

Language Action-Based Simulation of Business Processes

Peter Rittgen

University College of Borås, 501 90 Borås, Sweden

peter.rittgen@hb.se

Abstract

Simulation of a business process is often used to visualize its behaviour, to identify problems and to improve its performance. It requires the design of a simulation model, a detailed model of the processes, the resources and the variables to be monitored. Here we investigate how a language-action model of the business process can support the development of such a simulation model.

Keywords: Language-action model, simulation model, resources

1 Introduction

An important task of business process reengineering is to identify and put into practice alternatives to the current organization that improve the business with regard to key success factors (Hammer and Champy, 1993). In the case of a complex process this might be difficult to undertake even if we can avail ourselves of a model of the business process. In such a situation it can be very helpful to see the relevant process in action and to be able to “play with it,” i.e. to carry out a simulation (Hlupic and Robinson, 1998). Modeling becomes even more difficult in an interorganizational setting, where more than one business is concerned, due to the multiplicity of decision making levels involved (Giaglis et al., 1997), which makes the use of simulation even more advisable. This has the additional benefit of helping in assessing the impact of suggested changes before we try them out in “real life” where mistakes can be very costly (Giaglis et al., 1999). As a consequence a new area of research called Business Process Simulation (BPS; Paul et al., 1999) has been established.

We encountered a situation that suggested the use of simulation during an interorganizational case study we carried out in spring 2004. It involved two companies, a logistics provider and a retail chain. One of the principal problems we found was related to capacity reservation. The retail chain provides the logistics company with forecasts of future incoming and outgoing deliveries. These are taken as capacity reservations and are used for capacity planning. A new framework contract between the companies that was about to be negotiated should specify appropriate measures in the case of deviations from the reserved capacities, i.e. for the case that the actual deliveries do not agree with the planned ones for which respective capacities have been reserved. It seemed therefore reasonable to create a simulation model to study the impact of deviations on transaction costs and delayed deliveries. Such a model is typically developed “from scratch” with the help of a specific simulation language. A language-action model is therefore not a necessary prerequisite for this task but it can facilitate the latter considerably. This is particularly true in situations where we can make use of an existing language-action model which was not specifically developed for the purpose of simulation, i.e. it comes “for free”. Such was the case in the project described here.

In the following sections we first introduce simulation in the context of business processes, and then an action-oriented language for modeling business processes that is related to

DEMO (Dynamic Essential Modeling of Organization). The latter is complemented by an additional view, the resource view, to facilitate the development of a simulation model. We go on showing how the extended language-action model can contribute to the design of the simulation and conclude with a comprehensive example concerning the outbound logistics from our case study and the results from this simulation.

2 Simulation of Business Processes

A simulation model is an abstracted, formal description of some real or imagined system. A simulation is an enactment of such a model that allows us to

1. observe the potential behaviour of a system that does not (yet) exist, or to
2. observe the (potential) behaviour of an existing system at a much faster pace and at lower costs than that of the real system and without disturbing it.

If an appropriate abstraction is chosen the results of the simulation will represent a fair approximation of the behaviour of the real system (or the imagined system if it were built). In the area of business processes we primarily use a specific kind of simulation which is called discrete-event simulation (Fishman, 2001). In it a system is described in terms of states and events. A system remains in a state (i.e. unchanged) for some time until an instantaneous occurrence, an event, happens taking it to a new state. SIMULA (SIMULATION LANGUAGE) was the first language to create simulation models that could be processed by a computer (Nygaard, 1962). It provided passive objects (customers) and active objects (stations). In later versions they were merged into the single concept of a process. Towards the end of the 60s SIMULA had turned into a full-blown programming language (Dahl, Myhrhaug and Nygaard, 1968) which formed the root of object-oriented programming (Dahl, 2002). The language we use here, SimPy (Simulation in Python; SimPy Developer Team, 2004), is a modern, object-oriented and process-based discrete-event simulation language that continues the tradition of SIMULA. It provides processes, resources, monitors (variables to record observations of system behaviour), random variates to simulate stochastic behaviour, waiting times to simulate the duration of activities and user interfaces to present the data of the simulation.

