AMCIS 2009 Proceedings

Americas Conference on Information Systems (AMCIS)

2009

Design of Causal Coupling Patterns Supporting Causal Modeling and Flow Modeling in the System Dynamics Methodology

Christian Suchan
Otto-Friedrich Universitat Bamberg, christian.suchan@uni-bamberg.de

Recommended Citation

Suchan, Christian, "Design of Causal Coupling Patterns Supporting Causal Modeling and Flow Modeling in the System Dynamics Methodology" (2009). *AMCIS 2009 Proceedings*. 742.
http://aisel.aisnet.org/amcis2009/742

Design of Causal Coupling Patterns Supporting Causal Modeling and Flow Modeling in the System Dynamics Methodology

Christian Suchan
Otto-Friedrich-University of Bamberg
Chair of Information Systems
esp. Industrial Application Systems
Feldkirchenstraße 21
D-96052 Bamberg
Germany
christian.suchan@uni-bamberg.de

ABSTRACT

System dynamics (SD) is a well-known methodology for analyzing the structure and the behavior of business systems and the environment. The utility of SD has been proven in numerous cases. However, decision makers have reservations using SD to support their decision processes. The reasons can be a) decision makers have problems to design a causal model under given time constraints (aspect of methodology complexity and methodical support) and b) the constructed causal and flow model do not offer adequate structure and behavior consistency (aspect of model quality). Furthermore, there are high demands on the causal and flow model due to using it for the description of the behavior of the real system in terms of differential equations. This paper introduces causal coupling patterns using the task concept of KOSIOL (cf. Kosiol, 1976) to simplify causal modeling and to derive the corresponding flow models in order to increase the acceptance and the model quality of SD models.

Keywords

causal modeling, design patterns, flow modeling, system dynamics, task concept

INTRODUCTION

System dynamics (SD) (cf. Forrester, 1961; Sterman, 2000) is a methodology for analyzing the structure and the behavior of business systems and the environment for almost 50 years. The utility of SD has been proven in numerous cases (e. g. Cooke, 2003; Meadows, Meadows and Randers, 2004 and Strohhecker, 2005). However, decision makers have reservations using SD in their decision processes. This statement is supported by the results of a survey conducted by the author of the 500 largest German companies (n = 370, response rate = 11.34 %, total revenue > 1.5 Bill. €) from September to October 2008. One of the research questions was the analysis of the diffusion rate of simulation methods within scenario-based procedures as part of strategic planning. Simulation techniques such as SD were used by none of the respondents. In contrast, scenario based procedures are a major application field of the system dynamics methodology (SDM) (Höhnerloh, 1997). Although strategic planning has highest relevance to the company's success (as stated by the interviewed managers), managers spend only 8.6 % of their total working time for this task complex (Mintzberg, 1975; Reichwald and Goecke, 1996).

Corresponding to these results, we presume that the reservations of decision makers arise from high expenditure of time to apply the SDM, in order to achieve a high model quality of causal and flow models (structure and behavior consistency of the model to the real system). On the one hand this can be explained by the high complexity of the SDM and lack of methodological support, resulting in high requirements in education and experience in creating SD models. On the other hand the resulting models run the risk to exhibit less model completeness and model quality. In consequence, there is a need to improve the SDM itself.

As a starting point for SDM improvement, an experiment was conducted (in January 2009; n = 22) with members of a university course in management support systems. The average skills (theoretical and practical) of the test subjects in the

SDM were 2.83[1] and they offer an average cognizance time of SD of 2-5 weeks. In this experiment they were asked about their problems using the SDM focusing on problems in constructing causal and flow models. The test subjects rated their problems in causal modeling from 1 (no problems) to 5 (severe problems) as follows (cf. Figure 1).

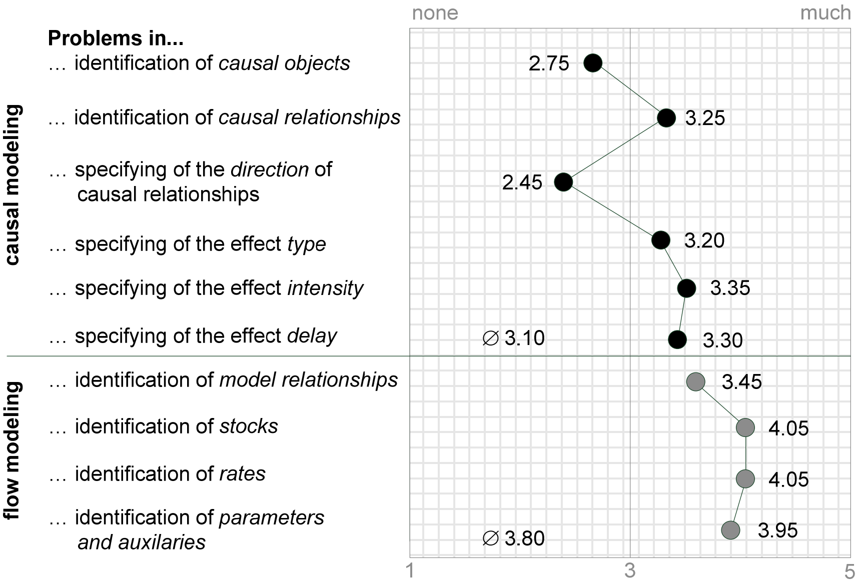

Figure 1. Problem Fields in Construction of Causal Models and Flow Models

In conclusion, the test subjects have significant more problems in flow modeling (avg. 3.80) than in causal modeling (avg. 3.10). This can be founded in more complexity of the flow meta-model and the missing relationship meta-model (RMM) between the causal and the flow meta-model. Unpleasant, the literature only offers a few *heuristics* to transfer a causal model in a flow model (e. g. Senge, Roberts and Smith, 1994 or the “snap shot test” by Sterman, 2000).

