Proceedings of the 2014 Winter Simulation Conference
A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, eds.

USING THE STRUCTRED ANALYSIS AND DESIGN TECHNIQUE (SADT) IN SIMULATION CONCEPTUAL MODELING

Fahim Ahmed
Stewart Robinson
Antuela A. Tako

School of Business and Economics
Loughborough University
Loughborough
Leicestershire, LE11 3TU , UNITED KINGDOM

ABSTRACT

Conceptual Modeling (CM) has gained a lot of interest in the recent years and it is widely agreed that CM is the most important phase of simulation study. Despite its significance, there are very few techniques that can help to develop well-structured and concise conceptual models. This paper proposes the use of the Structured Analysis and Design Technique (SADT) from software engineering to develop conceptual models. SADT has proven to be successful in the development of software systems, specifically in the requirements gathering phase. This paper contributes to the area of 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.

1 INTRODUCTION

Conceptual modeling (CM) can be divided into two main phases: 1) generation of the system description (SD) from the real world within the problem domain and 2) abstraction of the conceptual model from the SD (Kotiadis and Robinson 2008). This paper focuses on the first phase of CM. The domain of interest is primarily Discrete Event Simulation (DES). In the DES literature there are not many structured approaches to guide the modeler in generating a SD. There are number of interesting works that discuss the CM framework and provide effective guidelines on how to perform different phases of CM (Kotiadis, Tako, and Vasilakis 2013; Van der Zee and Van der Vorst 2007; Montevechi and Friend 2012; Balci and Ormsby 2007; Robinson 2008). However, most of this work is either domain-specific (healthcare, military, engineering) or descriptive in nature, and so lack a precise structure for creating conceptual models. Also, in all of these studies, there are discussion for activities, actions, objectives, communications, contents, inputs and outputs, but there is little attention on explicitly structuring different components of the SD (entities, resources, controls and activities) in a DES study. Such an approach requires a thorough investigation of these key individual components and interactions among them from the beginning of the CM process. If these details are put together in a structured and communicative form, then it can minimize the efforts required in the abstraction and design phase. We use the Structured Analysis and Design Technique (SADT) from software engineering for this purpose. Its generic style and notations makes it suitable for use in various problem domains. We believe that software engineering methods have potential benefits to be used for CM within the Modeling and

Simulation (M&S) discipline due to the analogy between the two fields (Arthur and Nance 2007; Ahmed, Robinson, and Tako 2014).

This paper is structured as follows. Section 2 presents a brief background of SADT, followed by a discussion of the rationale for using SADT in CM. we then present the proposed framework with the help of case study on Panorama Televisions production taken from Robinson (2014). The paper concludes with some discussion and future work in section 5.

2 STRUCTRED ANALYSIS AND DESIGN TECHNIQUE (SADT)

2.1 Background

SADT has a long history within software engineering. It was developed by Ross (1977) as a result of ongoing work (1969-1973) in problem solving dating back in 1950s at Softech. SADT is a graphical language and was used extensively for describing complex systems in communicative designs, military planning and computer-aided manufacturing (Dickover, Mcgowan, and Ross 1977). SADT was successfully applied in problem analysis and functional specifications; however, it has been used most effectively in the requirements definition phase for software design (Ross and Schoman 1977; Ross 1985). SADT was adopted as Icam DEFinition for Function Modeling (IDEF0) by US Air Force Integrated Computer Aided Manufacturing (ICAM) in 1980s. It was part of ICAM's program to fulfill the need of powerful modeling methods for system analysis and design. Several other IDEF packages were introduced during that period; mainly, IDEF1, IDEF2, IDEF3, and IDEF1x. These are not discussed in this paper, but details can be found in Menzel and Mayer (1998). Since our focus is mainly on the root definitions and labels developed in SADT, we prefer to use SADT instead of IDEF0. IDEF0 is mentioned, however, when any related work is addressed.

2.2 SADT Notation and Style

SADT notations consist of box-arrow diagrams (blocks), with four arrows on each side defined as: input, output, control and mechanism and one activity in the middle as shown in figure 1. Their definitions consist of the following:

Activity: An activity is any function or process that serves to transform inputs into outputs
Input: The data/information required by an activity to start the transformation process
Output: the data/information produced by the activity as a result of this transformation
Control: Any constraint that affects the behavior of activity in some way
Mechanism: Persons, resources, or any means that are required to run the activity

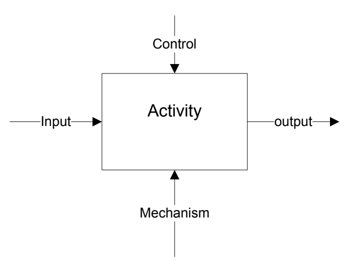

Figure 1: General notation of SADT.

