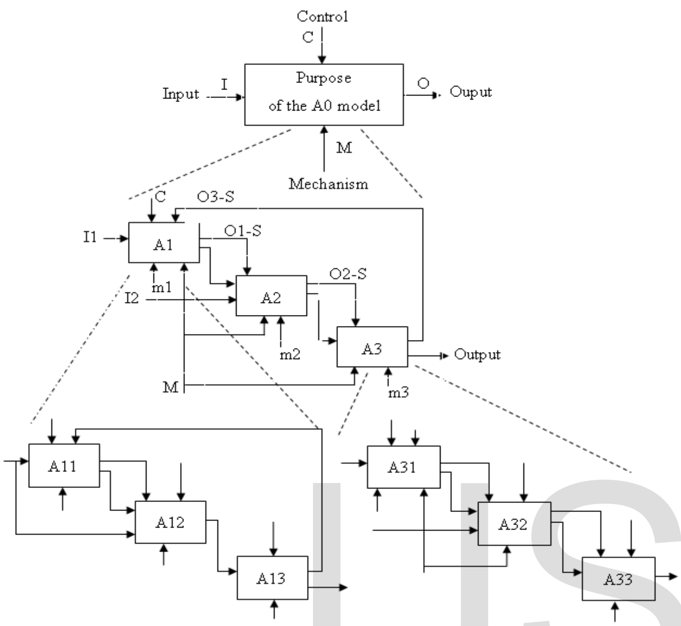

Fig. 1. Top-down, modular and hierarchical decomposition of SADT.

The boxes called ICOM's input-control-output-mechanisms are hierarchically decomposed. At the top of the hierarchy, the overall purpose of the system is shown, which is then decomposed into components-subactivities. The decomposition process continues until there is sufficient detail to serve the purpose of the model builder. SADT/IDEF0 models ensure consistency of the overall modelled system at each level of the decomposition.

Unfortunately, they are static, i.e. they exclusively represent system activities and their interrelationships, but they do not show directly logical and time dependencies between them. SADT defines an activation as the way a function operates when it is 'triggered' by the arrival of some of its controls and inputs to generate some of its outputs. Thus, for any particular activation, not all possible controls and inputs are used and not all possible outputs are produced. Activation rules are made up of a box number, a unique activation identifier, preconditions and postconditions. Preconditions and postconditions describe what is required for and what results from the activation. Both preconditions and postconditions are logical expressions of ICOM codes, where each ICOM code identifies a single control, input, output, or mechanism arrow for that particular box. When an ICOM arrow does not participate in activation, it is simply omitted from the precondition. Similarly, when some of the outputs of a box are produced during activation, the ICOM codes for those outputs not generated are omitted from the postcondition. A precondition expresses the required presence (or absence) of any of the objects associated with the inputs, controls, outputs, or mechanisms involved in the activity. A post condition indicates presence (or absence) after the activity has occurred.

In Tunisia, there are some thesis that have presented some applications of the SADT method for the system analysis of a grain silo [9], the proposal of a global methodology of a Human Resources management [10], the modelling of the performances of a Tobacco Manufacturing [11] and the modelling and the command of an agricultural greenhouse climate [12].

In the world, we present some thesis that have used the SADT method: indirect conditional preventive monitoring of a textile manufacturing [13], design of a syntax validation tool for requirements analysis using SADT method [14] (Dong H.J., 1990), piloting of the production systems towards the global performance [15] (Sénéchal O., 2004).

III. Review on the SADT method

This section presents some studies of the SADT method and its applicability in the industrial and pedagogical fields that have been presented in various researches:

Researchers, Lauras, M. & al. [16], have proposed an approach based on enterprise modelling tools (GRAI, SADT/IDEF0) that enables the integration of the best practices defined by these methods. Besides the indicators following the results and determinants, three types of indicators are introduced to analyze the performance: the facility viewpoint measures, the appropriateness of the resources available and the determinants of the activity.