Due to the preceding discussion we define the *analysis hypothesis (AH): A subject modeling causal models and flow models needs the support of an artifact, which helps the modeling subject to construct a causal model and to derivate a corresponding flow model. Aim of the artifact is higher model completeness as well as structure and behavior consistency of the constructed models to the real system.*

This paper addresses the analysis hypothesis developing construction hypothesis. It offers a proposal to the AH in form of an artifact consisting of a catalog of causal coupling patterns (CCP) and a RMM. Finally, the adaptability of the artifact is exemplified by an application of the CCP to the world model of MEADOWS, MEADOWS and RANDERS (Meadows, Meadows, and Randers, 2004). The research method used in this paper is designed based on the *radical constructivism* of VON GLASERSFELD (e. g. Glasersfeld, 2002) and *critical rationalism* of POPPER (e. g. Popper, 2004). Furthermore, it is verified according to the guidelines of HEVNER/MARCH/PARK/RAM (Hevner, March, Park and Ram, 2004).

MODELTHEORETICAL ASPECTS OF THE CONSTRUCTION HYPOTHESES

System Dynamics from a Radical Constructivistical and Model Theoretical Point of View

Initial point of the epistemological interpretation of the SD methodology (cf. Figure 2) is the *real world*. A *subject perceives* and *interprets* the real world as a *system* with system components and system relationships (*system theoretical view*). The subject has a *contextual relationship* with the real world. On this foundation the subject constructs (1st) a causal model considering *modeling aims*. (2nd) The subject transforms the causal model to a flow model. Both models map predominant the *structure* of the real system. Furthermore (3rd), an equation model corresponding to the flow model is specified, which maps predominant the *behavior* of the real system. Between the real system and the models *model mapping relationships* exist (Ferstl and Sinz, 2008). In order to derivate construction hypotheses, we analyze problems of the model process.

---

[1] The rating scale 1 (worst) up to 5 (best) was used.

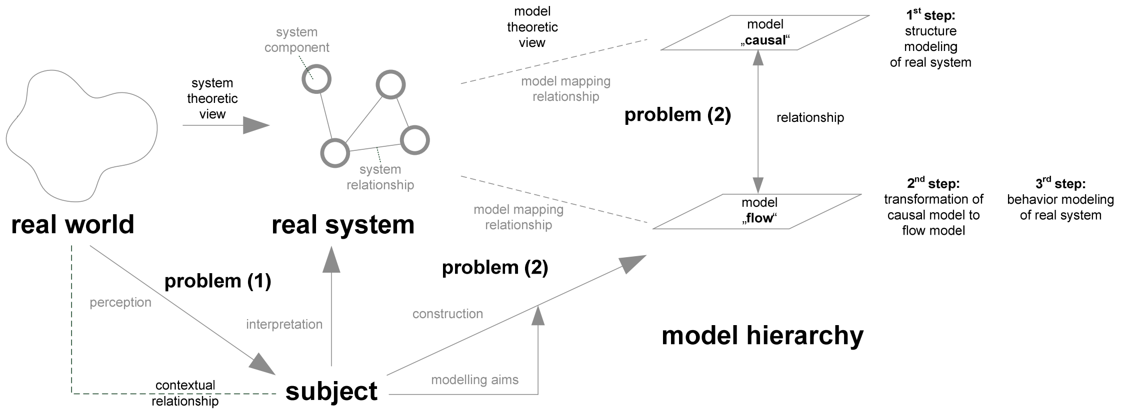

Figure 2. Radical Constructivistical and Model Theoretical Interpretation of the System Dynamics Methodology (cf. Ferstl and Sinz, 2008)

Problem (1): The perception of the real world often exhibits a *noise*. This noise will occur if a description of the real world or the subject’s perception of the real world is incomplete or disturbed. Subjects compensate this noise using experience and knowledge about the real world. However, this is error-prone and would lead to inconsistent structure and behavior consistency of the causal and flow model to the real system.

Problem (2): The subject constructs a causal and a flow model of the real system in considering modeling goals. If the problem (1) takes an effect on the construction of the causal model, it will be incomplete having less structure and behavior consistency. If the mapping of the causal model to the flow model is not *surjective* or *bijective*, the subject will have to add model components to the flow model, which is error-prone. Furthermore, because there is a lack of typification of causal objects, it is very difficult for a subject to assign a causal object to a corresponding flow object (*rate*, *stock*, *parameter* or *auxiliary*). You can only decide by means of the name of the causal object, which represents a (semantic) concept. However, there are only a few rules for naming the causal objects (Sterman, 2000), which can lead to vagueness (Black, 1937).

In order to tackle these problems, we provide the following construction ideas:

- **Patterns**: Patterns define rules about the construction of models to decrease the variety (Malik, 2008) of a subject and simplify the construction of models (vom Brocke, 2003). Furthermore, a semantic annotation regarding the structure will be given (Alexander, Ishikawa, Silverstein and Jacobson, 1977). Patterns help to secure the model completeness and reduce the modeling complexity.

- **Relationship meta-model**: The causal and flow model have to comply with a causal and flow meta-model[2], which defines the allowable model components and the allowable relationships. Between meta-models you can define a RMM, which specifies the mapping from one model component of one meta-model to one model component of the other meta-model (Sinz, 2002). Still, no RMM can be constructed, because there is no further *typification* of causal objects available allowing a *surjective* or *bijective* mapping between a model component of the causal model and the flow model. To enable this, we need an additional metaphor (Black, 1954) for the typification of causal objects.

Construction hypothesis (CH_1): An artifact to AH has to include patterns to be used by the subject during the modeling process of causal and flow models. Furthermore, a RMM between the causal and the flow meta-model is to be constructed. This will help to increase the model completeness and the structure and behavior consistency.

---

[2] The meta-models of the causal model and the flow model are available. In case of interest, please contact the author.

The Task Concept as an Additive to Develop Patterns

The main construction idea is the use of the *task concept* and the *task actor* concept of KOSIOL (Kosiol, 1976; further Ferstl and Sinz, 2008) as a metaphor for typification, which are strictly conform to the *system theoretic view* of the business system and the environment.