SADT uses the traditional "top-down" approach for the hierarchical modeling of functions. The process starts with specifying the highest simplified level of detail in the top-level diagram which is then decomposed into further details at each step until the required level of detail has been reached. This approach is shown in figure 2. In the first stage, a top-level diagram A is created (top figure 2). This diagram simply states that, the input I_0 is transformed to output O_0 by means of an activity A being constrained by control C_0, and using some resource R_0. Usually C_0 and R_0 are group notations which indicate that this particular type of constraint or resource would be needed to complete this transformation. In the next stage, A is decomposed into further sub-activities, showing a more detailed lower-level view of the process (bottom of figure 2). Each stage now shows individual inputs, outputs, controls and resources; however, they are working discretely to produce the same transformation (I_0 into O_0) as in the original high-level diagram.

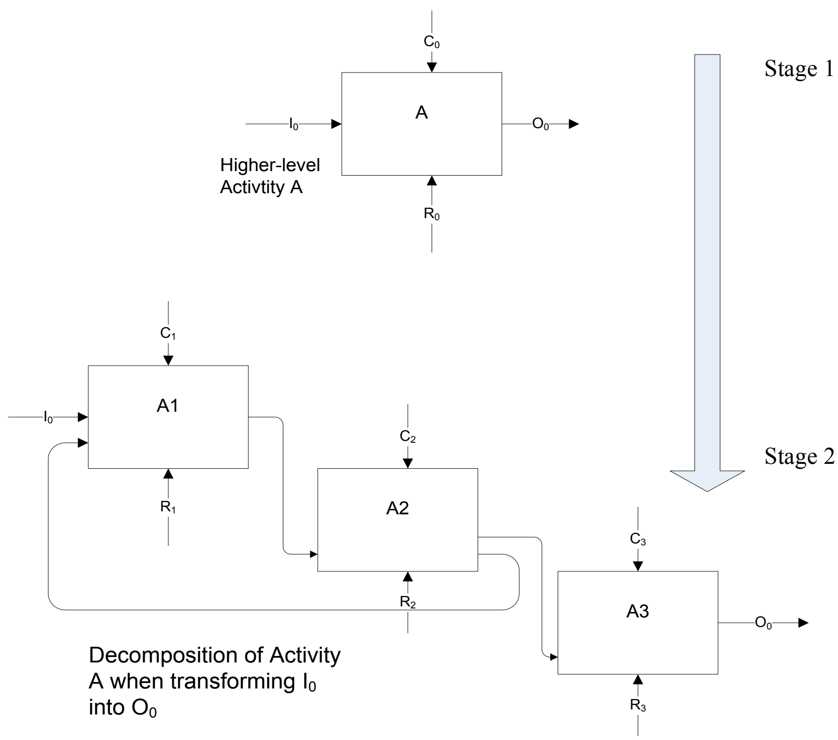

Figure 2: Hierarchical decomposition of SADT in "Top-down" style.

3 SADT CM

This section presents the rationale for using SADT in CM. First, related work is presented, followed by the proposed methodology.

3.1 Related work

There is not much evidence on the use of SADT within M&S, and more specifically in the CM phase. Some studies, however, indicate that SADT is a potential candidate for use in modeling service operations and queuing systems. For instance, Santarek and Buseif (1998) propose the use of SADT in creating high level system design specification for flexible manufacturing systems. Van Rensburg and Zwemstra (1995) demonstrate the process of developing simulation models of air traffic controller positions using IDEF0 and IDEF3 hierarchal modeling. Congram and Epelman (1995) explains a procedure to describe a service operation (individual tax-return) using SADT diagrams at each stage. Kim et al. (2003) discusses the integration of IDEF models with UML models in order to bring richer semantics in enterprise modeling. Whitman, Huff, and Presley (1997) describes a procedure for converting static models (built in IDEF0/IDEF3) to simulation models using some rules and model annotation. Jeong (2000) uses IDEF0 and IDEF3 for simulating optimization of scheduling systems. Some efforts have been made towards the use of IDEF0 in creating a standard documentation for modeling methodologies (Nathan and Wood 1991). However, all these studies do not focus explicitly on CM.

Some recent work (Montevechi et al. 2008; Montevechi et al. 2010) has proposed the use of IDEF0 combined with other techniques (SIPOC/flowchart/IDEF3) for CM. Montevechi et al. assert that IDEF0 alone can not capture enough details for DES CM study. Meanwhile, in the Eurpoean research project "PROTOCURE II" (Lucas et al. 2005) an extensive review of 70 modelling langauges is presented for the purpose of selecting a languge for the medical guideline development process. The selection is carried out on the basis of requirements and features desirable for a process modelling langauge, and takes into account whether the selected langauge is *living* in terms of references, documentation and known history of origin. Lucas et al. selected IDEF0 as the main process modeling language in preference to some other well-known modeling languages, for instance, E3, GRAI and UML2.

3.2 Methodology

Since we believe there is a tendency to over elaborate the SD, we propose an approach to CM using SADT with little modification. The proposed method creates SADT CM branches (activity based models) within the style and notation explained later on in this section. Each individual activity is screened and details of various components (entities, resources and controls) are added. The reason for selecting "Activities" as the starting point of the process is that they are central to DES. All other components are dependent on the individual activities in the system. Any input/output activities are also identified for the "Activity". This allows for modeling any one "Activity" in a system comprehensively along with all dynamics at a lower level.