Researchers, Lezina, O.V. and Akhterov A.V. [17], have presented the structure of information component of pedagogical knowledge management system in the chair of technical university and the possibility of using the ontologies and SADT methodology for the design of information component of such system. In fact, the modern stage of social development, the emergence of the knowledge based economy; the rapid dissemination of educational information and telecommunication technologies, as well as modernization of the system of higher education makes new requirements to preparation of graduates. This requirement necessitates the need to create and use of flexible pedagogical knowledge management system.

Researchers, Yulian C. & al. [18], have investigated and analyzed the production workflow in small and medium toy manufacturing enterprises by SADT and simulation analysis. They find out that tracking information is incomplete and information flow and material flow are out-sync due to lacking material and production process collaboration in current system. Thus, the tracking objective creates a need for systems to collaborate material flow and production flow in manufacturing enterprises. In fact, material safety and traceability is of great importance in toy manufacturing because there have been tougher requirements on toy product safety imposed by new international regulations.

Researchers, Demri A. & al. [19], have proposed to employ SADT, FMEA, SEEA and Petri networks methods to study a mechatronic system. In fact, a study of system reliability is generally preceded by a functional analysis, which consists of defining the material limits, the various functions and operations realized by the system and the various configurations. This stage does not give information about the modes of failure and their effects. It is necessary to complete it by a second one taking into account the dysfunctions in order to model suitably a complex system with Petri networks.

Researchers, Plateaux R. & al. [20], have proposed to integrate the entire downward side of the design V-cycle in order to achieve to a modelling continuity through the different levels of design approach (requirements, functional, components and structural). For this, they have proposed a hybrid methodology based on several tools, languages and methodologies such as SADT, SysML, Modelica, in a single environment: Dymola.

Researchers, Wenan T. & al. [21], have proposed respectively SADT-based e-learning process architecture and an SOA-based knowledge management mechanism. After that, they have discussed the process management model of e-learning from an overall lifecycle perspective. At last, the corresponding knowledge management architecture is presented to further support this process management.

Researchers, Goldshtein M. and Matveyev A. [22], have considered the subsystem of input and preprocessing of the information as the constituent part of an automated system for scientific research. With the purposes of systematization of methods and construction tools of the given subsystem depending on time and precision characteristics the informal model of its operation was developed. The subject knowledge of the researched object of modeling, methodology SADT, and also international standards were put at the heart of formalization.

Researchers, Ruzaev E. & al. [23], have discussed the use of SADT technology for the analysis of the world popular experiment of ISO 9000 application allowed to realize the possibility of using its model for a university. SADT technology application allows cutting down expensive errors at the early stage of system creation to follow the process of adapting decisions. In fact, the problem of international market development of educational services includes the supply of mutual diploma acknowledgement. The achieved effect is in the essential unification and simplification of documentation necessary for mutual acknowledgment of diplomas and to guarantee the teaching quality of graduates.

Researchers, Ahmed F. & al. [24], have proposed the use of SADT from software engineering to develop conceptual models. In fact, SADT has proven to be successful in the development of software systems, specifically in the requirements gathering phase. This researchers have contributed to the area of Conceptual Modeling (CM) by proposing a new framework for developing conceptual models which focuses mainly on the first phase of CM, that of System Description (SD). A simple case, the Panorama Televisions production plant, is used to illustrate the application of this approach. The benefits and limitations of this framework are discussed.

Researchers, Yahmadi R. & al. [25], have presented a degradation analysis of the lead acid battery plate during the manufacturing process. The different steps of the manufacturing process of plate such as manufacturing of lead oxide, paste mixing and manufacturing of grid, pasting, curing and drying are describe d by SADT. The general analysis of all the causes and potential factors causing a low quality of the plate during the manufacturing process is created by the Ishikawa diagram. This description is completed by the Causal Tree Analysis in order to seek the various possible combinations of events leading to the low quality of lead acid battery plate during the pasting, curing and drying process.

