

Analyzing the Dynamic Cost Factors of Process-Aware Information Systems: A Model-Based Approach

Bela Mutschler^{1,2}, Manfred Reichert², and Stefanie Rinderle³

¹ DaimlerChrysler Group Research, P.O. Box 2360, 89013 Ulm, Germany
bela.mutschler@daimlerchrysler.com

² Information Systems Group, University of Twente, The Netherlands
m.u.reichert@utwente.nl

³ Institute Databases and Information Systems, University of Ulm, Germany
stefanie.rinderle@uni-ulm.de

Abstract. Introducing *process-aware information systems* (PAIS) in enterprises is usually associated with high costs. It is therefore crucial to understand those factors that determine these costs. Though software cost estimation has received considerable attention during the last decades, it is difficult to apply existing approaches to PAIS. This difficulty particularly stems from the inability of these techniques to deal with the dynamic interactions of the many technological, organizational and project-driven cost factors which specifically arise in the context of PAIS. Picking up this problem, this paper presents an approach to investigate the complex cost structures of PAIS engineering projects based on evaluation models. We present a formalism to design such evaluation models, discuss one characteristic evaluation model and its derivation in detail (based on the outcome of an empirical study), and introduce the notion of *value-based evaluation patterns* to enable the reuse of evaluation models.

1 Introduction

Process-aware information systems (PAIS) separate process logic from application code [1], and orchestrate the processes at run-time according to their defined logic [2]. For implementing PAIS, numerous process support paradigms (e.g., workflow management, service flows, case handling), process modeling standards (e.g., WS-BPEL, BPML), and tools (e.g., ARIS Toolset, Tibco Staffware) have been introduced [3].

While the benefits of PAIS are typically justified by improved process performance [4,5] and cheaper process implementation [6], there exist no approaches for systematically analyzing related costs. Though software cost estimation has received considerable attention during the last decades and has become an essential task in information system engineering, it is difficult to apply existing approaches to PAIS. This difficulty stems from the inability of these approaches to cope with the numerous technological, organizational and project-driven cost factors which have to be considered in the context of a PAIS (and which do only partly exist in data- or function-centered information systems) [7]. As an example consider the costs for redesigning processes. Another challenge deals with the dependencies between these factors. Activities for *business process redesign* [8], for example, can be influenced by intangible impact factors like *process*

knowledge or *end user fears*. These dependencies result in dynamic economic effects which influence the overall costs of a PAIS engineering project. Existing techniques [9] are typically not able to deal with such dynamic effects as they rely on static models based upon snapshots of the considered software system.

What is needed is a comprehensive approach that enables engineers to model and investigate the complex interplay between the cost and impact factors that arise in the context of PAIS. For this purpose, this paper¹ introduces a sophisticated and practically approved, model-based methodology to better understand and systematically investigate the complex cost structures of PAIS engineering projects. We present a formalism to design qualitative evaluation models and discuss one characteristic evaluation model and its derivation in detail (based on the outcome of an empirical study). In response to the problems observed during the exploratory use of our methodology in practice, we additionally introduce the notion of *value-based evaluation patterns*.

The remainder of this paper is organized as follows. Section 2 describes our qualitative cost analysis methodology. Section 3 introduces value-based evaluation patterns. Section 4 discusses related work, and Section 5 concludes with a summary.

2 The EcoPOST Cost Analysis Methodology

This section describes the main steps of our approach for modeling, analyzing and understanding those factors and their complex interplay that determine the dynamic costs of PAIS engineering projects. Section 2.1 describes the terminology used in our approach. Section 2.2 introduces the notation of our evaluation models. Section 2.3 gives an illustrating example. Section 2.4 motivates the use of simulation for analyzing the dynamic implications as described by our evaluation models. Section 2.5 deals with the specification of simulation models. Finally, Section 2.6 summarizes major lessons learned from a (pilot) case study in the automotive domain.

2.1 Terminology

Basically, we distinguish between different kinds of evaluation factors. *Static Cost Factors* (SCF) represent costs that can be precisely quantified in terms of money. The value of a SCF does not considerably change during a PAIS engineering project (except for its time value, which is not further considered in this paper). Thus, the value of a SCF can be considered as constant. As typical examples of SCF consider software license costs, hardware costs, or costs for external consultants.

Dynamic Cost Factors (DCF), in turn, represent costs that are determined by activities related to a PAIS engineering project. The (re)design of business processes prior to the introduction of PAIS, for example, constitutes such an activity. These activities cause measurable efforts, which, as they are influenced by other, often intangible factors, can vary. A DCF "Costs for Business Process Redesign", for instance, may be influenced by an intangible factor "Willingness of Staff Members to support Redesign

¹ Our research has been conducted in the EcoPOST project [7,10] which deals with the development of an evaluation framework for analyzing PAIS from a value-based perspective (see <http://is.ewi.utwente.nl/research/>).

Activities". Obviously, if staff members do not contribute to a redesign project by providing needed information (e.g., about process details), any redesign effort will be ineffective and will increase costs. If staff willingness is additionally varying during the redesign activity (e.g., due to a changing communication policy), the DCF "Costs for Business Process Redesign" will be subject to more complex effects. In our framework, intangible factors like "Willingness of Staff Members to support Redesign Activities" can be represented by so called *impact factors*.