Due to the nature of DES, the original SADT notations and style have been modified to suit a DES modeling context. These are shown in figure 3. The box represents any one activity and the input/output arrows represent the preceding/following activity. The resources and control arrows represent the same components as in SADT. The output activity arrow includes an entity label indicating the flow of the entity on this route. This is an additional label to the original SADT activity block.

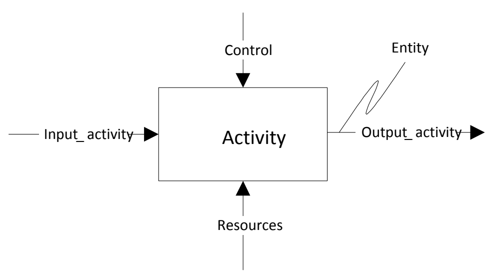

Figure 3: Proposed notation of SADT CM.

The definitions of the labels used are as follows (figure 3):

Activity: This corresponds to the activity under consideration (transformation activity). This is the central part of the analysis. Therefore, it is important to identify all activities carefully in the system. There is only *one* transformation activity in a single SADT CM block. The label uses a notation of alpha-numeric style indicating the operation or activity (e.g. OP1, and A1 to indicate operation 1 and activity 1 in the system).

Input activity: this determines all activities preceding the activity. Arrows can be replicated in case of more than one activity.

Output activity: this determines all activities following the activity. Arrows can be replicated in case of more than one activity.

Resources: Any resource (or group of resources) required by the activity is indicated by the bottom arrow of the box. The notation used is "R_x" where "x" represents the ID of a resource (or group of resources). In case of more than one ungrouped resource, "x" can indicate this value as a range for e.g. R_{1-3} means three individual resources (R_1, R_2, and R_3) are required on a particular activity. The values of "x" are unique unless used for shared resources (or group of resources) on different activities. Arrows can be replicated for two completely different types of resources on a single activity, for e.g. the operators and transportation resources can be both represented by R_x and T_x on two different arrows respectively.

Controls: this represents any constraint imposed on the activity by the environment, management, production plans, customer demand etc. A simple notation "C_X" is used, where, "x" denotes the ID number of the constraint. This label will also be the same for activities sharing any similar constraints. Arrows can be replicated for multiple constraints on the same activity.

Entities: this label represents the entities flowing out from the activity and following this route. These are represented by E_x where "x" represents the ID of the entity. In case of batch entities, it is denoted by E_xE_y where "X" and "Y" represents the IDs of two different entities.

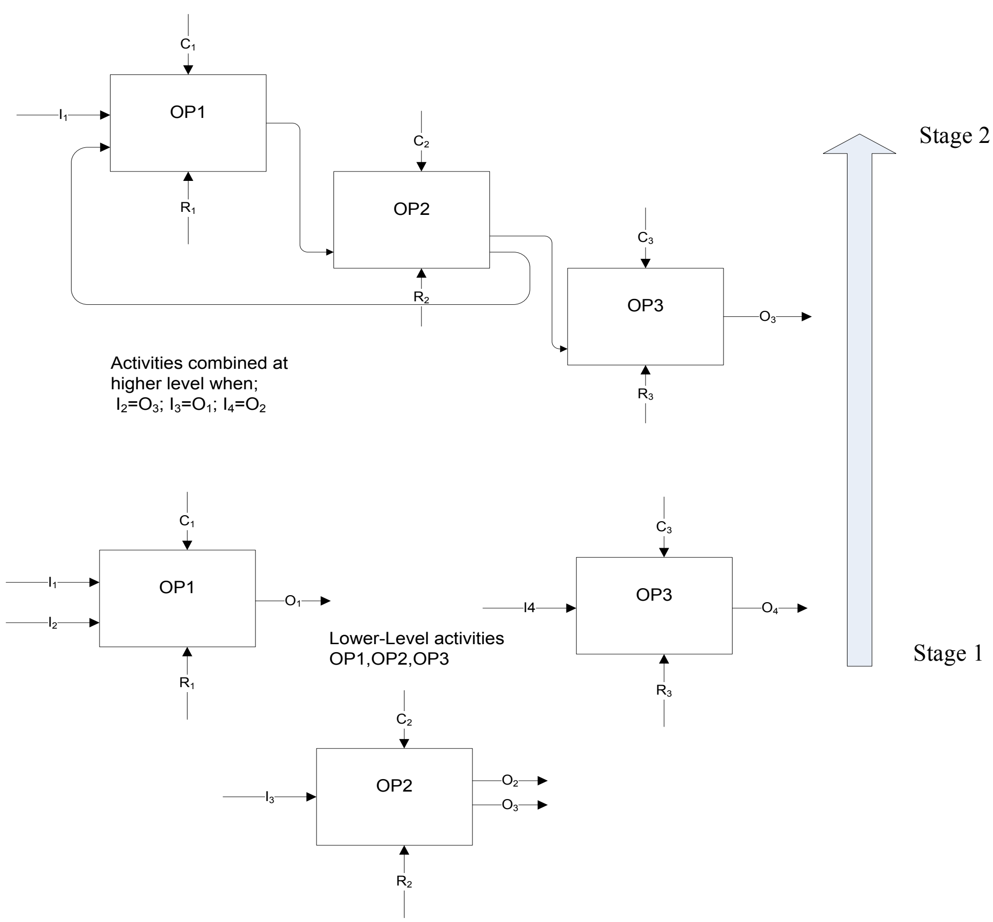

Figure 4: Proposed "Bottom-up" approach for SADT CM.

