Integrating System Dynamics with Object-Role Modeling

P. (Fiona) Tulinayo[1], S.J.B.A. (Stijn) Hoppenbrouwers[1], and H.A. (Erik) Proper[1,2]

[1] Institute of Computing and Information Sciences, Radboud University Nijmegen
Toernooiveld 1, 6525 ED Nijmegen, The Netherlands, EU
F.Tulinayo@science.ru.nl, stijnh@cs.ru.nl
[2] Capgemini, Papendorpseweg 100, 3500 GN Utrecht, The Netherlands, EU
e.proper@acm.org

Abstract. We put Object-Role Modeling (ORM) to work in the context of the creation of System Dynamics (SD) models. SD focuses 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 that determine the dynamics of the system (typically, a process in an organization). However, SD shows a lack of instruments for discovering and expressing precise, language-based concepts in domains. At the same time, the field of conceptual modeling has long since focused on deriving models from natural expressions. We therefore turn to ORM as a prime example of this school of thought to integrate its strong natural language based modeling approach into the creation of SD models. A two-step schema based approach for transforming an ORM domain model into a SD stock and flow diagram is presented. We discuss how typical ORM conceptualization can be linked to SD conceptualization and how such a transformation can be performed. Examples are provided.

Keywords: System dynamics and Object-Role modeling.

1 Introduction

Integration of methodological concepts can be viewed as an improvement of process development. This is because concepts from different approaches supplement each other; hence the weaknesses of one approach are overcome by the strength of the other. The current effort was inspired by an observed lack of concept-level modeling power in SD. We set out mainly to augment SD modeling by first laying down a sound foundation of domain concepts by means of fact-based ORM modeling. This enables us eventually to link the ORM model to an SD stock-flow diagram 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 [4]. The art of SD modeling lies in discovering

J. Stirna and A. Persson (Eds.): PoEM 2008, LNBIP 15, pp. 77–85, 2008.

and presenting the feedback processes and other elements of complexities that determine the dynamics of a system [10]. 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 [6]. In the context of enterprise modeling SD is typically used in process analysis design and optimization. SD models are usually supported by software that allows for process simulation. A main advantage of improving SD's conceptual foundation through ORM is that SD models can be more soundly and readily linked to databases.

In this paper we first explore the conceptual links between SD and ORM, and then present a three step schema based approach for transforming an ORM domain model into an SD stock and flow diagram. Thus this paper presents an exercise in usefully linking two existing and rather different methods in enterprise modeling.

2 System Dynamics Structure

In SD, two diagrams are most commonly used: causal loop diagrams (CLD) and stock-flow diagrams. CLDs show the main feedback loops in a process. They are composed of two concepts (*influences* and *elements*). The influences have a direction indicated by an arrow, and another indicator as to whether the influenced element is changed in the same direction (+) or in the opposite direction (-). The stock and flow diagrams include four different concepts (*Stocks, Flows-Rates, Connectors* and *Converters*). Stocks can be considered reservoirs containing quantities describing the state of the system. Flows (inflows to and outflows from the various levels) can be imagined as pipelines with a valve that controls the rate of accumulation to and from the stocks. The converters contain information in the form of equations or values that can be applied to stocks, flows, and other converters in the model [7].

Connectors and converters measure the quantities in levels and, through various calculations, control the rates. They appear as lines with arrows (connectors) and as circles (converters). In the conventions for the stock and flow diagram:

– Connectors can feed information into or out of flows and converters but only extract information out of the stock.
– Stocks are influenced by flows (in and out) and can influence flows or converters but cannot be influenced by other stocks and Converters.
– The flows can be influenced by stocks and converters but cannot influence converters or other flows, and converters can influence flows or other converters.

The Stock and Flow Diagram in fig. 5 provides an example.

3 Using ORM as a Foundation for SD Models and Modeling Processes

The first step in our approach consists of three sub-steps in which an ORM diagram is augmented and replaced by process concepts that lean towards SD-like conceptualization. The second step consists of two sub-steps that concern the construction of actual CLD and Stock and Flow diagrams. ORM is a fact-oriented approach for modeling information at a conceptual level [5]. Its use is comparable to that of ER [3]. In this paper we use as an example the procedures a paper might go through en route from writing to publication. The procedures are stated as:

1. A person (author) writes intent of submission. This can be in the form of an abstract.
2. Then the content (text of the paper) is submitted, whereby the paper becomes a submitted paper.
3. Each submitted paper receives a classification.
4. Each submitted paper is reviewed
5. Some submitted papers are accepted and some are rejected
6. For each submitted paper new content is submitted, which makes the paper a published paper that is added to the publications.