Impact Factors (ImF) are intangible evaluation factors that influence DCF, i.e., the activities underlying a DCF. In particular, ImF lead to the evolution of DCF, which, in turn, makes the estimation and analysis of DCF a difficult task to accomplish. As examples consider factors such as "End User Fears", "Availability of Process Knowledge", or "Ability to redesign Business Processes". Opposed to SCF and DCF, the values of ImF are not quantified in monetary terms, but in a qualitative manner. "End User Fears", for example, can be quantified by means of a "Degree of End User Fears" (which can be "low" or "high"). Also, ImF can be either *static* or *dynamic*. The value of a static ImF (ImF_S) does not considerably evolve during a PAIS engineering project and can be considered as constant (like the value of a SCF). The value of a dynamic ImF (ImF_D), by contrast, may be changing. Like the evolution of DCF, the evolution of dynamic ImF is caused by (both static and dynamic) ImF.

2.2 Evaluation Models

To better understand the evolution of DCF in PAIS engineering projects as well as DCF interference through ImF, we utilize *evaluation models*. In particular, each DCF is represented and analyzed by exactly one evaluation model. These models are specified using the System Dynamics [11,12] notation² (cf. Fig. 1A) [7].

Model Notation. An evaluation model comprises a set of *model variables* which are denoted as *evaluation factors*. In our context SCF, DCF, and ImF correspond to evaluation factors. Different types of variables exist. *State variables* can be used to represent dynamic factors, i.e., to capture changing values of DCF (e.g., the "Costs for Business Process Redesign"; cf. Fig. 1B) and dynamic ImF (e.g., a certain degree of "Process Knowledge"). A state variable is graphically denoted as rectangle (cf. Fig. 1B), and its value at time t is determined by the accumulated changes of this variable from starting point t_0 to present moment t ($t > t_0$); similar to a bathtub which accumulates – at a defined moment t – the amount of water which has been poured into it in the past. Each state variable needs to be connected to at least one *source* or *sink*. Both sources and sinks are graphically denoted as cloud-like symbols (cf. Fig. 1B).

² We decided to use System Dynamics based on a literature review of potential modeling formalisms. Out of the investigated formalisms, *System Dynamics* (SD) and *Bayesian Networks* (BN) promised to be of particular usefulness in our context. Both formalisms allow to explicitly model networks of evaluation factors. However, BN deal with uncertainty and focus on determining probabilities of events. SD, by contrast, neglects the issue of "(un)certainty" and deals with the analysis of dynamic effects which occur in networks of interacting factors. As we can typically determine whether a certain factor is relevant in a given scenario, we decided to use SD.

Values of state variables change through inflows and outflows. Graphically, both flow types are depicted by twin-arrows which either point to (in the case of an inflow) or out of (in the case of an outflow) the state variable (cf. Fig. 1B). Picking up again the bathtub image, an *inflow* is a pipe that adds water to the bathtub, i.e., inflows increase the value of a state variable. An *outflow*, by contrast, is a pipe that purges water from the bathtub, i.e., outflows decrease the value of a state variable. The DCF "Costs for Business Process Redesign" shown in Fig. 1C, for example, increases through its inflow "Cost Increase" and decreases through its outflow "Cost Decrease".

Returning to the bathtub image, we further need "water taps" to control the amount of water flowing into the bathtub, and "drains" to specify the amount of water flowing out. For this purpose, a *rate variable* is assigned to each flow (graphically depicted by a valve; cf. Fig. 1B).

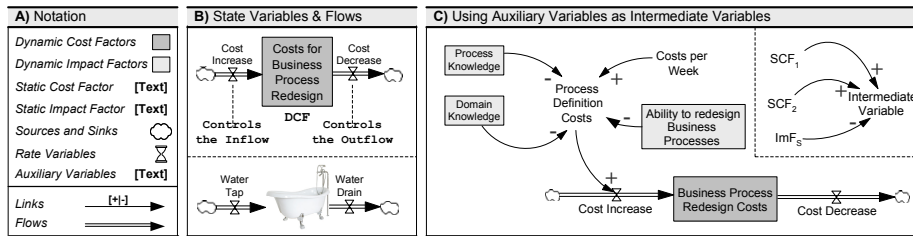


Fig. 1. Evaluation Model Notation and initial Examples

Besides state variables, evaluation models may comprise *constants* and *auxiliary variables* (which are both graphically represented by their name). Constants are used to represent static evaluation factors, i.e., SCF and static ImF in our context. Auxiliary variables, in turn, represent intermediate variables. As an example consider the auxiliary variable "Process Definition Costs" in Fig. 1C. Both constants and auxiliary variables are integrated into an evaluation model with *links* (not flows), i.e., labeled arrows. A *positive link* (labeled with "+") between x and y (with y as dependent variable) indicates that y will tend in the same direction if a change occurs in x. A *negative link* (labeled with "-") expresses that the dependent variable y will tend in the opposite direction if the value of x changes. Relationships as expressed by links either can be linear or non-linear (cf. Section 2.5 for details). Altogether, we can define:

Definition (Evaluation Model). A graph $EM = (V, F, L)$ is called evaluation model, if the following holds:

- $V := S \cup X \cup R \cup C \cup A$ is a set of model variables with
 - S is a set of state variables,
 - X is a set of sources and sinks,
 - R is a set of rate variables,
 - C is a set of constants,
 - A is a set of auxiliary variables,

- $F \subseteq ((S \times S) \cup (S \times X) \cup (X \times S))$ is a set of edges called flows,
- $L \subseteq ((S \times A \times Lab) \cup (S \times R \times Lab) \cup (A \times A \times Lab) \cup (A \times R \times Lab) \cup (C \times A \times Lab) \cup (C \times R \times Lab))$ is a set of edges called links with $Lab := \{+, -\}$ being the set of link labels, where
 - $(q_i, q_j, +) \in L$ with $q_i \in (S \cup A \cup C)$ and $q_j \in (A \cup R)$ denotes a positive link,
 - $(q_i, q_j, -) \in L$ with $q_i \in (S \cup A \cup C)$ and $q_j \in (A \cup R)$ denotes a negative link.

Model Correctness. For defining evaluation models we introduce additional constraints (*model design rules*) to be taken into account: (1) DCF and dynamic ImF have to be represented by state variables, (2) SCF and static ImF must be represented as constants, (3) every state variable v must be connected to at least one source or sink q , i.e., $\forall v \in S : \exists(q, v) \in F \vee \exists(v, q) \in F$ with $q \in X$, (4) every model variable must be used in at least one binary relation, i.e., $\forall v, q \in (S \cup X) : \exists(v, q) \in F \vee \exists(q, v) \in F$ and $\forall q \in (A \cup C) \wedge \forall v \in (A \cup R) : \exists(q, v, [+|-]) \in L$, (5) every rate variable of the evaluation model is influenced by at least one link, i.e., $\forall v \in R \wedge q \in (S \cup A \cup C) : \exists(q, v, [+|-]) \in L$, and (6) there exist no cycles solely consisting of auxiliary variables, i.e., $\neg \exists \langle q_0, q_1, \dots, q_r \rangle \in A^{r+1}$ with $q_0 = q_r$ and $q_k \neq q_l$ for $k, l = 1, \dots, r; k \neq l$.

Rules for the correct use of flows and links are illustrated in Fig. 2A and Fig. 2B. By contrast, Fig. 2C - Fig. 2H show examples of incorrect models. DCF and ImF_D , for example, may be only influenced by flows, and not by links as shown in Fig. 2C. Flows may be only connected to DCF and ImF_D , but not to auxiliary variables or constants as depicted in Fig. 2D. Links pointing to constants (e.g., SCF, ImF_S) as denoted in Fig. 2E and Fig. 2F are also not valid. Finally, flows and links connecting DCF to ImF_D (and vice versa) are also not considered as correct (cf. Fig. 2G and Fig. 2H).

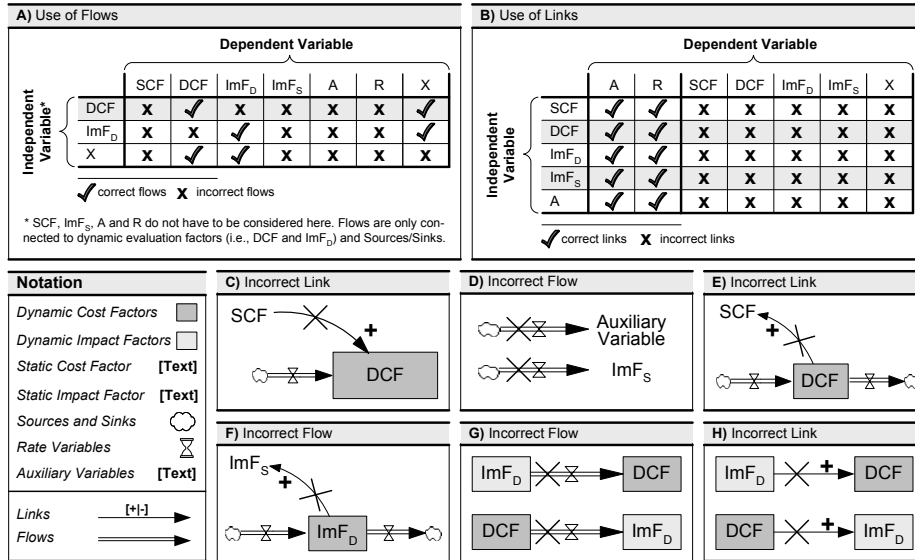


Fig. 2. Model Design Rules and Examples of Incorrect Modeling

Model correctness does not only presume compliance with existing model design rules. It also deals with the development of models that are suitable to represent real-world settings. Therefore, we accomplished user surveys and case studies (see below).