The next modification proposed is changing the modeling style from the original "top-down" style of SADT to a "bottom-up" approach for SADT CM. This is because we believe that the DES domain is different from that of software systems in this respect. The details in the DES model are present at the lowest individual decomposition level rather than the highest level of information as in enterprise wide modeling (Whitman, Huff, Presley 1997). All individual components modeled precisely at their places, help to define statics (positions) and dynamics (interactions) of the system more efficiently. These atomic models can then be connected in a "Bottom-up" style to represent the complete system. This would also maintain the consistency in representing the dynamics of the components at all levels of modeling. This approach has also been used in component based or hierarchal modeling in DES (Cetinkaya, Verbraeck, and Seck 2009; Sargent et al. 1993; Nance 1979). This style of modeling is the basis for creating local *agents* in Agent Based Modeling (ABM). *The agents' behavior is* specified at an individual scale and then the global behavior of the system emerges at a higher level as a result of these local interactions (Batty, Desyllas, and Duxbury 2003; Borshchev and Filippov 2004). Another benefit of this approach is that the individual modeling units can be studied at various locations (distributed modeling) which can then be combined at a central server to get the complete system information. This individual study can be carried out at lower levels even when knowledge about the complete system is unknown. Some interesting work in this area is from Pratt, Mize, and Kamath (1993).

The approach is shown in figure 4. First, the low-level diagrams are created, where all individual activities are investigated individually after listing all entities, activities and resources (explained in more detail in the next section). During this investigation, all SADT CM *branches* are created with the proposed notation (figure 3) and following the label conventions explained earlier in this section. Once these individual lower-level diagrams (OP1, OP2, OP3) have been created, they are combined together to generate a higher level (but detailed) description of the entire system. This combination follows from the inter-connection of individual input/output activities to represent entity paths and feedback loops. For example, OP1 is fed by two activity sources: an independent activity and an output activity from OP2. It should be noted that in figure 4, input/output labels are for illustration purposes only in order to give an analogy to SADT labels (figure 2). In the proposed framework, these labels represent input/output activities and are labeled similarly to the main activity. We next explain the framework for SADT CM using Panorama Televisions production plant case study (Robinson, 2014). A complete problem description for Panorama Televisions production plant is included in Appendix A.

Table 1: List of activities/resources/controls in Panorama Televisions production plant.

Activity	Label	Resource	Label	Control	Label
TV arrives into system	Sc(OP9)	Forklift truck	T_1	Production Schedule	C_1
Boxes loaded to pallet	OP10	Shift operator	R_1	n/a	
LCD assembly	OP20	Maintenance operator	M_1	n/a	
Circuit Board Assembly	OP30	Maintenance operator	M_1	n/a	
Electrical Assembly	OP40	Group of shift operators	$R_{2\text{-}5}$	n/a	
Test area one	T1	Maintenance operator	M_1	n/a	
Rework area one	RT1	Shift operator	R_6	n/a	
Back Assembly	OP50	Maintenance operator	M_1	n/a	
Finished TV Unloaded from line	OP60	Shift operator	R_7	n/a	
Test Area Two	T2	Maintenance operator	M_1	n/a	
Rework area two	RT2	Shift operator	R_8	n/a	
Packaging	OP70	Shift operator	R_9	n/a	
Finished TV Transported to Warehouse	OP80	Forklift truck	T_2	n/a	
Product exit form system	Sk(OP90)	n/a	n/a	n/a	

Where, "Sc" represents the entry point of entities into the system (source), "Sk" represents the exit point of entities from the system (sink)