The task concept (cf. Figure 3) can be introduced as *inside* and *outside* view. The inside view contains the action control and the actions, which are in relationship of a feedback control. The *action control* (AC) initiates the *actions* (A) using the *steering relationship* (st) and receives the results of the action execution in using the *control relationship* (c). The actions operate on the *task object* (TO) with the *actor* (a) and the *sensor* (s) *relationship*. The TO represents defined attributes of a real system. Action control and the actions are part of the *procedure* (P) of a task. The outside view contains a *goal* (G), one or more *objectives* (O), one or more *pre-events* (PrE) and *post-events* (PoE) as well as the task object. The incidence of one or more pre-events of a task activates the action control to control the actions. After the execution of the procedure one or more post-events occur. A task is parameterized by a goal (G) specifying the type and number of the task output and one or more objectives (O) specifying the expected goal achievement. The relationship between inside and outside view of a task can a) directly coupled (with feedback) or b) not directly coupled (no feedback) (Ferstl and Sinz, 2008).

With regard to the patterns, the structure of the task defines the allowed relationships between components of the patterns. Every task component corresponds to one causal object (*bijective* mapping). The inside view of the task can be regarded as a) *monolithic* or b) *decomposable*. In case a) a causal object represents a procedure, which is not further detailed. In case b) the procedure is decomposed in an action control and the actions. The action control gets the feedback of the actual state of the task object and decides the following actions to be executed based on the actual state of the task object.

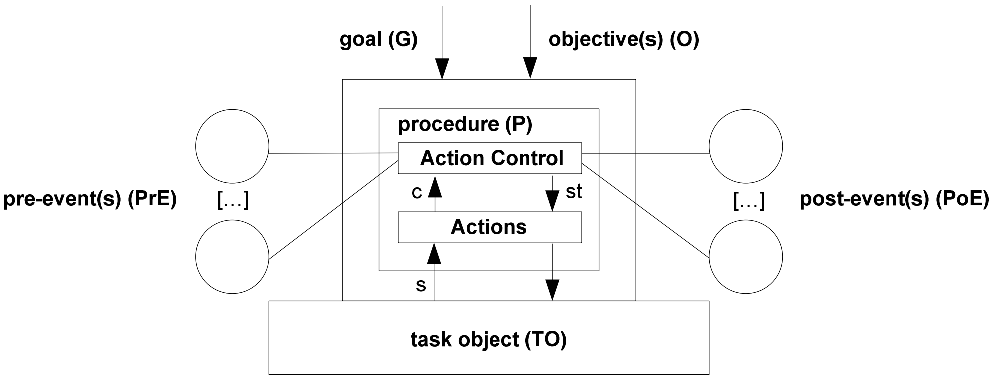

Figure 3. Structure of a Task (Task Inherent View) (Ferstl and Sinz, 2008)

In business systems the entire task can be decomposed in *multiple* tasks with relationships with each other. Besides, aspects of coupling of tasks have to be considered. The coupling of tasks can be viewed in analogy to the coupling of processors (Ferstl and Sinz, 2008; Tanenbaum and van Steen, 2008). In a case of two or more processors operating on the same memory, this is a *close coupling*. If two or more processors interact with messages in using a communication system (e. g. a bus system), this is a *loose coupling*. According to these concepts, you can identify three different types of task coupling (cf. Figure 4). Type a) refers to an *identical post-event of task A with a pre-event of task B*. If the post-event of task A occurs, simultaneous the pre-event of task B occurs and activates the procedure execution (loose coupling). In case of type b) *parts of the procedure or the whole procedure will be used by two or more tasks* (close coupling). This type of task coupling determines a semantic integrity constraint concerning the identity of the joint used procedures and eventually selected joint used task object instances. In case of type c) *task object types or task object instances are used by two or more tasks jointly*. This type of task coupling defines a semantic integrity constraint concerning the identity of the selected types of task object or the identity of selected task object instances (close coupling) (Mantel, Eckert, Schissler, Schäffner, Ferstl and Sinz, 2004).

Construction hypothesis (CH_2)*: A model driven typification of causal objects using the task concept improves the model completeness and simplifies the construction of patterns. Furthermore, it allows at least a surjective mapping of causal model components to flow model components.*

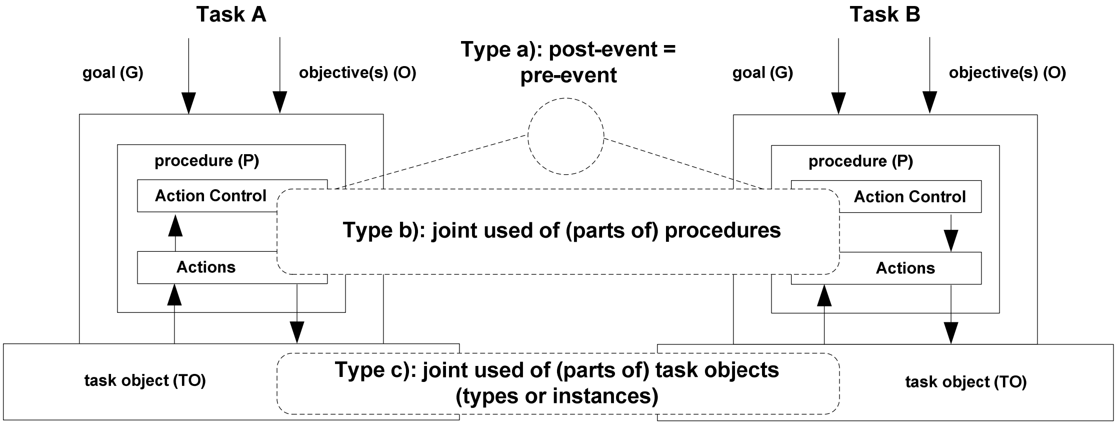

Figure 4. Coupling Variants of Multiple Tasks (Ferstl and Sinz, 2008)

The task actor concept refers to one or more persons as well as one or more machines executing a task (task to actor-relationship) (Ferstl and Sinz, 2008). Aspects of a task actor will be modeled in SD in the sense of a) a *capacity* and b) an *influenced object*. In case a) a task actor is modeled as *parameters (constants)* or *auxiliaries*, which influences *rates* via an information relationship. Otherwise (case b)) persons or machines are represented by a *stock*, which means that in sense of the task actor concept they do not *active* execute a task. By contrast, persons or machines are influenced (*passive*) and correspond to a task objective. In terms of the task concept a parameter or an auxiliary complies with a pre-event of a task.