Simulation can be useful for visualizing and evaluating business processes (Hlupic and Robinson, 1998; Giaglis et al., 1997; Giaglis et al., 1999; Paul et al., 1999). In the former role it enables and enhances understanding of a possibly complex process and it allows a viewer of the simulation to envisage the workings of the process and to assess the consequences of changes to the process for its overall behaviour. In this way problems can be detected more easily and possible solutions can be identified by experimenting with the model (e.g. by trying out alternative process structures). In the evaluating role the simulation gathers numeric data on the performance of the business process such as lead times, delays, throughput, waiting times, different types of costs and servicing times. Based on these results the process can be optimized with respect to costs, output, efficiency, quality of service and so on (see e.g. Weyland and Engiles, 2003). The data can also be used as a basis for business decisions such as the pricing of services. This point is elaborated in the example section.

3 A Language-Action Model of Business Processes

Language-action approaches have their theoretical foundations in speech-act theory (Austin, 1962; Searle, 1969) and the communicative-action theory (Habermas, 1984). Examples of such approaches are Conversation-for-Action (Winograd and Flores, 1986), Dynamic

Essential Modelling of Organizations (DEMO; Dietz and Habing, 2004; Liu et al., 2003; Reijswoud and Dietz, 1999), Action Workflow (Kethers and Schoop, 2000; Medina-Mora et al., 1992; Denning and Medina-Mora, 1995) and Business Action Theory and SIMM (Goldkuhl and Lind, 2004; Goldkuhl and Röstlinger, 1993; Goldkuhl, 1996). According to (Dietz and Habing, 2004) the world is divided into three areas: *actors* (whom we will also call subjects), the coordination world (C-world, also known as the intersubjective world) and the production world (P-world, also termed objective world). We call the members of the P-world *objects*. The actors act in both worlds (C and P). Actions in the P-world are called *productive actions*. They are performed by actors on objects and/or with the help of objects (e.g. machines). Actions in the C-world are called *communicative actions*. They involve two actors. The basic unit of communication is a speech act (Austin, 1962; Searle, 1969). A conversation is a structured sequence of actions. It is assumed that conversations follow characteristic patterns such as that of the workflow loop (or action workflow, Medina-Mora, Winograd, Flores, & Flores 1992) which consists of two consecutive conversations that aim at reaching an agreement about 1) the execution of an action, and 2) the result of that execution. The first conversation is called actagenic, the second factagenic. Some authors (e.g. Reijswoud and Dietz, 1999) define the concept of a transaction that includes the agreed-upon action.

3.1 Introducing Resources into the Language-Action Model

Resources do usually not play a prominent role in business process models from the language-action perspective. This abstraction is admissible and even useful when our primary concern is the structure of the business process where the communicative actions are in the center of our attention. But as soon as performance plays a role the focus shifts to the productive actions and the resources can no longer be neglected. When we plan to do a simulation we are often interested in measuring performance parameters. Hence resources form a vital part of simulation models.