2.3 Illustrating Example

Fig. 3 shows a model which describes the influence of the dynamic ImF "End User Fears" on the DCF "Costs for Business Process Redesign". More specifically, this model reflects the assumption that the introduction of a PAIS may cause end user fears, e.g., due to a high degree of job redesign and due to changed social clues. Such end user fears can lead to emotional user resistance. This, in turn, can make it difficult to get support from the users while introducing a PAIS. Such models are of significant value for PAIS engineers, e.g., due to their suitability to serve as a conscious-raising tool about basic economic effects in PAIS engineering projects.

Model Details. Basic to this evaluation model is a cyclic structure connecting the four ImF "End User Fears", "Emotional Resistance", "Ability to acquire Process Knowledge", and "Ability to redesign Business Processes". Their arrangement (cf. Fig. 3) illustrates the following coherence: Increasing end user fears result in increased emotional resistance of end users. This dependency is represented by a positive link from the ImF "End User Fears" to the "Resistance Growth Rate" (which controls the inflow of the ImF "Emotional Resistance"). An increasing emotional resistance of end users, in turn, results in a decreasing ability to acquire process knowledge. Reason is that an increasing emotional resistance makes profound process analysis (e.g., based on interviews with process participants) a difficult task to accomplish. This dependency is represented by a negative link from the ImF "Emotional Resistance" to the rate variable "Decreasing Ability to acquire Process Knowledge" (which, in turn, controls the inflow of the ImF "Ability to acquire Process Knowledge").

A decreasing ability to acquire process knowledge results in a decreasing ability to redesign business processes. Again, this dependency is represented by a positive link. Finally, an increasing ability to redesign business processes can even enforce end user fears since end users often consider business process redesign activities as a potential

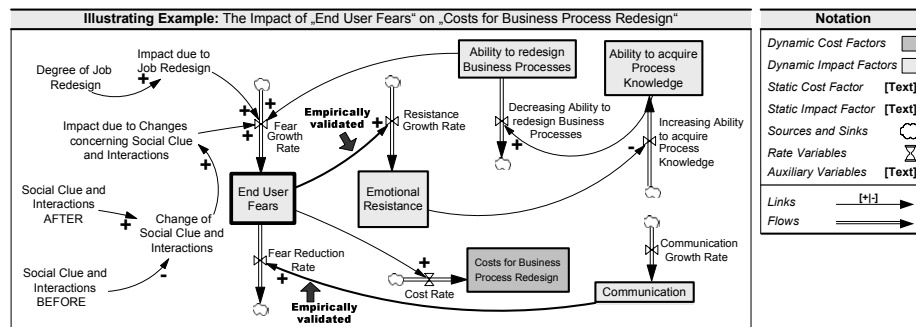


Fig. 3. Dealing with the Impact of End User Fears

threat for their own job. This dependency is represented by another positive link. Note that the "Fear Growth Rate" is not only biased by this link. It is also influenced by the "Degree of Job Redesign" and the "Change of Social Clue and Interactions" (which is calculated from the social clues and interactions before and after the business process redesign). Finally, "Communication" is considered as well. This ImF deals with the information of end users about the goals of introducing a PAIS.

Empirical Validation. To empirically confirm our assumptions as represented by this evaluation model (and the many other ones we have developed for PAIS engineers) we conducted an online survey³ among 70 business process management experts. Regarding the above example, we have analyzed the ImF "End User Fears" and the ImF "Communication" in more detail. First, we asked for the *relevance* of the factor (Question 1). Second, we asked whether there are potential *dependencies* between this factor and other ones (Question 2). Only those survey participants who confirmed the existence of dependencies were directed to two additional questions which dealt with the further specification of the confirmed dependency. Question 3 addressed the *semantic specification* of the dependency, whereas Question 4 asked for the *strength* of the dependency. Note that we interpret our survey results from a qualitative viewpoint, i.e., our results do not allow for precise quantifications of the investigated effects.

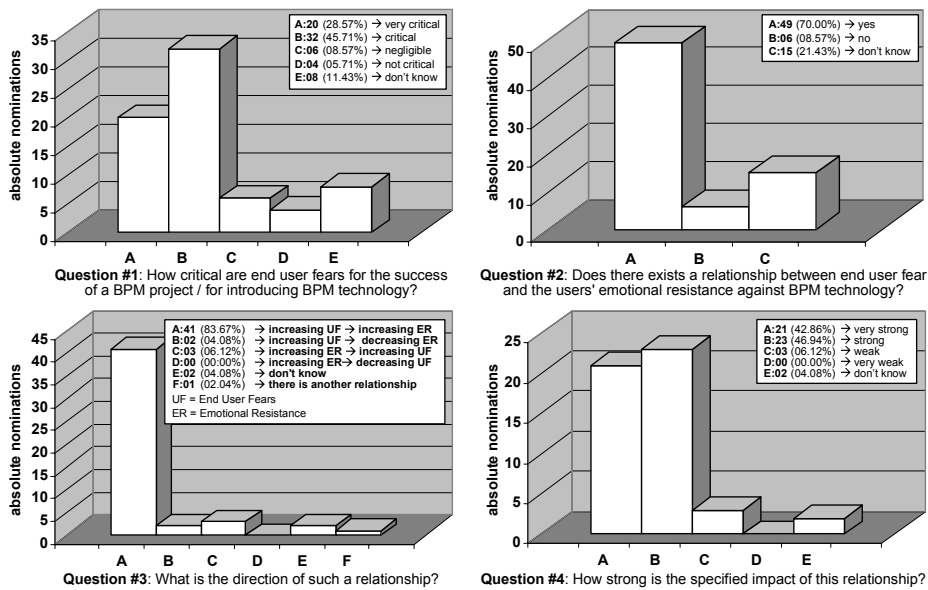


Fig. 4. Validating the Impact of End User Fears

Consider Fig. 4. The majority of 74.28% of the survey participants considers end user fears as "very critical" (28.57%) or "critical" (45.71%) for the overall success of