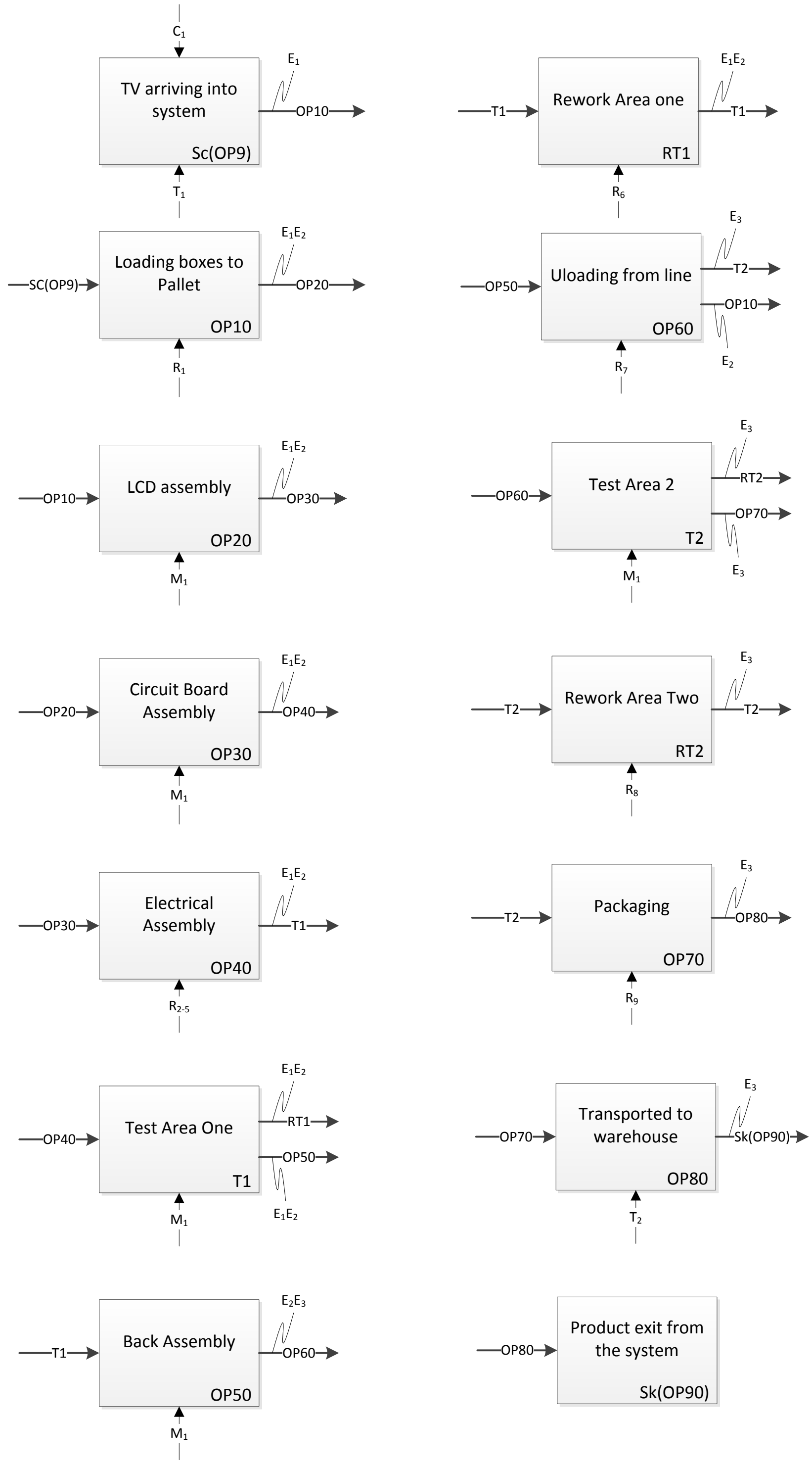

Figure 4: SADT CM branches for the Panorama Televisions production plant.