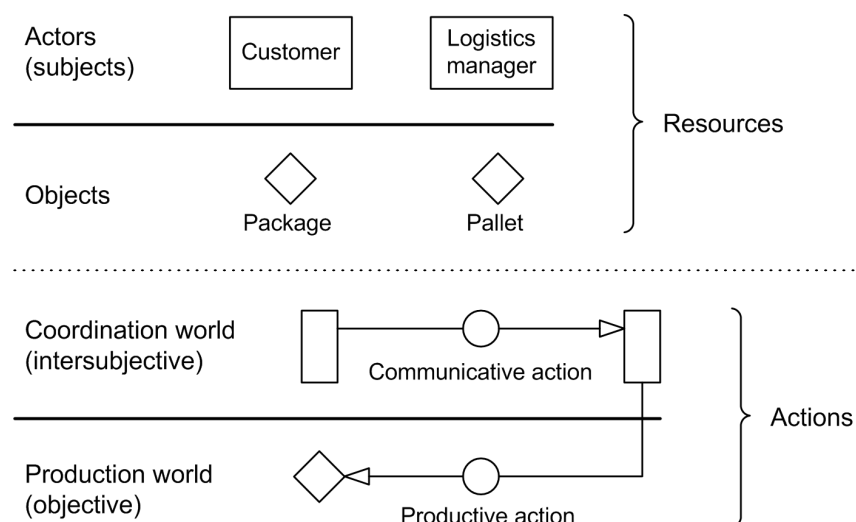


Figure 1: Resources and actions

As resources are closely related to actions it makes sense to introduce them into the action model. Fig. 1 suggests how this can be done. There are two types of resources: actors and objects. Actors are denoted by rectangles and objects by diamonds. This corresponds to the symbolism introduced in (Dietz and Habing, 2004). The actors can perform actions of two types, which are represented by a circle: A communicative action takes place mainly within

the coordination world and involves two actors. It can be a transaction, a conversation or a speech act. A productive action is performed by an actor on an object and/or with the help of objects. It involves only one actor.

There are two behavioural views: the action view and the process view (see fig. 2). The action view shows how actors interact with each other through transactions. The actors involved in a transaction are called *initiator* and *executor* with the arrow pointing towards the latter. A transaction is a complex communicative action that consists of three parts which are themselves actions: An actagenic conversation that is followed by a target action which is in turn followed by a factagenic conversation. The transaction is started by the initiator with the aim of getting a target action done by the executor. The target action is often a productive action but can also be a communicative action.

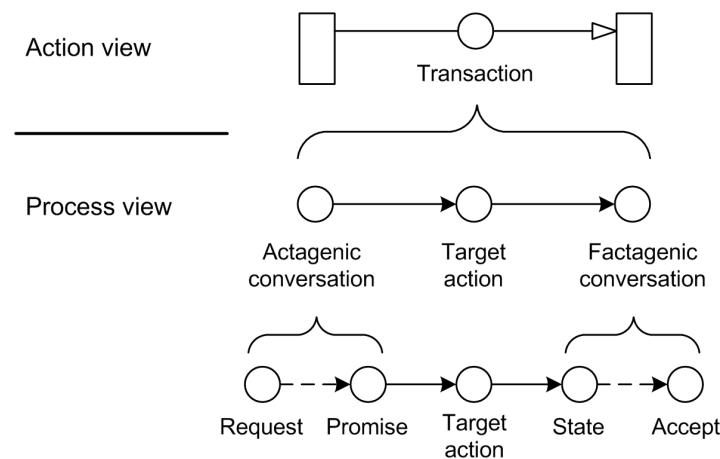


Figure 2: Action view and process view

The process view shows the actions on the conversation or speech-act level including the target action. Speech acts involve two actors, a *performer* and an *addressee*. They are not shown in the process view. The actions are ordered by causal and conditional relations between them. A causal relation is denoted by a solid arrow, where $A \rightarrow B$ means that B is a necessary consequence of A, i.e. if A has been done, B **must** also be done. A conditional relation is shown with the help of a dashed arrow from A to B meaning A is a necessary prerequisite of B, i.e. if A has been done, B **can** be done (but B cannot be done in the absence of A).

3.2 The Resource View

To perform an action different types of resources are required depending on the type of action. We will assume that each resource is used exclusively by a certain action during the whole execution. This implies that the action requests (and is ultimately granted) the sole use of the resource before it starts. Upon completion the resource is released and can subsequently be requested by other actions. We will not consider so-called consumptive resources that “die” in the process. A speech act takes place strictly in the intersubjective world and it therefore requires as resources only its participants, i.e. the performer and the addressee (see fig. 3, on the left). In the resource view the use of a resource is shown by a dashed arrow (i.e. a conditional relation) from the resource to the action. To facilitate the creation of the simulation model we can specify key performance characteristics of the action already in the resource view, for example execution time (deterministic number or probabilistic distribution), the costs associated with it or variables for performance measurement. If it

seems more appropriate they can also be added to the simulation model later. Please observe that the basic information for generating the resource view concerning communicative actions is already contained in most conventional language-action models of business processes so that no extra effort is required.

