Integrating System Dynamics with Conceptual and Process Modeling

P. Fiona Tulinayo

Institute of Computing and Information Sciences, Radboud University Nijmegen
Heyendaalseweg 135, 6525 AJ Nijmegen, The Netherlands, EU.
F.Tulinayo@science.ru.nl

Abstract. The purpose of this research is to provide a basis on which stakeholders make their decisions, to improve System Dynamics (SD) modeling by deploying methods and techniques from system development, and to support in the creation of SD models where, static and dynamic methods are applied together to achieve a common goal. To achieve this we will take a stepwise approach: identify the key concepts as used in different methods, map their constructs, derive transformations, create their syntax and semantics, and develop requirements specifications on which a tool can be based. For Conceptual modeling we use Object-role modeling (ORM), a fact-oriented approach for modeling information at a conceptual level. For process modeling we use a technique (workflow language) called Yet Another Workflow Language (YAWL). YAWL works as an extension of Petri Nets with constructs to address the multiple instances, advanced synchronization and cancellation patterns. This research is under the supervision of Dr. S.J.B.A (Stijn) Hoppenbrouwers, Dr.(Patrick) van Bommel and Prof. Dr. H.A. (Erik) Proper.

Keywords: *System Dynamics, Conceptual Modeling, Process Modeling*

1 Introduction

Integration of methods involves mapping and defining different concepts with an aim of using them under one umbrella. Paige [17] defines method integration as an involvement in defining relationships between different methods so that they may productively be used together to solve problems. He further gives more definitions inline with method integration in [18]. This study proposes to integrate System Dynamics (SD) with conceptual and process modeling. For Conceptual modeling we intend to use Object-Role Modeling (ORM) [10], a fact-oriented approach for modeling information at a conceptual level [11]. This is because of its strong verbalization, conceptualization and a fully formal link to predicate logic. ORM has graphical constraint notations that are far more expressive than, for example, Unified Modeling Language (UML) class diagrams or industrial Entity Relationships (ER) [11]. Halpin and Wagner further note that ‘*‘although ORM supports modeling of business terms facts, and many static integrity constraints and derivation rules, it cannot model the reactive behavior of systems which can be described using dynamic integrity constraints”*. This explains our use of YAWL

and SD to capture the dynamic part of the business process.
For process modeling we will use a technique (workflow language) called Yet Another Workflow language (YAWL) [26] because its one of the general representatives of process modeling. YAWL works as an extension of Petri Nets with constructs to address the multiple instances, advanced synchronization, and cancelation patterns. We will use Conceptual Modeling (ORM) and Process Modeling (YAWL plus Petri Nets) in the service of creating an integrated model.
We carry out this integration because it is hard to define complex dynamic models in complex organizational settings therefore, we need support based on ontology (conceptual structure). Secondly, for transferability purposes that's in-cases where information from one organization need to be reused by another. lastly to be able to have a basis for the development of a tool that will aid in understanding model behavior.

Figure.1 illustrates how we integrate the methods (SD, Conceptual (ORM)and Process (YAWL plus Petri Nets) modeling). Two types of mappings are shown: mappings between viewpoints are what we refer to as inter-viewpoint mappings, and the Mappings between specific viewpoints and the integrated meta model are refereed to as viewpoint meta model mappings. We use ORM, which is a state and event reporting, fact oriented modeling technique, as a graphical representation for the integrated meta-model. Petri nets Plus YAWL are used to model a discrete flow, and SD to model a quantitative flow.