³ We have summarized the complete results of this survey in [13].

business process management (BPM) projects (cf. Question 1 in Fig. 4). More specifically, 70% of the survey respondents confirm that there is a relationship between end user fears and the emotional resistance of end users against BPM technology (cf. Question 2 in Fig. 4). This particularly confirms the positive link connecting these two variables in Fig. 3. Out of these respondents, 83.67% share the opinion that increasing end user fears result in increasing emotional resistance (cf. Question 3 in Fig. 4). Finally, 89.8% of the respondents state (cf. Question 4 in Fig. 4) that the impact of end user fears on emotional resistance either is "very strong" (42.86%) or "strong" (46.94%).

The evaluation model shown in Fig. 3 also considers the ImF "Communication". The majority of 92.86% of the survey participants consider communication between a BPM project's stakeholders as an "essential" (47.14%), "very important" (35.71%) or "important" (10%) factor. Furthermore, 78.57% of the respondents confirm that there is a relationship between communication and end user fears (cf. Fig. 3). Out of these, 74.55% share the opinion that an increasing communication results in decreasing end user fears. Finally, 85.45% of the respondents state that the impact of communication on end user fears either is "very strong" (29.09%) or "strong" (56.36%).

2.4 Investigating the Evolution of DCF and Dynamic ImF Through Simulation

The change of DCF and dynamic ImF is caused by the interplay between the different elements of an evaluation model, i.e., by the complex interdependencies between dynamic and static evaluation factors, flows, and links. In this context, *feedback loops* are of particular importance. A feedback loop is a *closed cycle* of causes and effects. Within this cycle, past events (like the change of a DCF or dynamic ImF) are utilized to control future actions (like another change of the same evaluation factor). In other words, if a *change* occurs in a model variable, which is part of a feedback loop, this change will be propagated around the loop [12]. As an example consider the feedback loop depicted in Fig. 3 (cf. Section 2.3).

We distinguish between two types of *loop polarities*. First, *positive* (or *self-reinforcing*) loops generate growth of DCF and dynamic ImF (cf. Fig. 5A). Second, *negative* (or *self-correcting*) loops counteract and oppose growth (cf. Fig. 5B). If evaluation models contain both positive and negative feedback loops, more complex effects may emerge (cf. Fig. 5C and Fig. 5D).

It is important to mention that the dynamic effects which are caused by feedback loops are typically not easy to understand [14]. For this reason, we investigate the effects of feedback loops by simulating [15] respective evaluation models.

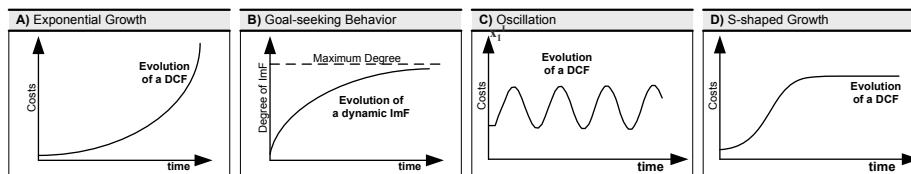


Fig. 5. Feedback in Evaluation Models: Overview of potential dynamic Effects

2.5 Specifying Simulation Models

In our EcoPOST framework, a *simulation model* consists of a number of *algebraic equations* — one for each model variable (i.e., dynamic and static evaluation factors as well as rate variables and auxiliary variables). The basic components of these algebraic equations are the model variables. In our approach, we use different types of algebraic equations for the different variables of an evaluation model (cf. Fig. 6A):

- **Static Evaluation Factors:** Static evaluation factors (i.e., SCF and static ImF) are specified using a numerical value in a *constant equation* (e.g., "Business Process Redesign Costs = 1000 \$/Week"). A specific variant of a constant equation is an *initially computed constant*. In fact, it will often become necessary to specify a constant in terms of another constant if the former depends on the latter and the former should change in any simulation run where the latter is given a new value [14]. As an example consider the following equation: Process Redesign Costs = 1000 \$/Week * Risk Factor. Note that initially computed constants need to be evaluated only once at the beginning of a simulation.
- **Dynamic Evaluation Factors:** Dynamic evaluation factors (i.e., DCF and dynamic ImF) are specified by *integral equations* in our approach [14]. Such equations specify the accumulation of a dynamic evaluation factor from a starting point t_0 to the present moment t (cf. Fig. 6B). More specifically, DCF and dynamic ImF integrate their *net flow*. The net flow during any interval $[t_1, t_2]$ is the area bounded by the graph of the *net rate* between the start and the end of the interval (cf. Fig. 6C). Thus, the value of a dynamic evaluation factor at t_2 can be calculated as the sum of its value at t_1 and the area under the net rate curve between t_1 and t_2 . In Fig. 6C, the value at t_1 is S_1 . Adding the area under the net rate curve between t_1 and t_2 increases the value to S_2 . The net flow is determined by one or several rate variables.