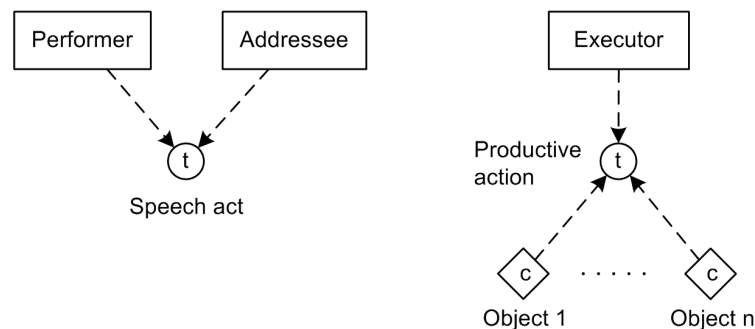


Figure 3: Resource view

Concerning the productive actions the situation is depicted in fig. 3, on the right. The executor is the only actor that is involved. The other resources are objects that are used in the action such as machines, tools etc. They might also include staff that does not perform in an actor role. The information about required objects is not always present in language-action models of business processes. This implies that some extra effort is required here. We found in our case study that this extra effort is by far outweighed by the savings in setting up the simulation model faster so we consider that this effort is well justified.

4 Designing the Simulation Model

Even the information that is present in the extended language-action model (i.e. the one including the resource view) is not sufficient to generate a complete simulation model automatically but it can contribute significantly. When we tried to design the simulation model described in the next section, we found the task very difficult (not being simulation experts) and even after investing several hours we did not succeed in arriving at a satisfactory model. We tried to make use of the existing language-action model of the business process but that did not help us as much as we had expected. When we investigated the reasons for this we found that we knew too little about the relations between the processes (that is what actions are called in many simulation languages) and the resources. That gave us the idea to create a resource view for the language-action model. With the help of that we were able to set up the skeleton of the simulation model in a very short time (this part of the job could actually be done automatically by a computer but in the absence of appropriate software we did it by hand). With this skeleton as a guideline we found it much easier to “fill in the gaps,” i.e. to specify the variables and monitors for performance measurement.

In the remainder of this section we give a detailed description of how to create the skeleton of the simulation model. We do so with the help of a commonly available simulation language called SimPy (Simulation in Python; SimPy Developer Team, 2004) which is based on the programming language Python (Python Software Foundation, 2005). SimPy is available under the GNU Lesser General Public License, Python under the Python Copyright which is GPL-compatible. We only make use of standard features of discrete-event simulation so that this approach should also work with most other simulation languages, systems and tools.

For each action a_i a simulation process has to be defined. This is done as follows (triple quotation marks enclose comments, variables in italics are to be replaced by the corresponding resources from the resource view; see fig. 3):

```
class a_i(Process)
    """ Define simulation variables """
    """ Define __init__(self) """
    def Run(self):
        """ Set simulation variables and monitors """
        yield request, self, Performer """ only in case of a speech act """
        yield request, self, Addressee """ only in case of a speech act """
        yield request, self, Executor """ only in case of productive action """
        yield request, self, Object 1 """ only in case of productive action """
        """ ... """
        yield request, self, Object n """ only in case of productive action """
        yield hold, self, t """ duration of action a_i """
        yield release, self, Performer """ only in case of a speech act """
        yield release, self, Addressee """ only in case of a speech act """
        yield release, self, Executor """ only in case of productive action """
        yield release, self, Object 1 """ only in case of productive action """
        """ ... """
        yield release, self, Object n """ only in case of productive action """
        """ Set simulation variables and monitors """
        a_i_finished.signal()
```