These statements can be represented on an ORM diagram as indicated below

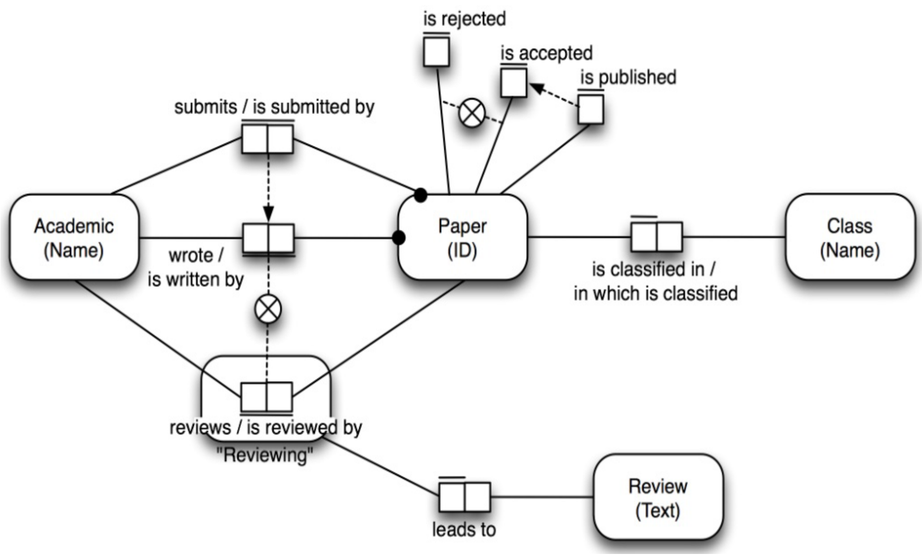

Fig. 1. Paper flow concepts in ORM

Step 1.(a) We started with fig.1 which is an ORM diagram of events (reported as elementary facts) that may be observed in a domain (in this case, the reviewing domain). This approach is in line with the PSM^2 [2] approach. Note the constraints requiring that the submitting academic is indeed one of the authors of a paper, and that a reviewer of a paper cannot be author of that paper.

Step 1.(b) We add temporal dependencies between the roles associated to the paper. This leads to the flow depicted in fig. 2. The left hand side depicts the