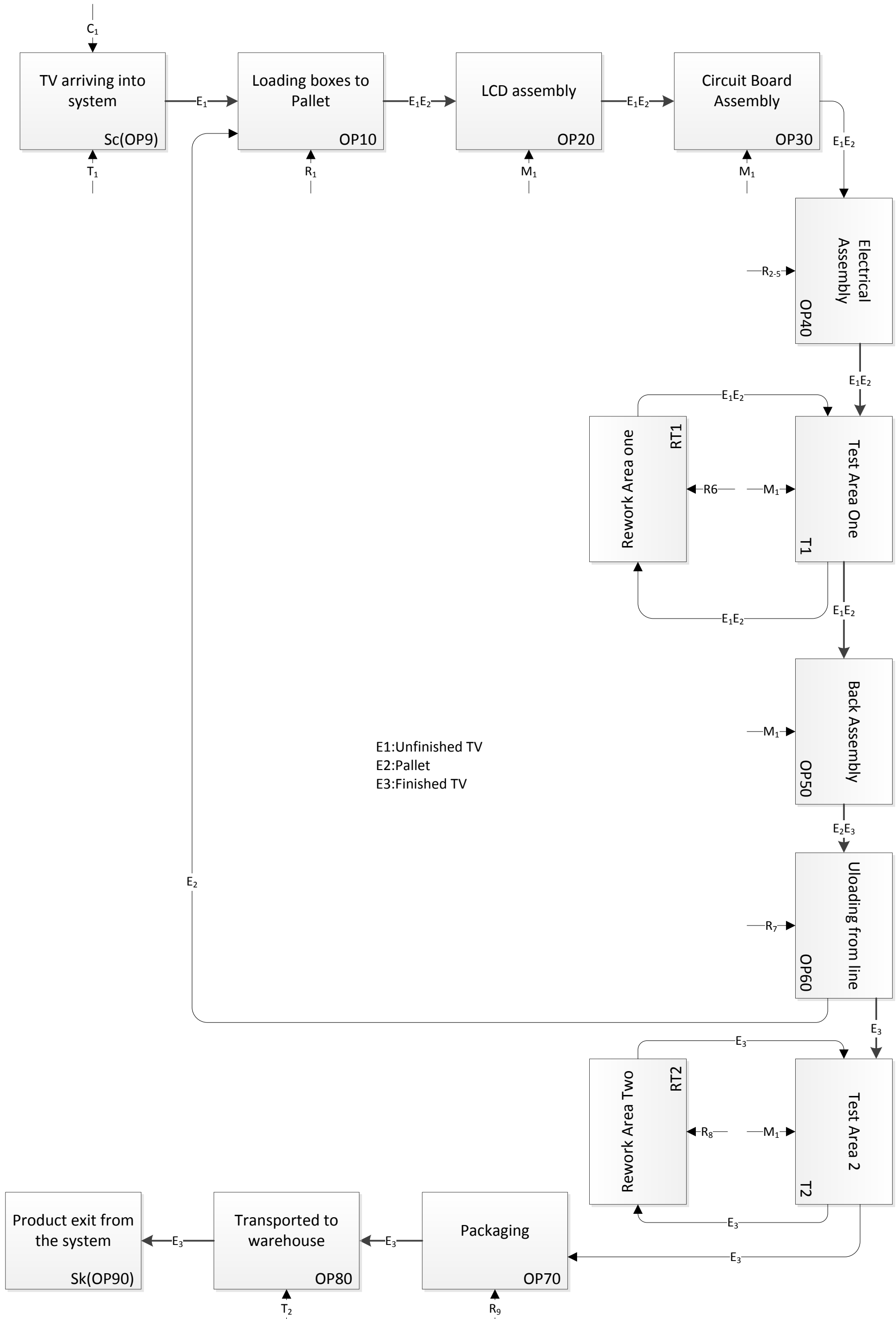

Figure 5: Complete SD diagram for Panorama Televisions production plant.

4 THE PROPOSED SADT CM FRAMEWORK

The framework for SADT CM consists of the following four steps:

1. Identification of entities
2. Identification of all activities (list any resource and constraint requirements related to each activity)
3. Draw SADT branches for each activity as explained in section 2.3.2
4. Generate the complete activity diagram by combining all SADT branches.

We next describe each step with the help of the Panorama Televisions case.

4.1 Step 1: Identification of entities

The first step is to identify all entities in the system. These include any type of raw material, product, moving objects, people, cars, unfinished products, pallets, fixtures etc. All entities are listed in a table allocating a unique label name to each entity. The following entities have been identified for the Panorama Televisions case:

- Unfinished TVs (E_1)
- Pallet (E_2)
- Finished TVs (E_3)

4.2 Step2: Identification of Activities (resources/controls)

In this step, all activities are defined and labels are assigned to them. Any resource or control related to each activity is also identified and labeled. Table 1 lists this data for the Panorama Televisions case.

4.3 Step 3: Drawing SADT branches

In this step, the SADT branches for each activity are created. The branches created for the Panorama Televisions case are shown in figure 4.

4.4 Step 4: Developing the complete SADT diagram

In this step, all branches created in step 3 are combined together to generate the complete SD. Figure 5 provides the complete SD for the Panorama Televisions production plant. The entities are marked on the output activity arrows in the complete diagram as shown in figure 5.

5 DISCUSSION AND FUTURE WORK

The paper presents a structured approach for CM using the Structured Analysis and Design Technique (SADT) from software engineering. The framework is tested using the Panorama Televisions case. We show that SADT has a potential for use in DES in order to structure a problem by defining all its components (entities, resources and controls) at a lower-level of modeling. This helps to identify individual interactions among these components along with any resource or constraint requirement. These atomic models can be created using SADT by various modelers and at different places. The generic notations and communicative style of SADT allows this information to be integrated into a complete system description (SD). The next phase of CM, the abstraction of the conceptual model from the defined SD, is being considered. It is, however, anticipated that the structured SD defined using the SADT approach, would allow an easier abstraction and design of simulation models. In addition, CM validation could also be carried out in a more efficient and transparent way. Direct comparison of the SADT SD diagrams with the SADT CM diagrams would enable the abstractions in the model to be identified, debated and validated with subject matter experts.

One of the limitations of SADT CM approach is handling the systems with a large number of activities and resources where the combination of individual models can prove to be cumbersome. This would require the use of an automated support tool where all individual models can be fed into it and the automatic generation of SD would be possible. This could be achieved by writing a computer program in object-oriented language (for example, JAVA or C++), using activities as objects under different classes (for example, main line, testing and packaging class). A user input would be required to indicate different parameters for each object (input/output activity, resource and control). In the final step, using the graphical library of the language, different objects (activities) under different classes can be combined to produce a complete SD in a diagrammatic form. However, in the absence of such a tool, the manual approach of SADT CM would still be useful because the individual models can be created manually without having a complete knowledge of the system.