Researchers, Zenniz Y. & al. [26], have presented the dependability of an automatic detection and extinction system with Halon. The risk control and analysis is done using three analysis methods, SADT, FMEA and FTA. The objective of the research is dependability planning optimization with the identification of the potential risks and these consequences on system. The optimal recommendations must be proposed. Decision tools used to improve production will be proposed. To achieve these objectives a detailed description of process structure and the main principles of both methods are given. These methods are applied on the system with a comparative study between the results given by these methods.

Researchers, Puilk E. & al. [27], have developed Reconfigurable Manufacturing Systems (RMS). With their modular structure, they can be integrated in a short period of time. Though this leaves more time for product development, it does not exclude the industrialization risks. Since configuration of equipment only works reliably if its process technology is well understood, it is needed that poorly functioning manufacturing processes are detected and addressed in an early stage. Only then, sufficient time is available for corrective actions to be taken. This paper presents a scientific framework to model the development of RMS. The method has the capability to uncover manufacturing risks during early development. In combination with RMS, the freeze of system architecture can indeed be pushed backwards in time. The method uses the SADT. The process risks, as outcome of the analysis process, are ranked using a FMEA to determine the severity of their impact.

Researchers, Bucur C. & al. [28], have presented a case study for shaft design made for machine tools using four Integrated Design Systems. At the beginning the SADT diagram used in CAD/CAM Industry and data transfer was elaborated. Based on these diagrams UML architecture was build. Transposing the UML in C++ Programming language in order to improve the data transfer in a Virtual Enterprise the agent source which can be transformed in an import/export agent by a programmer was obtained, then the users could import the part with historic from one IDS to another.

IV. WORKS TO AUGMENT THE SADT METHOD

The aim of this section is in one hand, to present the extended SADT with respect to timing constraints and formalization and in other hand, a Safe-SADT method for dependability studies.

A. *Extended SADT*

There are many method used for representing the processes and the activities: one of the most known is the SADT method. In fact, this formalism adopts a static modeling of the process which is a chain of activities destined to understand, specify and do organization diagnosis. Furthermore, this formalism doesn't permit simulation for estimation purposes that need the data and temporal introduction.

Researcher Feller A. and Rucker R. [29], has proposed an extended SADT method and has described the need for such a method more than 30 years ago. This extended SADT method has been used in many applications with respect to timing constraints and formalization. One of these applications is a proposal of a gait of a physical and economic performance analysis [20].

Figure 2 shows the representation of the activity using the SADT formalism and its equivalence in the extended SADT method.

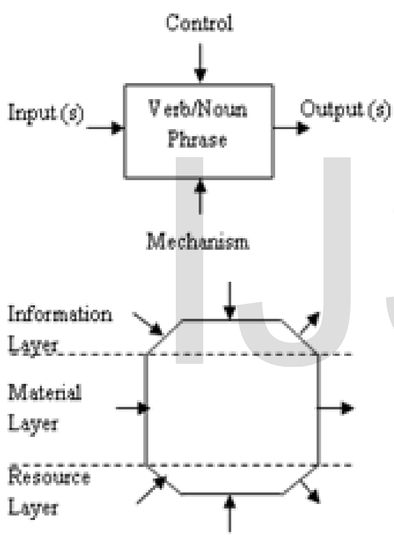

Fig. 2. Activity representation with SADT and its equivalence with the extended SADT.

The extended SADT formalism enables, for each activity, to define precisely its role in the transformation of three different flows: the principal flow (where the activity is to transform an input from an activity upstream of the same process into an output for an activity downstream of the same process, generating value), the flow of resource consumption (s) (where the activity is to transform the ability of the resource that can be requested for other activities, and generate costs corresponding to this capacity consumption), and the secondary activity flow must transform to be executed (for example, a request for intervention activity transforms the state 'launched' to the 'balanced' state, or a flow of lubricant that activity changes to 'clean' to the 'dirty' state).