Construction hypothesis (CH_3): Aspects of task actors (persons or machines) do not need to be mapped in a model component of a pattern. Because of the possibility to model task actor aspects with other model components, this has no influence to the model completeness as well as to the structure and the behavior consistency.

DESIGN OF THE CAUSAL COUPLING PATTERNS

The construction hypothesis $CH_{1,}$ CH_2 and CH_3 are realized in the artifact of the *causal coupling patterns (CCP)* and a *RRM*, which are arranged by the *model architecture frame* (Sinz, 2002) (cf. Figure 5). The causal meta-model is extended by the task concept. The causal objects are typified, which leads to a *goal*, *objective*, (*pre-* and *post-*)*event(s)*, *procedure (rather action control and actions)* and *task object* specified causal object. The CCP will be annotated to the causal meta-model, whereas the flow meta-model is left unchanged. Between the causal meta-model and the flow meta-model a RMM will be specified. A pattern consists of a *pattern core*, which means that the components are obligatory. Components of the *pattern shell* are optional.

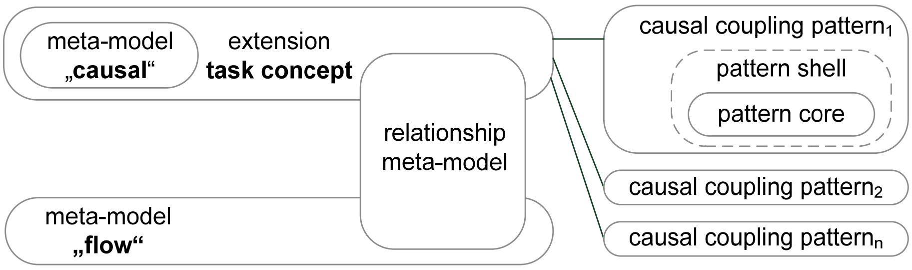

Figure 5. Model Architecture Frame

Regarding to the different perspectives on a single task (with variants procedure *monolithic* or *decomposable, with* or *without direct feedback* from task object to procedure) and multiple tasks, we distinguish between *task inherent view* and a *task comprehensive* CCP. This leads to one initial task inherent pattern and three task comprehensive patterns as well as their variants. The CCP will be introduced in (1st) a description of the pattern components and pattern relationships, (2nd) a description of the corresponding flow model components, (3rd) possibilities of modeling SD-Patterns and (4th) an example.

The task inherent CCP type a' represents the structure of one task (cf. Figure 6). The outside view contains the goal (G), objective (O), one or more pre-events (PrE) or post-events (PoE)[3] and a task object (TO). The inside view of a task includes the procedure (P). In a flow model the goal and the objective have to be modeled as a *parameter* or an *auxiliary*. The procedure represents the *rate*, which manipulates the task object in the shape of a *stock*. Pre-events and post-events can be modeled as *parameters* or *auxiliaries*. It is imperative for all CCP that objective(s), pre-event(s) and post-event(s) are optional. This CCP refers to a simple *inflow* or *outflow system* and the SD-Pattern *positive feedback loop*.

For example, a real system in form of an extract of a corporate finance department is modeled. The goal (G) is to invest 10 % of the 'profit' in the company considering the objective (O) of maximizing the 'company's goodwill'. The procedure (P) 'investing' increases the 'company's goodwill'. Pre-event (PrE) of 'investing' is the 'profit' of 1.000.000 $, whereas post-event (PoE) results in a 'staff requirement' of 1.000 persons.

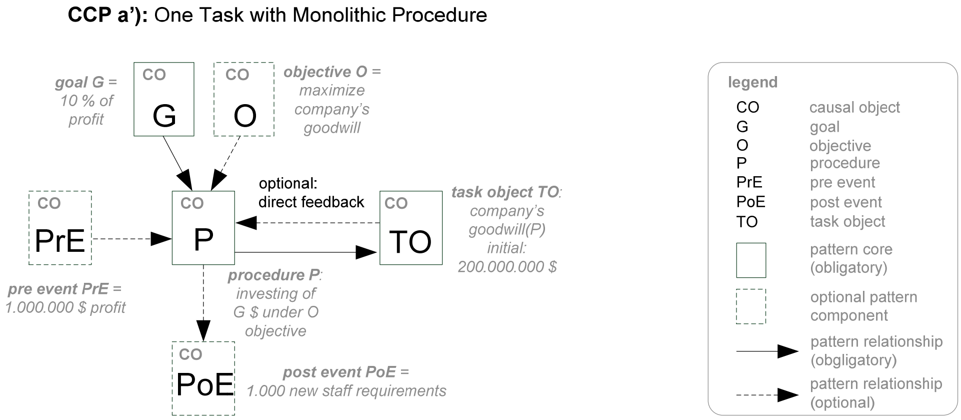

Figure 6. Task Inherent Causal Coupling Pattern Variant with a Monolithic Procedure

If a procedure will be decomposed (cf. Figure 7), the interaction between the action control and the actions will be cognizable (CCP type a''). This allows a more detailed modeling of the structure and the behavior of the real system. The flow model is similar to the CCP type a'. Instead of this, the action control as an *auxiliary* parameterizes the action as a *rate*. The rate manipulates the task object in form of a *stock*. This CCP corresponds to the SD-Pattern *negative feedback loop*.