In the main body of the program we add the following lines for each actor A_i and object O_i :

```
A_i = Resource (1)
O_i = Resource (capacity) """ where capacity = number of objects of type O_i """
```

Also in the main body of the program we add an activate command for each action. This command depends on the precedence order, more specifically on the incoming causal and/or conditional relations in the process view. Actions a_i without antecedents are activated as follows:

```
activate (a_i, a_i.Run(), delay = 0.0)
```

Actions a_i with at least one causal antecedent are activated with the commands:

```
yield waitenv, self, a_j_finished """ for each antecedent a_j """
activate (a_i, a_i.Run(), delay = 0.0)
```

Actions a_i that only have conditional antecedents are activated as follows:

```
yield waitenv, self, a_j_finished """ for each antecedent a_j """
if condition:
    activate (a_i, a_i.Run(), delay = 0.0)
```

To complete the simulation model we have to define and set the simulation variables and monitors as mentioned in the comments. In addition to that the collected data has to be output in a suitable format and an appropriate number of simulation runs has to be carried out.

5 Example: Outbound Logistics

5.1 Deriving the simulation model

In the course of our case study we developed simulation models for the inbound and outbound logistics performed by the logistics provider. The example we give is part of the outbound logistics. The corresponding action view is shown in fig. 4.

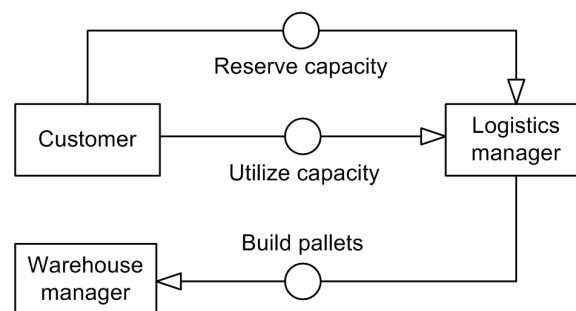


Figure 4: Outbound logistics (action view)

The customer (i.e. the retail chain) reserves the capacity for handling a certain number of packing units with two months notice. The reserved capacity pertains to one week, i.e. the reservation made in week 15 constitutes a forecast of the number of packing units to be handled in week 23. The logistics manager uses the capacity figures to schedule the staffing of the warehouse. Two months later the customer sends an order containing the actual packing units to be delivered to the shops during the current week. This constitutes a utilization of capacity but not necessarily the one that was reserved. The number of packages can (and often does) disagree with the amount for which capacity was reserved. In extreme cases the actual number was three times higher than the forecast putting the logistics provider under enormous pressure. But even in general the forecasts were not sufficiently reliable. The order is then forwarded to the warehouse manager who will instruct his staff to build the pallets, one for each shop, according to the order. If the number of packing units exceeds the forecast the warehouse manager has to mandate overtime or to call in extra staff. If these measures are not sufficient deliveries can be delayed.

When decomposing the transactions into the speech acts and productive actions we get the process view as shown in fig. 5. The business process starts with a request speech act for a certain capacity which is then used to schedule the warehouse staff accordingly (target action). Note that the promise part of the actagenic conversation is omitted because the framework contract forces the logistics provider to accept each request. After that the availability of the capacity is confirmed (speech act: State). Due to the framework contract we can omit the speech acts “promise” and “accept” in this and later transactions. The next step in the process is the order that is sent by the customer. Observe that this action is not caused by the confirmation of the capacity because the customer might decide not to make use of the capacity and not send an order. Hence there is no causal relation. But on the other

hand, the order cannot be sent without prior reservation of capacity which makes the relation conditional (dashed arrow).