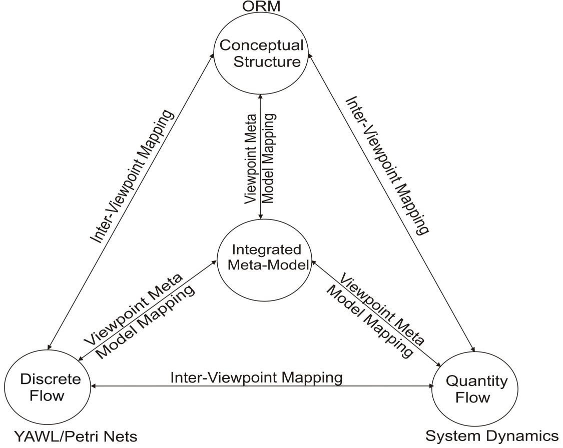

Fig. 1. Abstract View of the Integration

All in all, ORM will add high quality formal conceptualization to SD modeling; YAWL and Petri Nets will serve to bridge the gap between static ORM and Dynamic, flow-like aspects of SD.

2 Problem Definition and Research Motivation

Integrating system dynamics with conceptual and process modeling is the key issue this research will focus on. In order to solve this problem a number of factors need to be studied and analyzed to get a deeper understanding of the problem at hand. This issue has been identified in [21] where it is stated that; *"....there is a strong case for starting to apply systems dynamics methods more openly in the BPM and MIS research fields, as I feel the tools and techniques available are vastly under-rated in terms of their applicability and capability to provide novel representations of real-world situations....."*. This statement is the main idea behind this research. The proposed integration should enable the stakeholders to attain a high level of understanding of the dynamics and statics within the systems studied, enabling decision-makers to make more dependable decisions at different levels.

With ORM focusing on conceptual modeling, YAWL on process modeling and, SD on the dynamics within Business Process Modeling (BPM), this research expects to achieve a well grounded method on which a tool can be based. This will introduce a new breed within enterprize modeling where the static models are merged with dynamic models to give a clear guide to process model development.

2.1 Research Questions

To address this issue, the following questions have been derived:

1. How can we integrate system dynamics with conceptual and process modeling methods?
2. How can the different interactions (concepts) used in these methods be mapped and formally integrated to give a common foundation on which a tool is to be developed?
3. How can the developed integration be used, conceptualized and validated ?
4. How can we give a complete description of the model to fit a tool to be developed?

2.2 Research Objectives

To enable us answer the research questions, we have formulated Objectives to guide us achieve the intended goal. These include;

1. To Integrate System Dynamics with Conceptual and process Modeling.
2. To develop requirements (functional and non functional) on the basis of which a tool can be developed
3. To evaluate the use, conceptualization, and validation of the integrated models.
4. To operationalize the method by making the language usable and provide procedures and guidelines where need be.

3 Literature Review

3.1 System Dynamics Modeling

SD as a method has been in existence since 1961, developed by Jay Forrester to handle socio-economic problems with a focus on the structure and behavior of systems composed of interacting feedback loops. A review and history is given in [8]. SD provides a high level view of the system emphasizing the interactions between its constituent parts, as well as the impact of time on its dynamic behavior [14]. As a method, it has its focus on the structure and behavior of systems composed of interacting feedback loops. The art of SD modeling lies in discovering and representing the feedback processes and other elements of complexities that determine the dynamics of a system [22]. The dynamics arise from the interaction of two types of feedback loops, positive and negative loops. Positive loops tend to reinforce or amplify whatever is happening in the system. Negative loops counteract and oppose change. These loops all describe processes that tend to be self limiting, processes that create balance and equilibrium [19].

Simulation with SD models is used for learning about the dynamic complexity of systems, identification of optimal policies in existing systems, improvement of system behavior through parameter or structural changes. The method has been applied to a wide range of domains, from the management of production-distribution systems to the management of ecosystems. Comparisons between SD and Discrete-event system [2], and between SD and Petri nets [6] have been done. In these comparisons the main differences between SD and these methods are highlighted. [20] further identifies issues for the future of system dynamics.