However, the main adaptation that the Researcher [29] has brought to the extended SADT formalism on the control arc that he thinks it more generally as a secondary input flow not necessarily intended to control the activity, this function can be provided by the trigger arc (Figure 3).

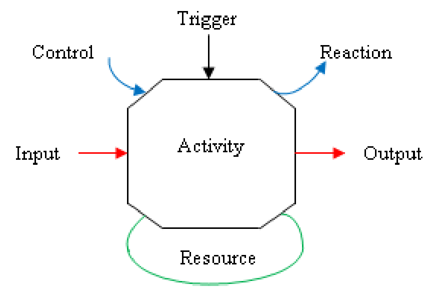

Fig. 3. Different arcs in the extended SADT formalism.

B. *Safe-SADT method*

Dependability evaluation is a fundamental step in automated system design. However, the current dependability evaluation methods are not appropriate given the level of complexity of such systems. Given the ineffectiveness of the current methods, Researcher Bernard V. [30] [31] has proposed the Safe-SADT formalism for dependability evaluation, an extension of the SADT method.

In this section, we present briefly in one hand the Safe-SADT model and in other hand, we show its applicability in industrial fields through two case study. In fact, dependability evaluation is crucial to controlling the risks associated with system failure, and for this reason, it is one of the fundamental steps in automated system design [32] [33] [34].

Indeed, the Safe-SADT approach deals progressively with complexity. Top-down and hierarchical, it focuses on the functions that the system must achieve through function entities and material entities. First, the system is described generally, and then the details are embedded as the analysis progresses. A Safe-SADT model is organized hierarchically [35].

At the top level, the system is summarized with a single global block A0. This block can be broken down at a lower level with more blocks that contain more information on the subsystems. This lower-level decomposition is performed until the parts that make up the overall system are listed (e.g., material entities within the operational architecture, which are specified at the bottom of a Safe-SADT block).

The advantage of this formalism is that it allows the formalization of functional interactions by integrating dependability parameters [30] [31]. The Safe-SADT approach provides a block representation to graphically define complex systems in terms of functional requirement specifications (FRS).

Thus, the formalism allows complex systems to be described in terms of systems, subsystems, and the relationships between subsystems. Each system decomposition is defined with a Safe-SADT block with the objectives of clearly specifying the input functions, output functions, and material entities executing the input functions under some constraints (Figure 4).

ISSN 2229-5518

Figure 4 shows the representation of the Safe-SADT block (Level A0).

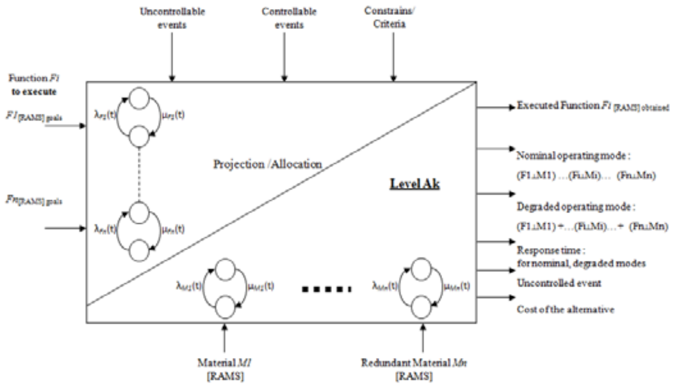

Fig. 4. Safe-SADT formalism.

By means of significant example, Researchers, Cauffriez L. & al. [36], have presented a study on the use of field buses combined with intelligent sensors and actuators which are opening up new possibilities for building control systems. If field buses seem to be a good solution to improve the dependability, it could be also a trap due to the new possible failures they may introduce. They have studied these failures and their effects on dependability parameters. Some elements are presented in order to provide designers with means to assess dependability at each design step by integrating field feedback. Assessing dependability is too often limited to an evaluation at the end of the design process, which often involves reselecting previous choices. To sum up, this contribution constitutes a structured overview of field bus faults given to help users to select the most suitable field bus for their applications, both in control and measurement.