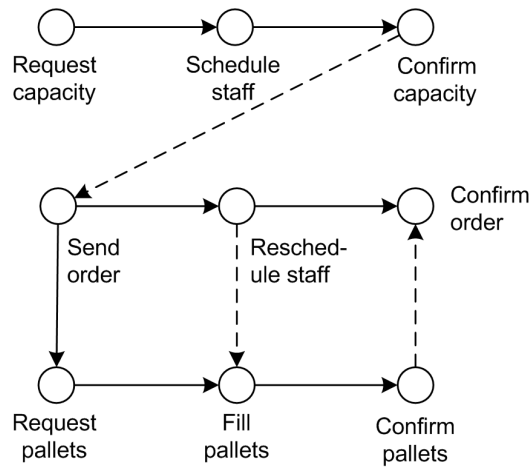


Figure 5: Outbound logistics (process view)

The information in the order is then used to reschedule the staff depending on the actual package load. This might involve that the outbound staff is required to do extra hours or that inbound staff is reassigned. At the same time the building of the pallets is requested but they cannot be filled before the staff for this task is rescheduled, so that sufficient staff is available. Each pallets is then filled with the goods destined for that particular shop, and the completion of the pallets is confirmed. This allows us to also confirm the completion of the whole order.

The work described so far was part of the initial business analysis where we also identified problems and goals. One of the most pressing problems (from the point of view of the logistics provider) was related to the discrepancies between planned and actual capacities so we suggested to do a simulation of the relevant parts of the overall process to determine how these deviations affect transaction costs and delays in delivery. As mentioned earlier we found it difficult to develop the simulation model with the sole help of the models developed so far. We therefore decided to introduce a new view of the business process to get a clearer picture of the use of resources by the actions in the process. As a result we got the resource view that is shown in fig. 6. It contains information than can be derived directly from the other views but also some information that is new.

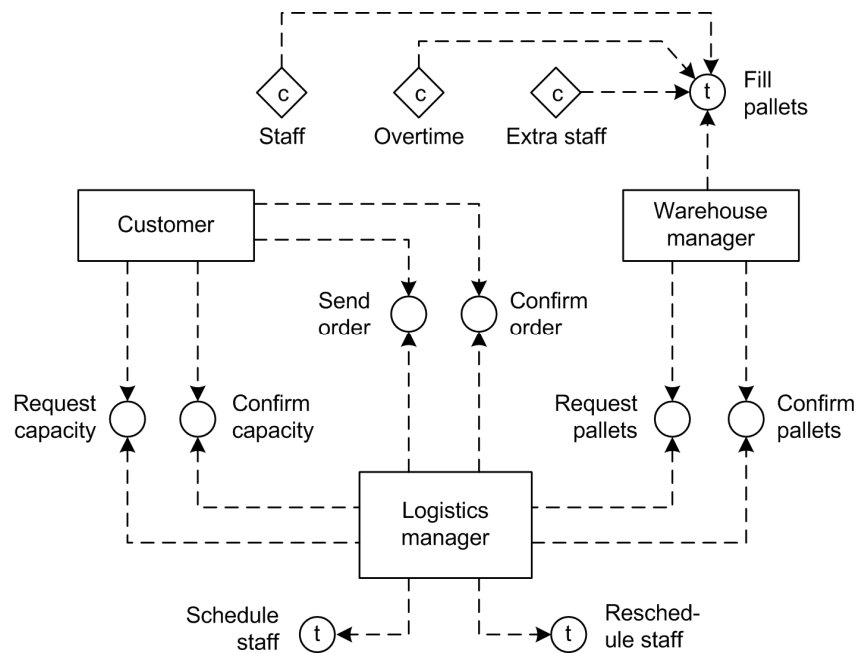


Figure 6: Outbound logistics (resource view)