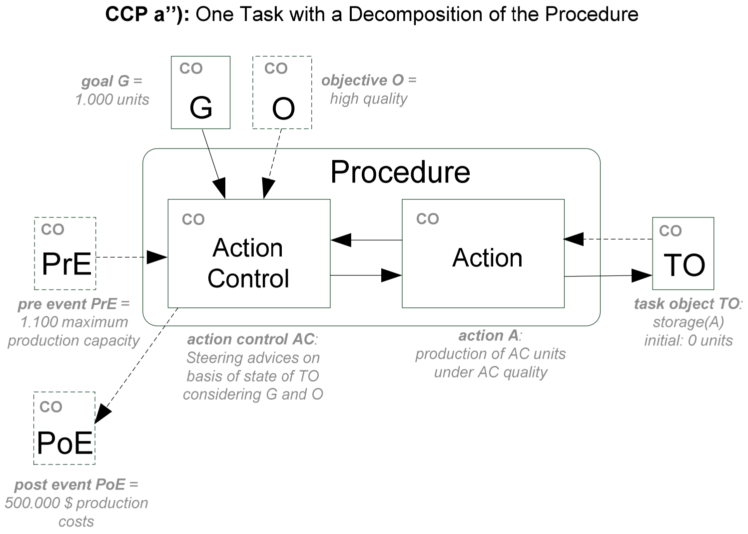

Figure 7. Task Inherent Causal Coupling Pattern Variant with a Decomposition of the Procedure

---

[3] Please note that one or more pre-events or post-events can occur.

For example, a real system of an industrial production is modeled. The action control is parameterized by the goal (G) to produce 1.000 units of a product. The objective (O) forces that the product should be in high 'quality'. Further a 'maximum production capacity' of 1.100 units per time restricts the output of the production. The action control triggers the action with steering commands. The action 'production' produces units in interacting with the task object (TO) 'storage' in form of manipulating the attributes of the task object. The action reports the quantity and quality of the produced units to the action control. Finally, the action control decides on the feedback information in which way the steering commands have to be changed.

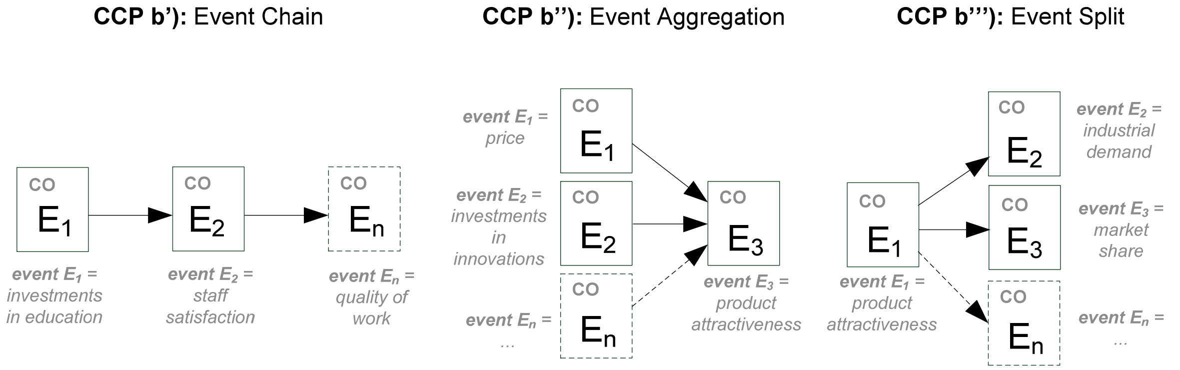

Figure 8. Task Comprehensive Causal Coupling Patterns: Event Sequences

In the CCP type b of the *event sequences* two or more events are in relationship with each other.[4] An event chain (CCP b') contains two or more events, which each have one single relationship from one event to another (cf. Figure 8, left part). For *n* events there are only *n-1* relationships between events, whereas an event has a maximum of one *inbound* and one *outbound* relationship. An event aggregation (CCP b'') enfolds at least three or more events and a minimum of two relationships. Two or more events (*active* causal objects) influence one other event (a *passive* causal object). Every active causal object has one outbound causal relationship and the passive causal object *n* inbound relationships, whereas *n* is the number of active causal objects in the pattern (cf. Figure 8, middle part). An event split (CCP b''') has one event (an active causal objects) parted in at least two or more other events (passive causal objects). Every active causal object has *n* outbound causal relationships and the passive causal objects one inbound causal relationship, whereas *n* is the number of passive causal objects in the pattern (cf. Figure 8, right part). This pattern uses the *task coupling type a*. In flow models, the causal objects of these patterns can be represented by *parameters* or *auxiliaries*. With a combination of the CCP type a'' and the CCP type b' the SD-Pattern *s-shaped growth*, *s-shaped growth with overshoot* as well as *s-shaped growth with overshoot and collapse* can be modeled.

For example, in case of CCP b' 'investments in education' of the staff of a company influence the 'staff satisfaction'. Consequently, the 'staff satisfaction' will affect the 'quality of work'. Furthermore, in case of CCP b'' increasing 'prices' have a decreasing 'product attractiveness' effect, whereas increasing 'investments in innovations' lead to increasing 'product attractiveness'. Finally, the' product attractiveness' has a positive effect on 'industrial demand' and 'market share' (CCP b''').

The pattern core of the CCP type c contains two procedures P_1 and P_2, which operate on the same task object (TO) (cf. Figure 9). The procedures are parameterized by the goals G_1 and G_2 as well as the objectives O_1 and O_2. Both procedures may have pre- and post-events. This pattern uses the *task coupling type c*. A joint use of task object types does not occur within the CCP. In a flow model the procedures represent the *rates*, which manipulate the *stock*. The stock is the task object of the procedures. Pre- and post-events are represented by auxiliaries and parameters. This CCP is designed in analogy to the SD-Pattern *inflow-outflow-model* with two rates and one stock.

For example, we reconsider a real system of an industrial production. The procedure 'production' (P_1) interacts with the task object 'storage' (TO) and increases the 'storage' by 'production' of an amount of units of a car. The number of units is forced by the goal (G_1) of 1.000 units of a car. Furthermore, the produced units should have high 'quality' (objective O_1). The procedure 'production' can only produce a maximum of 1.100 units. The production causes 1.000.000 $ of 'production costs'. The produced units of a car are placed in the 'storage'. The procedure 'distribution' decreases the number of units in the 'storage' throughout delivering the cars. The distribution of 1.000 units (goal G_2) should be processed with minimum

---

[4] Please note that the concept of *event* is a generalization of the *pre-* and *post-events*.