3.2 Conceptual modeling

The term conceptual modeling [27] is derived from a conceptual model which is an invention to provide an appropriate representation of the target system, appropriate in the sense of being accurate, consistent, and complete [16]. Conceptual models are similar to intermediate causal models proposed by [28] which were developed to capture the meaning of an application domain as perceived by an individual; a precise study can be found in [12]. The method provides artificial models which enable modelers bridge the gap between the experiential world and the abstract mathematical world. Conceptual modeling involves the representation of the entire information system content of the database being designed in somewhat abstract terms relative to the way data is physically stored [5]. A classic view of the conceptual modeling process is presented in [12] where they give a clear description on the fundamental view on the process of conceptual modeling. We use conceptual modeling because the methodologies are well developed and have proven to be quite successful for building information systems in a graphical way at the conceptual level [15].

3.3 Process modeling

Process modeling [4] becomes more and more an important task not only for the purpose of software engineering, but also for many other purposes [1]. Before defining what process modeling is, we start by defining what a process is; [13] describes a process as "*a set of partially ordered steps intended to reach a goal*". [4] notices that any component of a process is a process element and a process step is "*an atomic action of a process that has no externally visible substructure*". With that note a number of scholars have defined process modeling in different ways. [29] defines it as logically capturing and abstracting the systems components, relationships and behavior, with respect to modeling objectives. [4] defines a process model as an abstract description of an actual or proposed process that represents selected process elements that are considered important to the purpose of the model and can be enacted by human or machine. These definitions all state that process models are abstract representations meaning that the system depicts the behavior of an actual system in place. Abstraction can help the modeler to study the behavior of any system with out tampering with the operational system, hence enabling exploration of various options before decisions are made by Stakeholders. Under process modeling we opt to use a Workflow language called YAWL as a workflow method [26]. Workflows are used to define, validate, and automatically manage and monitor the execution of operations (business processes) in organizations. They aim at formalizing activities involving the coordinated execution of multiple tasks performed by different processing actors [3]. History and the various articles on YAWL can be found on the YAWL website [1]

4 Design Approach

4.1 Conceptual Linking and Transformations

We start by identifying the methods to be used in both conceptual (ORM) and process modeling (Petri nets and YAWL). By so doing, we are able to have a clear scope on which methods to use in this study and why. After that, we identify the key concepts as used in these methods, then come up with conceptual links between them. First we consider ORM and SD; then ORM, SD and Petri Nets. We use Petri nets because it is the basis on which YAWL was developed. By starting with Petri nets (foundations of YAWL) we give a strong foundation to the integration. Having mapped the concepts and identified their transformation statements, we then use the model elements and concepts developed to come up with a generic meta-model plus semantics/syntax of the model. After that we will develop the requirements specification on which a tool can be based.

4.2 Case Study and Experimental Modeling Sessions

Case study is an exploratory (single in-depth study) or explanatory (cross-case analysis) research strategy, that involves an empirical investigation of a particular contemporary phenomenon within its real life context using multiple sources of evidence [30]. We will use case study methodology to focus on understanding the dynamics present within a single setting [7], and to understand them within a particular context [31]. We chose to use this method because of its use of many techniques when collecting empirical data.

We will also carry out experimental sessions in different settings to enable us gauge the applicability of the model developed. In these sessions; SD, ORM, Petri nets and YAWL will be used as they all have different but important roles to play as explained earlier. They will also help in better understanding of the integrated model.

---

[1] http://www.yawl-system.com/

4.3 Tool support

In the final development of the integrated model, we will consider the following: Conceptualization links, Mapping, Transformation, Syntax/ Semantics and Description of model integration to fit tool making. Modern SD packages will be used to model the SD model because of their graphical interface making the modeling of a complex system much easier. The SD model(s) will be built based on the Case study results which provide a descriptive model on which the SD conceptual feedback structure will be developed. The feedback structure model will be developed with the help of a Causal Loop Diagram (CLD). CLDs will be converted into Stock and Flow Diagram(SFD) which is a formal quantitative model. Mathematical relationships between or among variables that enable the simulation of the model will be defined thereafter, simulations of the key variables will be run.

5 Preliminary Results

We have so far identified the conceptual links between SD and ORM [23]. To achieve that we used a working example the procedures a paper might go through en route from writing to publication. By using an example we came up with different illustrations to clearly show the link and conceptualize the different concepts as used in both methods.