Researchers, Cauffriez L. & al. [37], have presented a computer-aided design tool software for modeling and comparing several architecture design choices early on in the design process. Its originality is based on operational architecture composed of function entities executed by material entities. A Monte Carlo approach allows simulation of ‘‘possible life history’’ and points out design's weaknesses using sensitivity analysis. The researchers have illustrated the tool functionalities with a temperature system. Possibilities for future research in terms of software development and industrial applications are provided.

C. Domain Modelling SADT

In this section, we present the augmentation approach proposed by the researcher Marca D. [38] [39] [40] which has explained how SADT domain modeling can bring correct and complete context, to today's software development methods. In fact, SADT has over 35 years of domain modeling experience, across a vast number of problems involving systems ranging from tiny to huge, in a wide variety of industries.

Indeed, SADT is a proven way to model any kind of domain. Its power and rigor come from: (1) a synthesis of graphics, natural language, hierarchical decomposition, and relative context coding, (2) distinguishing controls from transformations, (3) function activation rules, and (4) heuristics for managing model complexity [40.

Furthermore, domain modeling is at the core of SADT, and when properly used, the method can produces holistic domain models that can address any level of complexity or abstraction. Thus, SADT can produce a set of very concise, small models, with tightly connected context and content.

Figure 5 shows where SADT augments software development methods such as Unified Modeling Language (UML), Agile and Usability Engineering Methods.

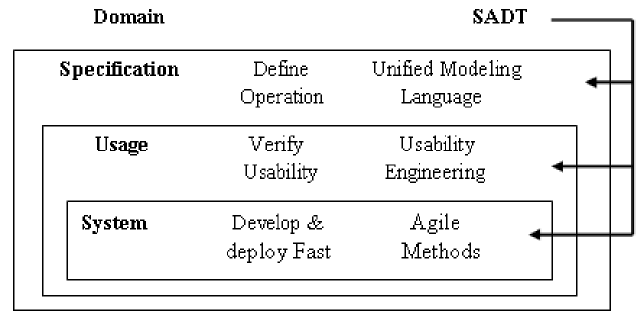

Fig. 5. Augmentation approach for software development methods [40].

The distinguishing, unique aspect of SADT is its ability to holistically describe an entire domain to any desired low level of detail, and to describe its context to any desired high level of abstraction. It is thus, SADT has an extremely simple graphic language and a model creation technique that, from the same starting point of any particular subject, can describe: (1) all details (i.e. decompose complexity), (2) the context of that subject (i.e. context modeling).

Since context preservation is crucial for domain modeling, SADT has merit for augmenting the software development methods. In fact, three of its core features are: context, model, and viewpoint. Not only can SADT correctly, comprehensively and consistently describe an entire domain and not just the immediate context of a software system, it can describe that domain in rich and varied ways using carefully designed in-context supplements.

V. Conclusion

Conceptual Modeling (CM) has gained a lot of attention in the recent years and it is generally agreed that CM is the most important phase of simulation study. Although its significance, there are very many techniques that can assist to develop well-structured and concise conceptual models.

In this paper, we have presented a research on the different works done to augment the SADT method into the extended SADT method and the Safe-SADT method. Then, we presented an augmentation approach to software development methods by using SADT.

In fact, SADT has been extended with respect to timing constraints and formalization. This new formalism enables to define three different flows: the principal activity flow, the flow of resource consumption and the secondary activity flow.

Furthermore dependability evaluation is a fundamental step in automated system design. Thus, given the ineffectiveness of the available methods, the Safe-SADT method of dependability evaluation was presented.

Moreover SADT method has been invented for general purpose domain modeling. In fact, a set of SADT diagrams and supplements that correctly and completely describe the domain. So, an approach has been taken to providing benefits to software development methods by using SADT.

REFERENCES

[1] D.T. Ross, Structured Analysis (SA): A language for communicating ideas, IEEE Transaction On Software Engineering, 3(1), 1977, pp. 16-34.