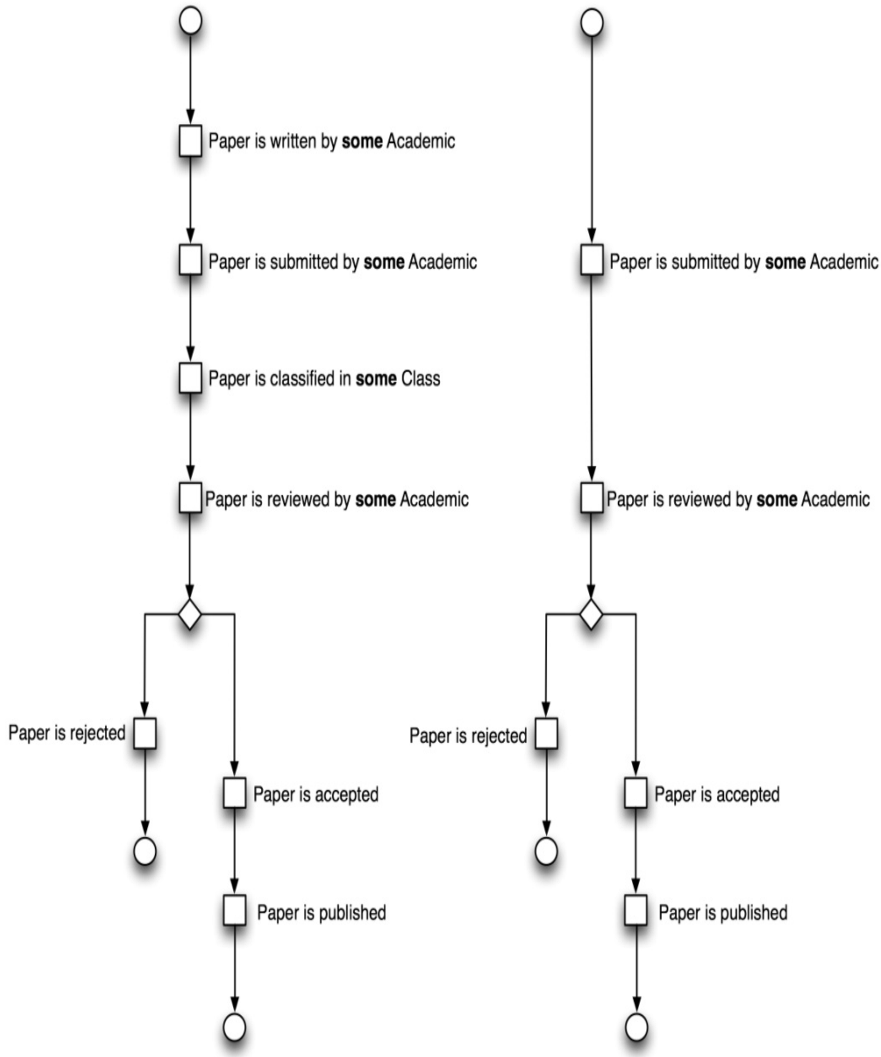

Fig. 2. Paper Flow

full diagram based on the facts types (event types) in the original ORM diagram. The diamond shape is the BPM [11] symbol for a XOR split. Our interest is in the flow statics of submissions, reviews, acceptance, rejection and publication. This leads to the abstracted view depicted on the right hand side.

3.1 ORM Integrated with SD

Step 1.(c)we now make explicit the relations between, in particular, the ORM model and stock-flow model.

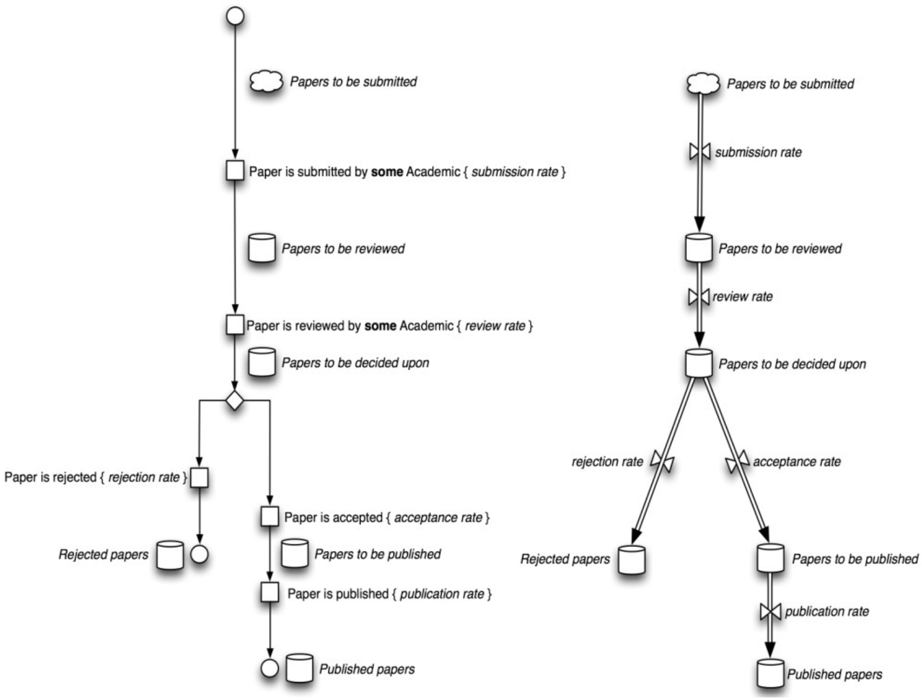

Fig. 3. An integration of ORM with SD

In fig. 3 the left hand side shows fig. 2 (right) with some extra information. The extra information pertains to the flow based interpretation. We now see stores of papers that are ready to flow from one state to another. Each time a paper "flows", this is an event (the original events related to ORM diagram, fig. 1). So:

- A paper is **reviewed**
- A paper is **decided upon**
- Etc.

Associated to the event-types, we can now also add a rate. Leading to:

- Review rate
- Acceptance rate
- Etc.

The right hand side then depicts the SD diagram. This is the prelude to the complete SD Stock and Flow diagram as depicted in fig.5

Step 2.(a) We now embark on identifying the key variables for the SD model: first we create a CLD (in our case, using Vensim simulation software). This helps us identify the influences ('+', '-') variables have on each other.