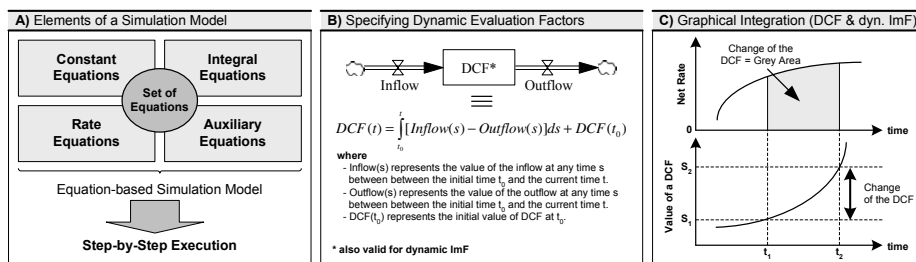


Fig. 6. Integration of Flows for Dynamic Evaluation Factors

- **Rate Variables:** Are expressed by *rate equations*. Rate equations specify the change of DCF or dynamic ImF between two computed conditions. Flows connected to DCF are specified by rate equations describing the amount of costs flowing to, from, or between DCF. Rate equations for flows connected to dynamic ImF specify the impact flowing to, from, or between dynamic ImF. In any case, a rate equation uses information (i.e., values) from other model variables (SCF, DCF, dynamic ImF,

and auxiliary variables) to calculate a specific change. In the context of a specific rate variable, the relevant information is represented by those model variables that are connected to the rate variable by links (cf. Section 2.2).

- **Auxiliary Variables:** Are specified by *auxiliary equations*. Constituting elements of these equations may be SCF, DCF, ImF, rate variables, and auxiliary variables. Note that auxiliary equations are evaluated after the integral equations on which they depend, and before the rate equations of which they are part.

Often, an ImF has a nonlinear impact on DCF. Such nonlinearities have to be represented in our simulation models as well. For this purpose, we use a specific kind of auxiliary equation (implying that nonlinearities require the introduction of additional auxiliary variables in our evaluation models). Specifically, we use *table functions* transferring an input value (e.g., a certain degree of process knowledge) into a corresponding output value (e.g., expressing a specific effect on a DCF) through a *lookup function* f [16]. Linear interpolation is used for values lying between the specified table values. Fig. 7, for example, shows typical table functions. Dependent on the degree of an ImF a specific *impact rating* is derived. An impact rating less than 1 results in decreasing costs (cf. Fig. 7A). A rating equal to 1 neither does increase nor decrease costs. A rating larger than 1 results in increasing costs (cf. Fig. 7B and Fig. 7C). Quantifications based on such impact ratings are also known from software cost models like COCOMO [17].

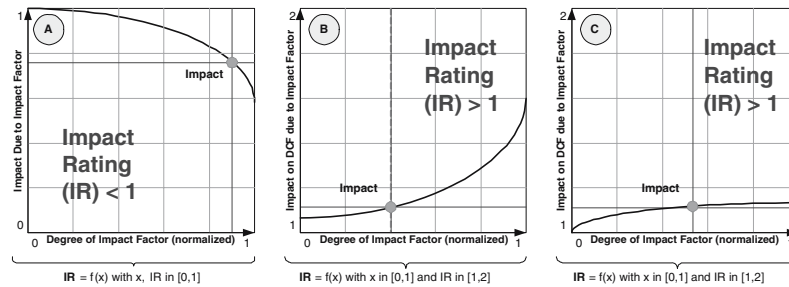


Fig. 7. Table Functions for quantifying Impact Factors

For the design of our evaluation models as well as their simulation existing System Dynamics modeling and simulation tools can be used (e.g., *Vensim* [18]). To support administrative tasks related to our framework, we have implemented the *EcoPOST Cost Benefit Analyzer*. Among other things, this tool comprises a knowledge base module for storing and managing VBEP as well as entire evaluation scenarios and a module for visualizing EcoPOST evaluations.

2.6 Using the Methodology in Practice - Lessons Learned

We have applied our methodology in an exploratory case study in the automotive domain. In this case study, we have analyzed cost overruns observed during the introduction of a large PDM system for the integrated support of engineering processes.

The initial business case for this project comprised seven major cost categories. In our case study, we analyzed three of them in more detail: *process management costs*, *IT system realization costs*, and *specification and test costs*. In particular, we analyzed whether the observed cost overruns in these cost categories could have been predicted using our cost analysis method. Based on real project data, interviews with project members, two user surveys, and practical experiences, we developed a set of evaluation models using our methodology and analyzed the effects described by these models using simulation. Taking our evaluation models, we were able to explain the observed cost overruns. Moreover, our models helped project members to better understand the complex cost structure of the analyzed project.