[2] D.T. Ross, Applications and Extensions of SADT, IEEE Computer, 18(4), 1985, pp. 25–34.

[3] P. Jaulent, SADT un langage pour communiquer, IGL Technology, Eyrolles, Paris, France, 1989.

[4] P. Jaulent, Génie logiciel les méthodes : SADT, SA, E-A, SA-RT, SYS-P-O, OOD, HOOD, Armand Colin, Paris, France, 1992.

[5] M. Lissandre, Maîtriser SADT, Editions Armand Colin, Paris, France, 1990.

[6] D. Marca and C.L. McGowan, SADT: structured analysis and design technique, New York: McGraw-Hill Book, 1988.

[7] K. Schoman and D.T. Ross, Structured analysis for requirements definition, IEEE Transaction on Software Engineering, 3(1),1977, pp. 6-15.

[8] IEEE 1320.1-1998, IEEE Standard for Functional Modeling Language-Syntax and Semantics for IDEF0, IEEE, 1998.

[9] M.N. Lakhoua, Systemic analysis of a production environment to implement an information system, Thesis, ENIT, Tunisia, 2008.

[10] T. Ben Jouida, Contribution à la mise en œuvre d'une gestion stratégique de la production par l'analyse des postes de travail selon une démarche systémique, Thèse de doctorat, ENIT, Tunisie, 2008.

[11] M.K. Jbira, Modeling and evaluation of the performance of the production system of the MTK, Thesis, ESSTT, Tunisia, 1997.

[12] M. Souissi, Modelling and control of agricultural greenhouse climate, Thesis in physical Sciences, FST, Tunisia, 2002.

[13] L . Nabli, Monitoring Preventive conditional estimated indirect to a unit of Filiature Textile: the quality approach, Thesis, USTL, Lille, France, 2000.

[14] H.J Dong, Design of a Syntax Validation Tool for Requirements Analysis Using Structured Analysis and Design Technique, vol.5, 1990.

[15] O. Sénéchal, Pilotage des systèmes de production vers la performance globale, Mémoire d'habilitation à diriger des recherches, LAMIH, Université de Valenciennes, 2004.

[16] M. Lauras, J. Lamothe, and H. Pingaud, Une méthode orientée processus pour le pilotage par la performance des systèmes industriels, Journal Européen des Systèmes Automatisés, 41(1), 2007, pp.71-100.

[17] O.V. Lezina and A.V. Akhterov, Designing of the information component of pedagogical knowledge management system in a chair of technical university, International Conference on Interactive Collaborative Learning (ICL), 2013, pp. 544-546.

[18] C. Yulian, L. Wenfeng, S. Wei and W.A. Chaovalitwongse, Collaborative material and production tracking in toy manufacturing, 17th International Conference on Computer Supported Cooperative Work in Design IEEE CSCWD, 2013, pp. 645- 650.

[19] A. Demri, A.Charki, F. Guerin and H. Christofol, Functional and dysfunctional analysis of a mechatronic system, Annual Reliability and Maintainability Symposium RAMS, 2008, pp. 114-119.

[20] R. Plateaux , J.Y. Choley , O. Penas and A. Riviere, Towards an integrated mechatronic design process, IEEE International Conference on Mechatronics ICM, 2009, pp. 1-6.

[21] T. Wenan, Z. Guosheng, J. Chuanqun and C. Xiaoxia, Knowledge management mechanism and system supporting overall e-learning lifecycle process, IEEE International Conference on Systems, Man, and Cybernetics, 2011, pp. 1897-1901.

[22] M. Goldshtein and A. Matveyev, The informal modeling of input-preprocessing subsystems for scientific research, IEEE 10th Digital Signal Processing Workshop and the 2nd Signal Processing Education Workshop,2002, pp. 140- 144.