The resource view shows the actors and objects that are involved in each action. We assume that an actor who is engaged in an action cannot perform another action at the same time. Most of the information contained in fig. 6 can be derived from the action and process views (figs. 4 and 5, respectively) in the following way: For each action in the process view find the corresponding transaction in the action view and from there the actors involved (initiator and executor). These are the resources of the respective speech act. The initiator becomes the performer of the request and accept acts and the addressee of the promise and state acts. Likewise the executor will be the performer of the promise and state acts and the addressee of the request and accept acts. If the action is productive we drop the initiator and record only the executor as a resource. In the example of fig. 6 this procedure yields an almost complete diagram with only three resources missing. These concern the action “Fill pallets” that requires additional resources: the scheduled staff, extra staff that might be called in and overtime of the scheduled staff. The use of these resources is associated with certain costs (“c” in fig. 6). The time (“t”) for filling the pallets depends on the actual number of packing units to be handled, the number of available staff (incl. extra staff), the overtime and the time required for handling a unit. The latter is assumed to be normally distributed with given μ and σ . Packing units that cannot be handled during the week in question have to be treated in the following week which leads to delays and further overtime. The time for (re)scheduling is also normally distributed with given μ and σ . All other actions are assumed to require a negligible time.

5.2 Results of the simulation

Table 1 shows the simulation results for 5 successive weeks (out of 52). The complete results suggest a new pricing model where the unit price is raised by 0.10 € if the reserved capacity is exceeded by more than 20 %, and by 0.02 € for each percentage point the actual value falls short of the forecast. In certain individual cases the transaction costs can differ considerably from this pricing model due to the non-linear nature of the overall process, but on the whole it compensates the logistics provider fairly well for the increased transaction costs and it is definitely superior to the original model which assumed a constant unit price.

Table 1: Simulation results

Capacity	Actual units	Scheduled staff	Extra staff	Overtime	Handled units	Excess units	Total costs	Costs per unit
3121	2775	6	0	-18,00	2775	0	3.600,00 €	1,30 €
2459	3777	5	0	50,00	3125	652	4.000,00 €	1,28 €
3423	3226	7	0	-21,92	3226	0	4.200,00 €	1,30 €
2142	3554	4	0	40,00	2500	1054	3.200,00 €	1,28 €
3769	4988	8	1	39,04	4988	0	6.180,80 €	1,24 €

6 Conclusion

Simulation can be helpful in visualizing a complex business process or in assessing the impact of changes to a process on performance characteristics. But setting up a simulation model is a difficult task even for experienced modelers. We started from the assumption that both the language-action model and the simulation model are essentially business process models and that therefore the former should be helpful in developing the latter. But the two types of models focus on different aspects of the process: the former on coordination and the latter on performance. The concept that holds the two together is that of a resource. It exists implicitly also in language-action models but it has to be made explicit when defining the simulation model. By introducing an explicit notion of resource and the corresponding resource view into the language-action perspective we facilitate the elicitation of resource dependencies and thereby support the design of a simulation model.

References

- Austin JL (1962) *How to Do Things with Words*. Oxford University Press, Oxford, UK.
- Dahl OJ (2002) *The Roots of Object Orientation: The Simula Language*. In *Software pioneers: contributions to software engineering*, Proceedings of the Software Pioneers' Conference, Bonn, June 2001 (Broy M and Denert E, Eds), pp. 78-90, Springer, Berlin, Germany.
- Dahl OJ, Myhrhaug B and Nygaard K (1968) *SIMULA 67 Common Base Language*. Norwegian Computing Center, Oslo, Norway.
- Denning PJ and Medina-Mora R (1995) *Completing the Loops*. *Interfaces* 25(3), 42-57.
- Dietz JLG and Habing N (2004) *The Notion of Business Process Revisited*. In *Proceedings of the OTM Confederated International Conferences, CoopIS, DOA, and ODBASE* (Meersman R and Tari Z, Eds), pp. 85-100, Springer, Berlin, Germany.
- Fishman GS (2001) *Discrete-Event Simulation: Modeling, Programming, and Analysis*. Springer, Berlin, Germany.
- Giaglis GM, Paul RJ and Doukidis GI (1997) *Simulation for Intra- and Inter-Organisational Business Process Modelling*. *Informatica* 21(4), 613-620.
- Giaglis GM, Paul RJ and Hlupic V (1999) *Integrating Simulation in Organisational Design Studies*. *International Journal of Information Management* 19(3), 219-236.