In Table.1. we show a summary of the mapping, transition statements of the key variables plus their elements. A detailed explanation is formulated and is under review. Having achieved that, we are currently working on a meta-model where we use the mappings plus model elements identified to derive the syntax/semmantics of the methods.

Table 1. SD with ORM and Petri nets

System Dynamics	**ORM**	**Petri nets**	**Transitional Statement**	**Elements**
Stock	Unary fact types	Places	They all contain “things”	Containers
Quantity	Objects	Tokens	These can be looked at as the things that flow with in the system	Contents
Flows (Inflow and Outflow)	Object types	Transitions	They all connect items: Stocks (SD), Unary fact types, (ORM) and Places (Petri Nets)	Homogeneous connectors
Connectors	Fact types	Arcs	They are all active and involve activities that cause a change to the recipient or destination	Heterogeneous connectors

6 Conclusion

This study aims at integrating dynamic and static methods. The methods applied have been in existence for sometime, are well founded and widely applied to different settings in the modeling world. We chose to use these methods because of their complementary dynamic and static aspects in modeling systems. During this study we will come up with a generic meta-model as illustrated in figure.1, and requirements on which a tool can be based. We will apply the approach presented in context of various case domains. We will further develop and refine the method (its models as well as the stepwise process): By devoting more attention to integrate existing formalizations (syntax and semantics), but also to operationalization of the modeling procedures. Finally, we intend to explore further links between SD and process modeling (already initiated by the Petri Net involvement), in particular with the YAWL method [25].

References

1. Becker, J., M., Rosemann, C. von Uthmann.: Guidelines of Business Process Modelling, in: W.M.P van der Aalst et.al.,(Eds.), Business Process Management: LNCS 1806: 30–49, Springer-Verlag, Berlin (2000)
2. Brailsford, S.C. and Hilton, N.A.: "A Comparison of Discrete Event Simulation and System Dynamics for Modelling Healthcare Systems." Proceedings of ORAHS, Glasgow Caledonian University, pp. 18-39 (2000)
3. Chen, P. P., Thalheim, B. and Wong, L. Y.: Future Directions of Conceptual Modeling, in Conceptual Modeling, P. P. Chen, J. Akoka, H. Kangassalo, and B. Thalheim, Eds. Springer LNCS 1565, Berlin, Germany (1999)
4. Curtis, B., Kellner, M. I., and Over, J.: "Process Modeling," Communications of the ACM (35:9), 75–90 September (1992)
5. Date, C. J.: An Introduction to Database Systems, Volume 1, Fifth Ed, Reading: Addison-Wesley (1990)
6. Duggan, J.: "A Comparison of Petri Net and System Dynamics Approaches for Modelling Dynamic Feedback Systems." 24th International Conference of the Systems Dynamics Society, Nijmegen, The Netherlands, July (2006)
7. Eisenhardt, K.M.: Building theories from case study research", Academy of Management Journal, 14 (4): 532–50 (1989)
8. Forrester J.W.: Industrial Dynamics. The MIT Press, Cambridge (1961)
9. Guizzardi, G., Pires, L. F., Sinderen, M. J. v. (2002): On the role of Domain Ontologies in the design of Domain-Specific Visual Modeling Languages. In: Proc. 17th ACM Conference on Object-Oriented Programming, Systems, Languages and Applications (OOPSLA 2002)
10. Halpin, T.: Object-Role Modeling (ORM/NIAM), Handbook on Architectures of Information Systems, Springer, Heidelberg, Ch. 4 (1998)
11. Halpin. T and wagner, G.: Modeling reactive Behavior in ORM. LNCS 2813, :567–569. Springer-Verlag Berlin Heidelberg (2003)
12. Hoppenbrouwers, S.J.B.A, Proper, H.E., van der Weide, T.: A Fundamental View on the Process of Conceptual Modeling. In: Conceptual Modeling - ER 2005 – 24 International Conference on Conceptual Modeling. Volume 3716 of Lecture Notes in Computer Science. 128 - 143 (2005)
13. Humphrey, W.S. and Feiler, P.H: Software process development and enactment: Concepts and definitions. Tech. Rep. SEI-92-TR-4. Pittsburgh: Software Engineering Institute, Carnegie Mellon University (1992)