[23] E. Ruzaev, E.Tsapko and V. Borikov, The project of system management quality for education process for technical university using SADT-technology The 4th Korea-Russia International Symposium on Science and Technology, vol. 2, 2000, pp. 60-63.

[24] F. Ahmed, S. Robinson, A.A. Tako, Using the structred analysis and design technique (SADT) in simulation conceptual modeling Simulation Conference (WSC), 2014 Winter Year: 2014, pp.: 1038-1049.

[25] R. Yahmadi, K. Brik and F. Ben Ammar, Degradation analysis of the lead acid battery plates in the manufacturing process 12th International Multi-Conference on Systems, Signals & Devices SSD, 2015, pp. 1 - 6

[26] Y. Zennir and R. Bendib, Modeling and dependability analysis of an industrial plant: Case study International Conference on Industrial Engineering and Systems Management IESM, 2015, pp. 1012-1018.

[27] E. Puik, D. Telgen, L. Moergestel and D. Ceglarek, Structured Analysis of Reconfigurable Manufacturing Systems, A. Azevedo (ed.), Advances in Sustainable and Competitive Manufacturing Systems, Lecture Notes in Mechanical Engineering, Springer International Publishing Switzerland, 2013.

[28] C. Bucur, T. Aurite, CAD - CAM integration optimizing data transfer between different integrated design systems in virtual enterprise anvironment, U.P.B. Sci. Bull., Series C, Vol. 72, Iss. 3, 2010.

[29] A. Feller and R. Rucker, Extending Structured Analysis Modelling with A.I.: An Application to MRPII Profiles and SFC Data Communications Requirements Specifications, IFIPS Conference, 1989.

[30] V. Benard, Evaluation de la sûreté de fonctionnement des systèmes complexes, basée sur un modèle fonctionnel dynamique : la méthode SAFE-SADT, PhD. Thesis, Valenciennes University, LAMIH UMR CNRS 8530, 2004.

[31] V. Benard, L. Cauffriez and D. Renaux, The Safe-SADT method for aiding designers to choose and improve dependable architectures for complex automated systems, Reliability Engineering & System Safety, 93(2), 2008, pp. 179-196.

[32] L. Cauffriez, Method and model for the dependability assessment of complex automated Systems: Application to production line performances, application to the design of intelligent distributed systems, Habilitation, LAMIH, Valenciennes University, France, 2005.

[33] L. Cauffriez, V. Benard, D. Renaux, A new formalism for designing and specifying RAMS parameters for complex distributed control systems: the Safe-SADT formalism, IEEE Transactions on Reliability, Vol. 55/3, 2006, pp. 397-410.

[34] L. Cauffriez, J. Beugin, D. Renaux and P. Millot, Design of Urban guided transport management system: a dependability point of view, Deliverable D6, safety conceptual approach & guidelines, 5th Framework Programme, 2005, pp. 70–75.

[35] L. Cauffriez, J. Cicottelli, B. Conrard, and M. Bayart, Design of Intelligent Distributed Control Systems: A Dependability Point of View, Reliability Engineering and System Safety 84, N°1, 2004, pp. 19-32.

[36] L. Cauffriez, B. Conrard, J-M Thiriet, M. Bayart, Fieldbuses and their influence on dependability, Instrumentation and Measurement Technology Conference IEEE IMTC, Vail, CO, USA, 2003.

[37] L. Cauffriez, D. Renaux, T.Bonte and E.Cocquebert, Systemic modeling of integrated systems for decision making early on in the design process, Cybernetics and Systems: An International Journal, 44, 2013, pp. 1-22.

[38] D. Marca, Augmenting SADT to develop computer support for cooperative work, Proceedings 13th International Conference on Software Engineering, 1991.

[39] D. Marca, IDEF0 and SADT: A Modeler's Guide. Boston: OpenProcess, Inc., 3rd edition, 2006.

[40] D. Marca, SADT/IDEF0 for Augmenting UML, Agile and Usability Engineering Methods, Software and Data Technologies, 2012, pp. 38-55.