- Goldkuhl G (1996) Generic business frameworks and action modelling. In Proceedings of the First International Workshop on Communication Modeling (Dignum F, Dietz J, Verharen E and Weigand H, Eds), Electronic Workshops in Computing, Springer, Berlin, Germany.
- Goldkuhl G, Lind M (2004) The generics of business interaction - emphasizing dynamic features through the BAT model. In Proceedings of the 9th International Working Conference on the Language-Action Perspective on Communication Modelling LAP 2004 (Aakhus M and Lind M, Eds), pp. 1-26, Rutgers University, New Brunswick, NJ, USA.
- Goldkuhl G and Röstlinger A (1993) Joint elicitation of problems: An important aspect of change analysis. In Human, Organizational, and Social Dimensions of Information Systems Development (Avison D, Kendall J and DeGross J, Eds), North-Holland, Amsterdam, The Netherlands.
- Habermas J (1984) The Theory of Communicative Action 1, Reason and the Rationalization of Society. Beacon Press, Boston, MA, USA.
- Hammer M and Champy J (1993) Reengineering the Corporation: A Manifesto for Business Revolution. Harper Business, New York, NY, USA.
- Hlupic V and ROBINSON S (1998) Business Process Modelling and Analysis Using Discrete-Event Simulation. In Proceedings of 1998 the Winter Simulation Conference (Medeiros DJ, Watson EF, Manivannan M and Carson J, Eds), pp. 1363-1369, The Society for Computer Simulation, San Diego, CA, USA.
- Kethers S and Schoop M (2000) Reassessment of the Action Workflow Approach: Empirical Results. In Proceedings of the Fifth International Workshop on the Language-Action Perspective on Communication Modelling LAP 2000 (Schoop M and Quix C, Eds), pp. 151-169, RWTH Aachen University, Germany.
- Liu K, Sun L, Barjis J and Dietz JLG (2003) Modelling dynamic behaviour of business organisations - extension of DEMO from a semiotic perspective. Knowledge-Based Systems 16(2), 101-111.
- Medina-Mora R, Winograd T, Flores R and Flores F (1992) The Action Workflow Approach to Workflow Management Technology. In Proceedings of the Conference on Computer-Supported Cooperative Work CSCW'92 (Turner J and Kraut R, Eds), pp. 281-288, ACM, New York, NY, USA.
- Nygaard K (1962) SIMULA: An Extension of ALGOL to the Description of Discrete-Event Networks. In Information Processing, Proceedings of IFIP congress 62, Munich, Aug 1962 (Popplewell, CM, Ed), pp. 520-522, North-Holland, Amsterdam, The Netherlands.
- Paul RJ, Giaglis GM and Hlupic V (1999) Simulation of Business Processes. American Behavioral Scientist 42(10), 1551-1576.
- Python Software Foundation (2005) Python Programming Language. <http://www.python.org/>
- Reijswoud VEV and Dietz JLG (1999) DEMO Modelling Handbook. TU Delft, The Netherlands. Online version available at <http://www.demo.nl/documents/handbook.pdf>.
- Searle JR (1969) Speech Acts, An Essay in the Philosophy of Language. Cambridge University Press, London, UK.
- SimPy Developer Team (2004) SimPy Homepage. <http://simpy.sourceforge.net/>

Weyland JH and Engiles M (2003) Towards Simulation-based Business Process Management. In Proceedings of the 2003 Winter Simulation Conference (Chick SE, Sánchez PJ, Ferrin D and Morrice DJ, Eds), pp. 225-227, Institute of Electrical and Electronics Engineers, Piscataway, New Jersey, USA.

Winograd T and Flores F (1986) Understanding Computers and Cognition: A New Foundation for Design. Ablex, Norwood, NJ, USA.