In SD the system behavior is described as a number of interacting feedback loops, balancing or reinforcing and delay structure. The arrows come together to

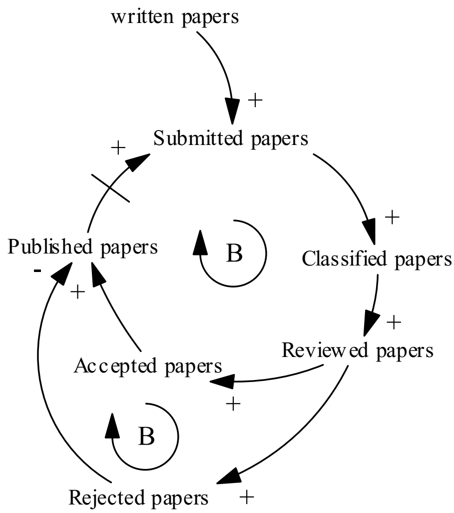

Fig. 4. A causal Loop Diagram for paper flow

form loops, and each loop is labeled with an (R) or a (B). "R" means reinforcing; i.e., the causal relationships within the loop create exponential growth or collapse and "B" means balancing; i.e., the causal influences in the loop keep things in equilibrium. In this case we only have balancing loops. Causal links from one variable to another can be marked as positive or negative based on how their variables change. If number of links with negative sign on the loop is even, then the loop is self-reinforcing (R) and if the number of links with negative on the loop is odd, then the loop is a balancing loop (B).

In Fig.4 a number of variables are used inline with the prior diagrams, each with a direction (arrow) and an indicator (+/-). There is a delay mark between variable published papers and submitted papers indicated with a single slash on the arrow. The delay mark implies that there is a time lapse before the variable at the arrow tip is affected. In this case the submitted papers move in the same direction as classified papers; if one increases the other also increases; the reverse is also true. This holds for all cases where there is a '+' influence (polarity). Once the papers are submitted they are reviewed and the results are released. Based on this, papers are either accepted or rejected. The rejected papers affect the published papers in the opposite direction; an increase in the rejected papers causes a decrease in the published papers and *vice-versa*. Yet the accepted papers affect the papers to be published in the same way where an increase in the accepted papers causes an increase in the published papers.

3.2 Stock and Flow Diagram

After developing the CLD and identifying the different polarities required, we convert the model into a Stock and Flow diagram to show how the system components interact. The Stock and Flow diagram is more complex and detailed than the CLD because it includes nodes for each of the model parameters. It is also used to develop a set of equations, which are used in a numerical simulator to generate (simulate) the behavior of the system.

Step2.(b) The stock and flow diagram is constructed using the Powersim application which takes its name from "Powerful Simulation." It is a simulation tool based on the system dynamics methodology. The Stock and Flow diagram is used to show flow dependencies and how quantities are distributed within the system. Stocks hold quantities that are subject to accumulation through inflows, or are subject to reduction through outflows.

We have the stocks as submitted papers, papers to be decided upon, rejected papers, Accepted papers, and published papers. These stocks have inflows and out flows that are regulated by means of valves. The valves determine the rate at which an inflow or outflow of material applies to the stock (box). In this model there are different factors that affect the flows, and these are either positive or negative as reflected in the CLD. These effects can be indicated as constants linked to the flows or stocks with a connector. During the development of the stock and flow model a number of experimental simulations is normally run to show the different behaviors of the system studied. In our case, such simulations were also carried out, on selected simulation parameters. While doing so, the

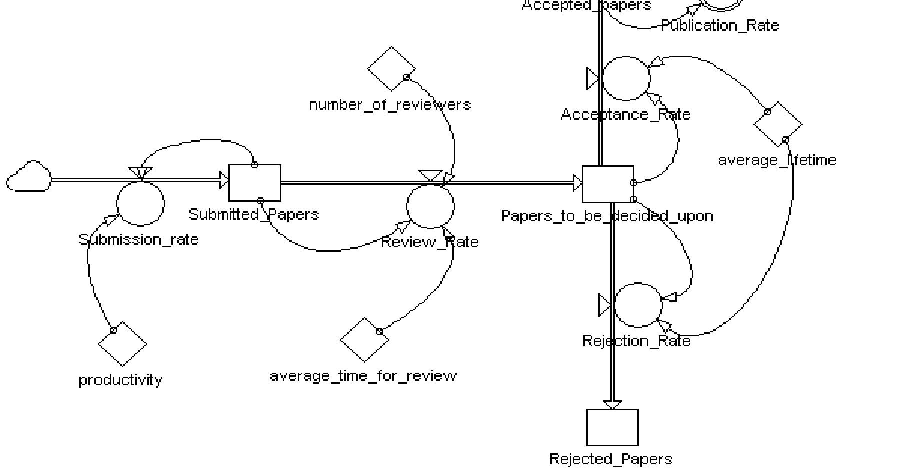

Fig. 5. A stock and flow diagram for the paper life cycle

model can be paused, and each of the stocks continues to hold their quantity for observation. If the value of a particular stock is not important to the problem at hand, then the stock is shown as a cloud, to indicate that it is outside the boundary of the model. This procedure is also known as sensitive analysis [8].