‘costs’ (objective O_2) and is limited by the ‘distribution capacity’ of 500 cars per given time. After the distribution there are ‘distribution costs’ of 25.000 \$.

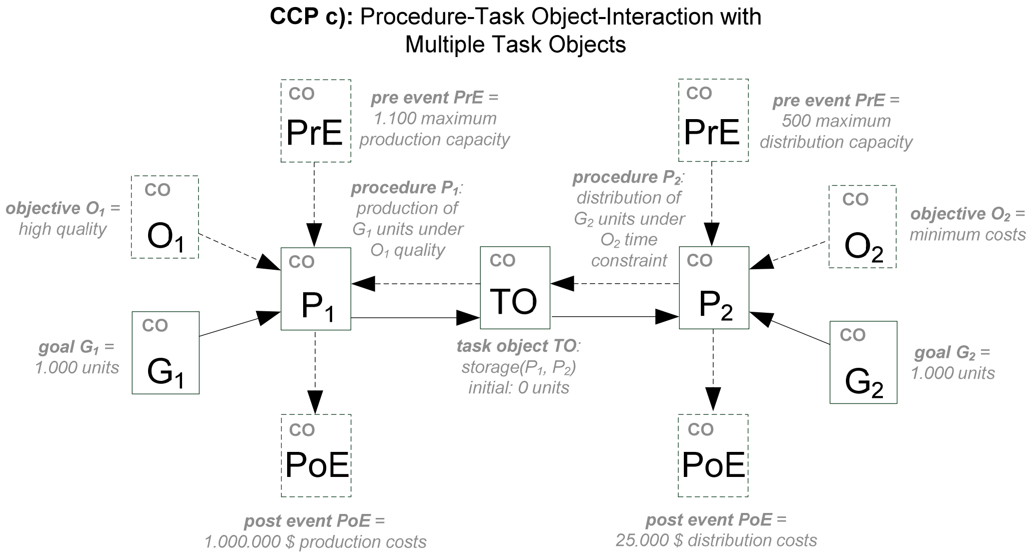

Figure 9. Task Comprehensive Causal Coupling Patterns: Procedure Task Object Interaction with Multiple Procedures

The CCP type d consists of two task objects (TO_1 and TO_2), a procedure (P) and a goal (G) as the pattern core (cf. Figure 10). The procedure P interacts with the two task objects (TO_1 and TO_2) and manipulates attributes of the two task objects. Supplemental, one or more pre- or post-events (PrE and PoE) as well as one or more objectives (O) are part of the pattern shell. This pattern uses the *task coupling type b*. In the flow model the procedure (P) represents a rate and is connected with two stocks (the task objects TO_1 and TO_2). The goal (G), objective (O), pre-event(s) (PrE) and post-event(s) (PoE) can be modeled in using *parameters* and *auxiliaries*. This causal coupling pattern is designed in analogy to an *aging chain* SD model.

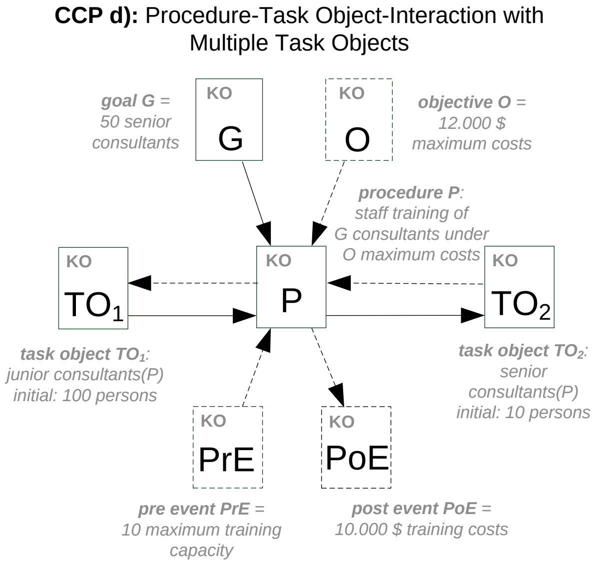

Figure 10. Task Comprehensive Causal Coupling Patterns: Procedure Task Object Interaction with Multiple Task Objects

For example, a real system in form of a human resource management department of a management consulting is modeled. The whole staff is divided into ‘junior consultants’ (TO_1), who are hired directly after the master degree and ‘senior *consultants' (TO_2), who must have more experience than the 'junior consultants'. To increase the number of 'senior consultants', a 'staff-training' will be initiated. This is represented by the procedure (P) connected with two task objects (TO_1 and TO_2). The number of 50 'senior consultants' has to be reached (goal G), so in dependency of the initial value of 10 'senior consultants' in sum 40 'junior consultants' should be trained. The 'staff training' is limited by the objective (O) in form of 12.000 $ maximum 'costs' and the maximum 'training capacity' of 10 consultants per training. In conclusion, four times a staff training of 10 consultants takes place to reach the number of 50 'senior consultants'.*

Relationship meta-model

The RMM (cf. Figure 11) consists of an extract of the causal meta-model and the flow meta-model. The causal objects are typified by using the task concept. The flow meta-model is left unchanged. The relationships consist of cardinalities (*minimum*, *maximum*) as integrity constraints (*surjective* mapping). An '*aim'* or an '*event*' *typified* causal object can be represented by an '*auxiliary'* or a '*parameter'* flow object. An '*action control'* causal object is an *auxiliary*, whereas '*action'* and '*procedure'* causal objects are represented by a '*rate'* flow object (all objects *without system state memory*). A '*task object'* causal object is represented by a *stock* (object *with system state memory*). The causal relationship between a procedure or actions and the task object corresponds to a *flow* interaction relationship. For all other cases the causal relationships are transformed to an *information interaction relationship*.