14. Hustache, J.–C., Gibellini, M., Matos, P.L.: A System Dynamics Tool for Economic Performance Assessment in Air Traffic Management 4th USA/Europe Air Traffic Management R and D Seminar Santa, December 3–7 (2001)
15. Jarrar, M., Demey, J., Meersman, R.: On Using Conceptual Data Modeling for Ontology Engineering. In: Aberer K., March S., and Spaccapietra S., (eds.): Journal on Data Semantics, Special issue on "Best papers from the ER/ODBASE/COOPIS 2002 Conferences", LNCS Vol. 2800, Springer. ISBN: 3-540-20407-5. October 185–207 (2003)
16. Norman, D. A.: Some observations on mental models. In Dedre Gentner and A. L. Stevens (Eds) Mental Models, Lawrence Erlbaum Associates, Hillsdale, NJ. 7-14 (1983)
17. Paige, R.: A meta-method for formal method integration. Proc. Formal Methods Europe, LNCS 1313, Springer-Verlag.(1997)
18. Paige, R.: When are methods complementary? Information and Software Technology 41(3), February, (1999)
19. Pfahl, D. and Lebsanft, K.: Integration of system dynamics modeling with descriptive process modeling and goal-oriented measurement. Journal of Systems and Software, 46: 135-150 (1999)
20. Richardson G.: Problems for the future of system dynamics. System Dynamics Review 12: 141–157 (1996)
21. Sharif, A. M.: Industrial Viewpoint can systems dynamics be effective in modeling dynamic business systems? Business Process Management Journal 11(5): 612–615 q Emerald Group Publishing Limited (2005)
22. Sharma, D., Sahay, B.S., Sachan, A.: Modeling Distributor Performance Index Using System Dynamics Approach., 16(3) (2004)
23. Tulinayo, P.F, Hoppenbrouwers, S.J.B.A.(S.), and Proper, H.A.(E.).: Integrating System Dynamics with Object-Role Modeling. IFIP International Federation for Information Processing. J. Stirna and A. Persson (Eds.): PoEM 2008, LNBIP 15: 77-85 (2008)
24. van der Aalst. W.M.P, ter Hofstede, A.H.M., Kiepuszewski,B. and Barros A.P.: Workflow Patterns. Distributed and Parallel Databases, 14(1):5-51 (2003)
25. van der Aalst, W.M.P, Hofstede, A.H.M.t.: YAWL: Yet Another Workflow Language. Information Systems, 30(4), 245–275 (2005)
26. van der Aalst. W.M.P. and ter Hofstede, A.H.M.: YAWL: Yet Another Workflow Language. QUT Technical report, FIT-TR-2002-06, Queensland University of Technology, Brisbane (2002)
27. Wand, Y., Storey, V., Weber, R.: An Ontological Analysis of the relationship Construct in Conceptual Modeling. In: ACM Transactions on Database Systems, 24(4): 494–528 (1999)
28. White, B.: Intermediate causal models: A missing link for successful science education? In R. Glaser (Ed), Advances in Instructional Psychology, volume 4, Lawrence Erlbaum Associates, Hillsdale, NJ. 177-252 (1993)
29. Williams, D.: Challenges of System Dynamics to Deliver Requirements Engineering Projects: faster, better and cheaper (2003)
30. Williams, D.: Dynamic Synthesis: A Theoretical Framework for Research in Requirements Engineering Process Management. Operational Research Society, ISBN: 0 903440202 (2000)
31. Yin, R. K.: Case Study Research: Design and Methods Applied Social Research Method, Sage, Beverly Hills, California (1988)