Our case study has also revealed several difficulties. In particular, it has turned out that the design of evaluation models constitutes a complex and time-consuming task. Evaluation models tend to become rather complex due to the large number of potential SCF, DCF, ImF and causal dependencies between them, and each evaluation model had to be designed from scratch. This resulted in the loss of valuable modeling experiences. In response to these issues the following section introduces the notion of *value-based evaluation patterns* (VBEP).

3 Value-Based Evaluation Patterns (VBEP)

The introduction of PAIS in production environments often exhibits similarities. Picking up these similarities by means of customizable generic evaluation patterns would be a useful step to simplify the use of our methodology and to increase model reuse. Therefore, we introduce a VBEP as a predefined, but customizable evaluation model.

Characterization. Basically, VBEP use the same elements as introduced in Section 2, i.e., they consist of an *evaluation model* and an associated *simulation model*. More precisely, each VBEP constitutes a template for specific sets of DCF and/or ImF we encounter when introducing PAIS.

Our approach distinguishes between primary and secondary VBEP. A *primary* VBEP describes a particular DCF, and a *secondary* VBEP describes an ImF. Characteristic VBEP are summarized in Table 1 (primary VBEP) and Table 2 (secondary VBEP).

Table 1. Primary VBEPs

Name	Description
<i>Process (Re)Design</i>	This VBEP deals with the costs of business process redesign activities prior to and during the development of a PAIS. Such a redesign may become necessary for several reasons, e.g., to increase the degree of automated process activities or to eliminate process performance flaws.
<i>Organizational Change</i>	This VBEP deals with the costs of changing an organization due to the introduction of a PAIS. As examples consider the adaptation of organizational structures like team structures and single jobs.
<i>Process Evolution</i>	This VBEP deals with the costs caused by the adaptation of business process changes. In fact, many business processes are continuously changing due to evolving business requirements. Any process change necessitates the adaptation of the supporting PAIS.
<i>Enterprise Architecture</i>	This VBEP deals with the costs caused by preparing an enterprise architecture for the introduction of a PAIS (e.g., costs for implementing interfaces to other legacy systems).
<i>Work Profile Change</i>	This VBEP deals with the costs related to changes in work profiles of end users of a PAIS. In particular, costs are caused by simultaneously holding up the new and the old work profile for some time.

Table 2. Secondary VBEPs

Name	Description
<i>Process Knowledge</i>	This VBEP deals with the effects of process knowledge, e.g., about data and control flows. Acquiring process knowledge causes efforts, e.g., for conducting interviews with process participants. However, process knowledge can also have a positive impact on other activities such as business process redesign.
<i>Domain Knowledge</i>	This VBEP deals with the impact of domain knowledge, e.g., of the experience of project members, on the costs of introducing a PAIS. Acquiring domain knowledge causes efforts, e.g., for the time needed to understand a complex domain. However, domain knowledge can also have a positive impact.
<i>Process Ownership</i>	This VBEP deals with the effects of a clear or unclear ownership of the business processes to be supported by a PAIS. The definition of explicit process ownerships typically implies efforts. However, clear process ownerships will have a positive impact if they are well-defined.
<i>Process Transparency</i>	This VBEP deals with the effects of process transparency during the introduction of a PAIS. A high process transparency has a positive impact on other activities such as the redesign of business processes.
<i>End User Fears</i>	This VBEP deals with the impact of end user fears on the ability to redesign business processes. We have already discussed this VBEP (cf. Fig. 3) in Section 2.3.

All these VBEP have been systematically derived based on the results of case studies we conducted in several information system engineering projects in the automotive domain (e.g., in projects dealing with the introduction of PDM and ERP systems). Furthermore, we rely on results of several online surveys. However, we do not claim for completeness here, and we are continuously working on the extension of our pattern collection.

Generally, VBEP enable the reuse of historical evaluation data. This reduces the need for designing evaluation models from scratch. Moreover, VBEP are useful as a means to increase the awareness for cost effects in PAIS engineering projects.

Customization. Customization becomes necessary as VBEP are applied in different evaluation context. Thereby, we distinguish between the customization of the *evaluation model* (Step I in Fig. 8) and the *simulation model* (Step II in Fig. 8) of a VBEP. The former always requires the subsequent adaptation of the underlying simulation model, while the latter is also possible without adapting the assigned evaluation model.

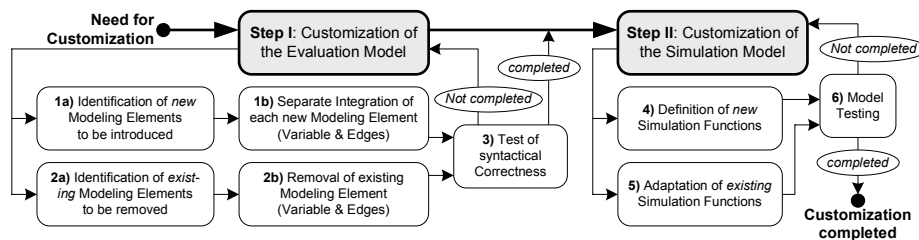


Fig. 8. Step-by-Step Customization of VBEP

Adapting an evaluation model can be achieved by adding or removing model variables, flows, or links (Step 1a/b and Step 2a/b in Fig. 8). The VBEP "End User Fears" (cf. Fig. 3), for example, could be customized by introducing an ImF "Management Commitment" to take into account the impact of this factor on end user fears. Therefore, the new ImF "Management Commitment" is connected to the ImF "End User Fears".