A APPENDIX: PANORAMA TELEVISIONS CASE (SOURCE: ROBINSON, 2014)

Panorama Televisions have been involved in the manufacturing of electrical goods since the early days of the radio. They now concentrate on the production of high quality, premium priced televisions for the international market. There are four televisions in their product range: small, medium, large and internet enabled. Last year, to meet increased demand, Panorama invested in a new television assembly plant. Also, after some negotiation with the unions, all areas of the site moved to continuous working over a five day week. However, the plant has never achieved its target throughput of 500 units per day. In fact, daily throughput is only just over 400 units. The assembly plant processes are described as follows. Plastic molded boxes are loaded to a pallet by an operator at OP10. A production schedule, which is based on projected demand, determines the type of box to be loaded (small, medium, large or for internet enabled). At OP20 the LCD is assembled to the box before the circuit board is added at OP30. The televisions travel on a conveyor and five manual operators assemble the electrical equipment, OP40. The television is then tested and any failures go to the rework area. Good televisions have the back assembled at OP50 and are unloaded from the line at OP60 by an operator. The empty pallets are returned by conveyor to OP10 and the televisions are stored on a circular sling conveyor. A television is taken from the conveyor when a final test booth becomes available. Televisions failing this test are sent for final rework. Televisions passing are stored on another sling conveyor and are packed at OP70. Packed televisions are transported to the warehouse by forklift truck. The final test and packing area are often short of work and there is enough spare capacity to achieve 500 units per day. The management at Panorama believes that the throughput problem is a result of the number of stoppages on the main assembly line. There are a significant number of breakdowns on automated machines, and set-ups are required every time there is a change of product in the production schedule. There is only one maintenance engineer per shift who is required to attend all machine breakdowns and set-ups. There seems to be little opportunity to improve the efficiency of the machines, nor can the production schedule be changed since it is driven by customer demand. The solution being considered is to increase the buffering between the operations to dampen the effects of stoppages. Design engineers have considered this proposal and believe that, due to physical constraints on space, the buffering could be increased by a maximum of 200%. This will also require further pallets to be bought. In fact, there is some uncertainty as to whether enough pallets are currently being used. Increasing the number of pallets may provide a solution without the need for further storage. Extra storage is expensive, so before investing Panorama want to be sure it is necessary. Also, special pallets have to be used at a cost of $1,000 each, so it is important to minimize the number required. Target throughput must be achieved, but expenditure should be kept to a minimum. The management at Panorama is looking for some proposals on how to improve the line within ten working days.

REFERENCES

Ahmed, F., Robinson, S. & Tako, A. 2014. "Conceptual Modelling: Lessons from Computer Science". In: B.T Jahjono, C. H., S. Onggo, and D-J. Van Der Zee (eds.) *7th Simulation Workshop Conference (SW14).* The Abbey Hotel Golf and Country Club, Worcestershire,England B98 9BE. UK.

Arthur, J. D. & Nance, R. E. 2007. "Investigating the use of Software Requirements Engineering techniques in Simulation Modelling". *Journal of Simulation,* 1**,** 159-174.

Balci, O. & Ormsby, W. F. 2007. "Conceptual Modelling for Designing Large-Scale Simulations. *Journal of Simulation,* 1**,** 175-186.

Batty, M., Desyllas, J. & Duxbury, E. 2003. "The Discrete Dynamics of Small-Scale Spatial Events: Agent-based Models of Mobility in Carnivals and Street Parades". *International Journal of Geographical Information Science,* 17**,** 673-697.

Borshev, A. & Filippov, A. 2004. "From System Dynamics and Discrete Event to Practical Agent Based Modeling: Reasons, Techniques, Tools". *Proceedings of the 22nd International Conference of the System Dynamics Society*.

Cetinkaya, D., Verbraeck, A. & Seck, M. 2009. "Towards A Component Based Conceptual Modeling Language for Discrete Event Simulation". http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.403.8115 (accessed March 2014)

Congram, C. & Epelman, M. 1995. "How to Describe Your Service: An Invitation to The Structured Analysis and Design Technique". *International Journal of Service Industry Management,* 6**,** 6-23.

Dickover, M. E., Mcgowan, C. L. & Ross, D. T. 1977. "Software Design using: SADT" *Proceedings of the 1977 annual conference.* ACM.

Jeong, K.-Y. 2000. "Conceptual Frame for Development of Optimized Simulation-based Scheduling Systems". *Expert Systems with Applications*, 18, 299-306

Kim, C.-H., Weston, R. H., Hodgson, A. & Lee, K.-H. 2003. "The Complementary use of IDEF and UML Modelling Approaches". *Computers in Industry,* 50**,** 35-56.

Kotiadis, K. & Robinson, S. 2008. "Conceptual Modelling: Knowledge Acquisition and Model Abstraction". *Proceedings of the 2008 Winter Simulation Conference.* S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, J. W. Fowler, eds. 951 – 958, Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Kotiadis, K., Tako, A. A. & Vasilakis, C. 2013. "A Participative and Facilitative Conceptual Modelling Framework for Discrete Event Simulation Studies in Healthcare". *Journal Of The Operations Research Society*. 65,197–213.

Lucas P., Hommerson A., Galan J.C., Marcos M., Coltell O., Mouzon O., Polo C., Rosenbrand K., Wittenberg J., van Croonenborg J. 2005. "New model of guideline process". Specific Targeted Research Project Information Society Technology, Universitat Jaume I, IST-FP6-508794.

Menzel, C. & Mayer, R. J. 1998. "The IDEF Family of Languages". *Handbook on architectures of information systems.* Springer.