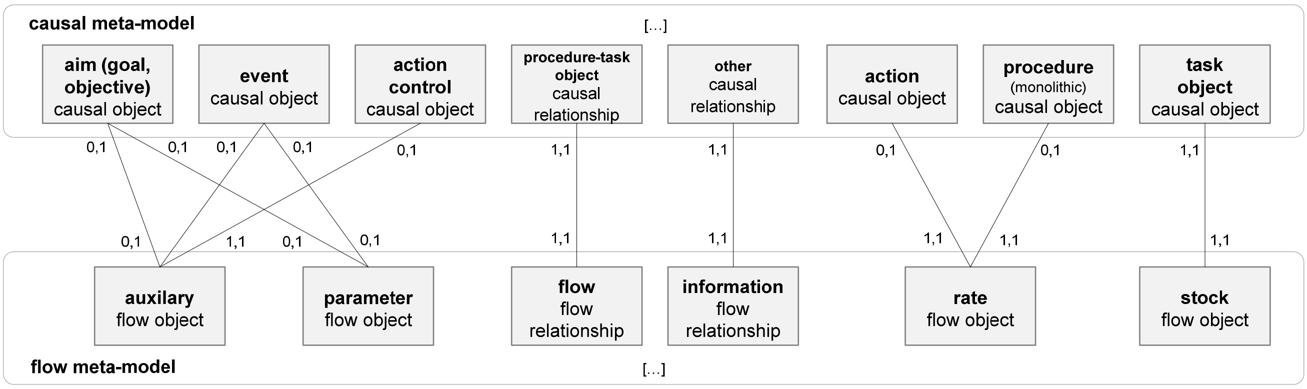

Figure 11. Relationship Meta-Model between the Causal Meta Model and the Flow Meta-Model

APPLICATION OF THE CAUSAL COUPLING PATTERNS EXEMPLIFIED BY THE WORLD MODEL

Following, the CCP will be applied to the world model (cf. Figure 12) by MEADOWS/MEADOWS/RANDERS (cf. Meadows et al., 2004). The loops L_1 and L_2 represent the development of the world population. The *births per year* increase in analogy to a procedure (P_{A1}) the *population*, which represents a task object (TO_A). Alike the *deaths per year* decrease (P_{A2}) the *population* (TO_A). P_{A1} and P_{A2} are parameterized by the goals *fertility* (G_{A1}) and *mortality* (G_{A2}). These objects belong to the CCP_A type c. The causal objects *education and family planning*, *services per person* and *health services* correspond to the events EB_2, $EB_1$and EB_3 (CCP_B type b'''). The *non renewable resources* (TO_{C1}) and *service capital* (TO_{C2}) as task objects are manipulated by the *industrial output* as a procedure (P_C). An increase of the *industrial output* decreases the non-renewable resources and simultaneous increases the *service capital*. Pre-event of the procedure (P_C) is the *industrial capital* (PrE_C) and the *industrial output per person* represents a post-event (PoE_C). These objects are part of the CCP_C type d. The loops L_3 and L_4 represent the industrial capital of the world economy. The *investments* increase in analogy to a procedure (P_{D1}) the *industrial capital*, which represents a task object (TO_D). The deprecation decreases (P_{D2}) the *industrial capital* (TO_D). P_{D1} and P_{D2} are parameterized by the goals *investment rate* (G_{D1}) and *average lifetime per capital* (G_{D2}). These objects belong to the CCP_D type c. Under considering the typification of the CCP and the RMM (cf. Figure 11) you can derivate a corresponding flow model without addiction of model components.

CONCLUSION AND FURTHER RESEARCH

In this paper we developed a catalog of CCP and a RMM to solve the introduced problems. Using the example of the world model we have shown that the CCP are applicable. Furthermore, a simple transfer of the causal objects of the world model to a flow model is possible. The artifact offers the potential of model completeness and high structure and behavior consistency of resulting causal and flow models.