From fig.5 each rate is clearly defined as follows;

Let X and Y be the input and output of some flow $X \Rightarrow Y$:
Then the rate for this flow at time t is defined as

$$\mathsf{Rate}(X \Rightarrow Y) \triangleq \frac{\mathsf{TotalPop}_t(Y)}{\mathsf{TotalPop}_t(X)}$$

Where Totalpop $_t$(X) is the set of all instances ever of stock.
For cases where we have more outflow rates from the same stock and more inflow rates from different stocks we would have for each stock X:

$$\mathsf{TotalPop}_t(X) \triangleq \mathsf{Pop}_t(X) \cup \bigcup_{Y:X \Rightarrow Y} \mathsf{AddedFlow}_t(X,Y)$$

$$\mathsf{AddedFlow}_t(X,Y) \triangleq \mathsf{TotalPop}_t(Y) - \bigcup_{X':X' \Rightarrow Y \wedge X' \neq X} \mathsf{TotalPop}_t(X')$$

$$\mathsf{TotalPop}_t(X) \triangleq \mathsf{Pop}_t(X) \cup \bigcup_{Y:X \Rightarrow Y} \left(\mathsf{TotalPop}_t(Y) - \bigcup_{X':X' \Rightarrow Y \wedge X' \neq X} \mathsf{TotalPop}_t(X') \right)$$

Note that in general this will lead to a recursive system of equation. As all instances have assigned unique location at each moment, this recursive system of equations will have a unique location.

4 Conclusion

In this paper we have identified the extent to which features of ORM static models can be transformed (with added information) into SD models. The two methods are rather different but when used together for a common goal we believe the results are not only better grounded but also more decisive and reliable. The ORM methodology equips the modeler with strong conceptualization of the domain. This is key to developing any model. By combining SD concepts with ORM style modeling we manage to better capture the static part of the model, and to link it satisfactorily with the dynamic aspect. This can enable stakeholders to make better decisions in BPM and process optimization.

5 Further Research

In the near future we will apply the approach presented in context of various case domains. We will further develop and refine the method and its diagrams and also devote more attention to formalizing its syntax and semantics. In addition we intend to use the techniques suggested in this paper in collaborative settings such as group model building which is a sub discipline with in the field of SD [9]. Finally, we intend to explore further links between SD and process modeling, in particular with the YAWL method [1].

References

1. Aalst, W.M.P.v.d., Hofstede, A.H.M.t.: YAWL: Yet Another Workflow Language. Information Systems 30(4), 245–275 (2005)
2. van Bommel, P., Frederiks, P.J.M., van de Weide, T.P.: Object–Oriented Modeling based on Logbooks. The Computer Journal 39(9), 793–799 (1997)
3. Chen, P.P.: The entity-Relationship model-Towards a unified view data. ACM Transactions of database systems 1(1), 9–36 (1976)
4. Forrester, J.W.: Industrial Dynamics. The MIT Press, Cambridge (1961)
5. Halpin, T., Wagner, G.: Modeling Reactive Behavior in ORM. In: Song, I.-Y., et al. (eds.) ER 2003, vol. 2813, pp. 567–569. Springer, Heidelberg (2003)
6. 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)
7. Leaver, J.D., Unsworth, C.P.: System dynamics modeling of spring behavior in the Orakeikorako geothermal field. Elsevier Ltd., Amsterdam (2006)
8. Mutschler, B., Reichert, M.: On Modeling and Analyzing Cost Factors in Information Systems Engineering. In: Bellahs‘ene, Z., Leonard, M. (eds.) CAiSE 2008. LNCS, vol. 5074, pp. 510–524. Springer, Heidelberg (2008)
9. Rouwette, E., Hoppenbrouwers, S.J.B.A.: Collaborative systems modeling and group model building: a useful combination? In: 26^{th} International Conference of the System Dynamics Society (2008)
10. Sharma, D., Sahay, B.S., Sachan, A.: Modeling Distributor Performance Index Using System Dynamics Approach., vol. 16(3) (2004)
11. White S.A.: Business Process Modeling Notation (BPMN) Version 1.0. BPMI. org (May 3, 2004), `http://www.bpmn.org`