Montevechi, J., Da Silva Costa, R. F., Leal, F., DE Pinho, A. F., Marins, F., Marins, F. F. & De Jesus, J. T. 2008 . "Combined Use of Modeling Techniques for the Development of the Conceptual Model in Simulation Projects". *Proceedings of the 2008 Winter Simulation Conference* .S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, J. W. Fowler, eds. 987-995, Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Montevechi, J., Leal, F., De Pinho, A. F., Da Silva Costa, R. F., De Oliveira, M. L. M. & Da Silva, A. L. F. 2010. "Conceptual Modeling in Simulation Projects by Mean Adapted IDEF: An Application in a Brazilian Tech Company". *Proceedings of the 2010 Winter Simulation Conference*. B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan,eds. 1624-1635, Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Montevechi, J. A. B. & Friend, J. D. 2012. "Using a Soft Systems Methodology Framework to Guide the Conceptual Modeling Process in Discrete Event Simulation". *Proceedings of the 2012 Winter Simulation Conference.* C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A.M. Uhrmacher, eds. 1 – 12, Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Nance, R. E. 1979. "Model Representation in Discrete Event Simulation: Prospects for Developing Documentation Standards". Academic Press, Inc., New York.

Nathan, B. & Wood, J. 1991. "The Use of IDEF0 to Document a Methodology-a Novices Point of View". *Automation and Robotics Research Institute.* Fort Worth, Texas.

Pratt, D., J. Mize, and M. Kamath, 1993, "A case for bottom-up modeling" . *2nd Industrial Engineering Research Conference Proceedings*, IIE, Norcross. GA, pp. 430-434

Robinson, S. 2008. "Conceptual Modelling for Simulation Part II: A Framework for Conceptual Modelling". *Journal Of the Operation Reserach Society,* 59**,** 291-304.

Robinson, S. 2014. *Simulation: the practice of model development and use*, 2nd *ed*. Palgrave, London.

Ross, D. T. 1977. "Structured Analysis (SA): A Language for Communicating Ideas". *IEEE Transaction on Software Engineering, ,* SE-3**,** 16-34.

Ross, D. T. 1985. "Applications and Extensions of SADT". Computer, vol. 18, no. 4, pp. 25-34, doi:10.1109/MC.1985.1662862.

Ross, D. T. & SCHOMAN, K. E. 1977. "Structured Analysis for Requirements Definition". *IEEE Transaction on Software Engineering,* 3**,** 6-15.

Santarek, K. & Buseif, I. M. 1998. "Modelling and Design of Flexible Manufacturing Systems using SADT and Petri Nets Tools". *Journal of Materials Processing Technology,* 76**,** 212-218.

Sargent, R. G., Mize, J. H., Withers, D. H. & Zeigler, B. P. 1993. "Hierarchical Modeling for Discrete Event Simulation (panel)". *Proceedings of the 1993 Winter Simulation Conference.* G.W. Evans, M. Mollaghasemi, E.C. Russell, W.E. Biles, eds. 569 – 572, Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Van Der Zee, D. J. & Van Der Vorst, J. G. A. J. 2007. "Guiding Principles for Conceptual Model Creation in Manufacturing Simulation". Proceedings of the 2007 Winter Simulation Conference. S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, eds. 776-784, Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Van Rensburg, A. & Zwemstra, N. 1995. "Implementing IDEF Techniques as Simulation Modelling Specifications". *Computers & Industrial Engineering,* 29**,** 467-471.

Whitman, L., Huff, B. & Presley, A. 1997. "Structured Models and Dynamic Systems Analysis: The Integration of the IDEF0/IDEF3 Modeling Methods and Discrete Event Simulation". *Proceedings of the 1995 Winter Simulation Conference.* S.Andradbttir, K. J. Healy, D. H. Withers, and B. L. Nelson , eds. 518 - 524, Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

AUTHOR BIOGRAPHIES

FAHIM AHMED is a PhD student at the School of Business and Economics, Loughborough University. He holds an MSC in Control Engineering from Chalmers University of Technology, Sweden. Previously, he has worked with the project that involves formal verification of Discrete Event Systems at the department of computing and software, McMaster University, Canada. Key areas of interest are simulation modeling for discrete event systems, application of software engineering methods to simulation modeling and cross-disciplinary study between computer science and operational research. His email address is: f.ahmed@lboro.ac.uk.

STEWART ROBINSON is Professor of Management Science and Associate Dean Research at Loughborough University, School of Business and Economics. His research focuses on the practice of simulation model development and use. Key areas of interest are conceptual modeling, model validation, output analysis and alternative simulation methods (discrete-event, system dynamics and agent based). Professor Robinson is co-founder of the Journal of Simulation and President of the Operational Research Society. Home page: www.stewartrobinson.co.uk. Email: s.l.robinson@lboro.ac.uk.

ANTUELA A. TAKO is a Lecturer in Operations Research at the School of Business and Economics, Loughborough University. She holds a PhD in Simulation and an MSc in Management Science and Operational Research from the University of Warwick. Her research interests include the comparison of simulation approaches (discrete-event simulation and system dynamics), facilitated and participative simulation modeling and conceptual modeling. She is an Associate Fellow of the Operational Research Society (AFORS). Home page: www.lboro.ac.uk/departments/sbe/staff/profiles/takoantuela/tako-antuela-anthi.html Email: A.Takou@lboro.ac.uk.