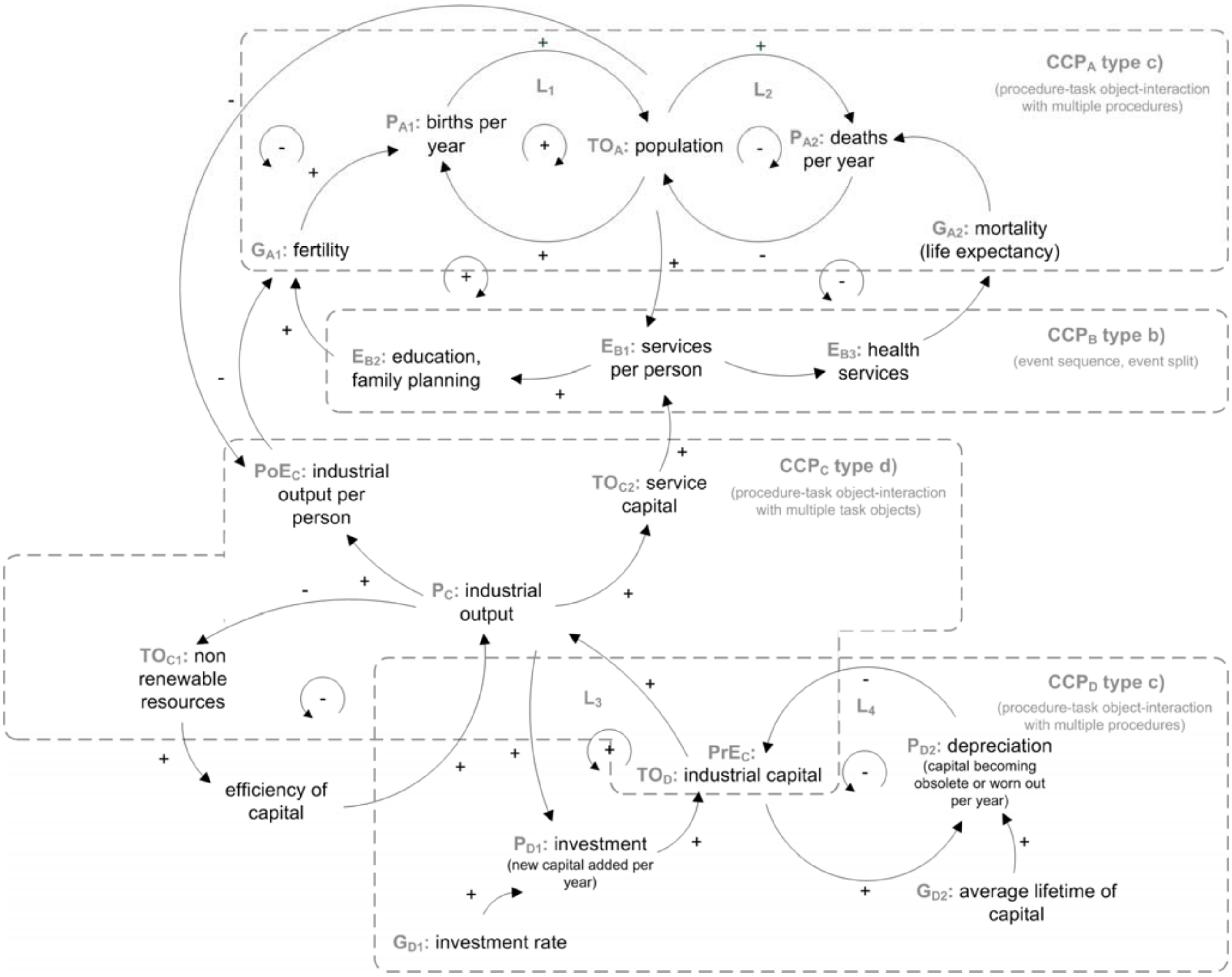

Figure 12. Identification of Causal Coupling Patterns Exemplified by the World Model

Regarding the experiment introduced at the beginning of this paper, the test subjects (knowledge about the task concept and task actor concept with an avg. of 3.58)[5] were further asked about their guess of the relationship between the components of the task concept as well as the task actor concept and the components of a flow model. The test subjects *intuitively* assign the *goals* correctly with 83 % (avg.), *objectives* with 84 %, the *procedure* with 68 %, the *task object* with 83 %, the *events* (pre- or post-events) with 26 %, and the *task actor* with 90 % in the sense of the RMM. These results imply traceability and comprehensibility of the artifact. In conclusion, we can declare that the *construction hypotheses* CH_1, CH_2 *and* CH_3 *are temporarily approved*.

Nevertheless, the CCP and RMM add more complexity to the SDM. This complexity must be reduced by using a software tool to abstract from the introduced concepts and simplify the application of the CCP. Furthermore, the presented CCP will be part of a comprehensive approach to extend the SDM to simplify the modeling process and allow considering qualitative and quantitative variables within SD models. Finally, the CCP will be reviewed in an extensive case study in cooperation of a company to identify additional improvements.

---

[5] The scale 1 (worst) up to 5 (best) was used.

REFERENCES

1. Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M. (1977) A pattern language. Oxford University Press, New York.
2. Black, M. (1937) Vagueness: An exercise in logical analysis, *Philosophy of Science*, 4, 4, 427-455.
3. Black, M. (1954) The metaphor, *Proceedings of the Aristolian Scoiety*, 55, no issue, 273-294.
4. Cooke D. L. (2003) A system dynamics analysis of the Westray mine disaster, *System Dynamics Review*, 19, 2, 139-166.
5. Ferstl, O. K. and Sinz, E. J. (2008): Grundlagen der Wirtschaftsinformatik [Fundamentals of information systems, in German], 6th Edition, Oldenbourg-Verlag, Munich.
6. Forrester, W. (1961) Industrial dynamics, Wiley, Cambridge.
7. Glasersfeld, E. V. (2002) Radical constructivsm: A way of knowing and learning, Falmer Press, London.
8. Hevner, A. R., March, S. T., Park, J., Ram, S. (2004) Design science in information systems research, *MIS Quarterly*, 28, 1, 75-105.
9. Höhnerloh, A. (1997) Unscharfe Simulation in der Betriebswirtschaft [Fuzzy simulation in the business administration, in German], unitext-Verlag, Göttingen.
10. Kosiol, E. (1976) Organisation der Unternehmung [Organization of the enterprise, in German], 2nd Edition, Gabler-Verlag, Wiesbaden.
11. Malik, F. (2006) Management performing living, Campus-Verlag, Frankfurt am Main.
12. Mantel, St., Eckert, S., Schissler, M., Schäffner, C., Ferstl, O. K. and Sinz, E. J. (2004) Eine Entwicklungsmethodik für die betriebliche Integration von Anwendungssystemen [A developement methodology supporting the integration of application systems, in German], in Bartmann, D. and Mertens, P. (Eds.) *Überbetriebliche Integration von Anwendungssystemen [Intercompany integration of application systems, in German]*, Shaker-Verlag, Aachen, 21-29.
13. Meadows, D. H., Meadows D. and Randers J. (2004) Limits to growth: the 30-year update, Chelsea Green Publishing, Vermont.
14. Mintzberg, H. (1975) The manager’s job: folklore and fact, *Harvard Business Review*, 53, 4, 49-61.
15. Popper, K. (2004) The logic of scientific discovery, Taylor & Francis, London.
16. Reichwald, R. and Goecke, R. (1997) Telekommunikation im Management [Telecommunication in the management, in German], Schäffer-Poeschel-Verlag, Stuttgart.
17. Senge, P., Roberts, Smith (1994) The fifth discipline fieldbook: strategies and tools for building learning organisations, Bantam Dell, New York.
18. Sinz, E. J. (2002) Architektur von Informationssystemen [Architecture of information systems, in German], in: Rechenberger, P., Pomberger, G. (Eds.): Informatik-Handbuch, Hanser-Verlag, Munich, 1055-1067.
19. Sterman, J. D. (2000) Business dynamics: systems thinking and modeling for a complex world, McGraw-Hill, Boston.
20. Strohhecker, J, (2005) Scenarios and simulations for planning the Dresdner Bank’s E-day, *System Dynamics Review*, 21, 1, 5-32.
21. Tanenbaum, A. S. and Steen M. van (2008) Distributed systems: principles and paradigms, 2nd edition, Prentice-Hall International, Upper Saddle River.
22. Vom Brocke, J. (2003) Referenzmodellierung: Gestaltung und Verteilung von Konstruktionsprozessen [Reference modeling: design and distribution of construction processes, in German], Logos-Verlag, Berlin.