In our example this can be achieved with a negative link to denote that increasing management commitment results in decreased user fears. The correctness of a customized VBEP is ensured through the design rules discussed in Section 2.2.

Customizing a simulation model, by contrast, requires adaptations of the equations of the simulation model (Step 4 and Step 5 in Fig. 8). As examples of potential customizations consider changes of SCF values or adaptations of rate functions.

Merging VBEP. Customization becomes also necessary when VBEP are merged. Assume, for example, that an ImF "End User Fears" (cf. Fig. 9B) has to be considered in the context of a DCF "Costs for Business Process Redesign" (cf. Fig. 1C). This can be realized by merging a secondary VBEP (specifying the additional ImF) with a primary VBEP (specifying the DCF). Regarding our evaluation models, this merge can be (partially) automated. As input, a respective algorithm needs two evaluation models EM1 and EM2. The merge of EM1 and EM2 is then accomplished through a systematic comparison of all model variables from EM1 with all model variables from EM2. If a model variable from EM1 (e.g., a DCF) has the same name and type as a model variable from EM2, both variables (and their links) will be merged.

Applying this algorithm requires that the evaluation models to be merged exhibit some overlap, i.e., both models have to contain at least one identical model variable. In our example, the ImF "Ability to Redesign Business Processes" has been the mixing

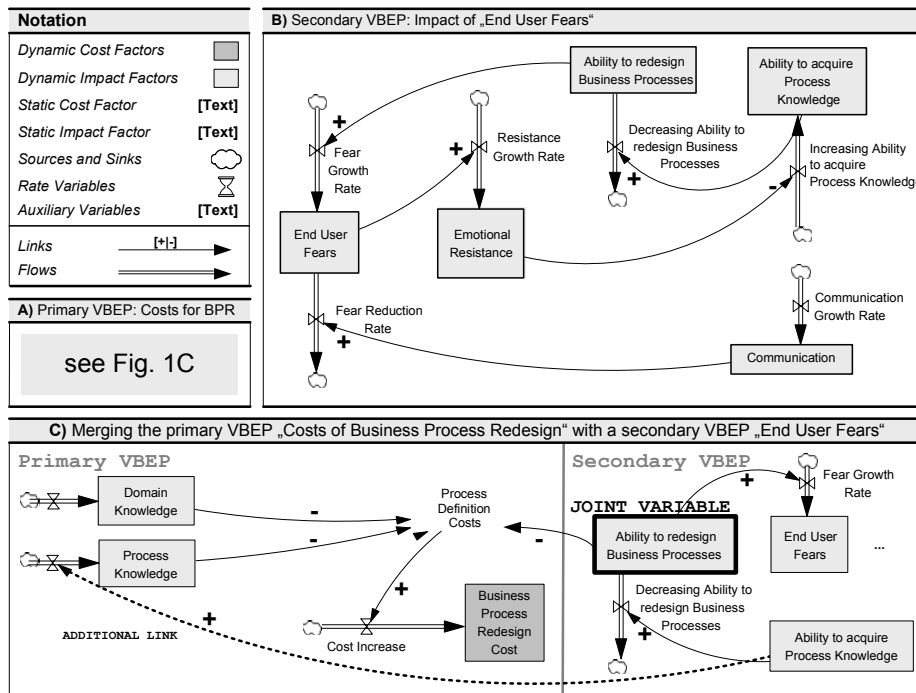


Fig. 9. Combining primary and secondary VBEP

point (cf. Fig. 9C). If there exist no identical variables, evaluation models can be merged manually. Besides, any merge typically requires an additional editing of the newly derived model (regardless whether the merge has been automatically conducted or not). In Fig. 9), for example, we introduce an additional link between the ImF "Ability to acquire Process Knowledge" and the ImF "Process Knowledge".

4 Related Work

Boehm et. al [19] propose a classification of cost estimation techniques into six major categories. In particular, they distinguish between *model-based approaches* (e.g., COCOMO, SLIM), *expertise-based approaches* (e.g., the Delphi method), *learning-oriented approaches* (using neural networks or case based reasoning), *regression-based approaches* (e.g., the ordinary least squares method), *composite approaches* (e.g., the Bayesian approach), and *dynamic-based approaches* (which explicitly acknowledge that cost factors change over the duration of the system development). Picking up this classification, our methodology can be considered as an example of a dynamic-based approach (the other five categories rely on static analysis models).

The use of patterns has been widely discussed since the advent of computer science research. At present, the software community is using numerous variations of patterns largely for software architecture (*conceptual patterns*), design (*design patterns*), programming (*XML schema patterns*, *J2EE patterns*, etc.), as well as for software development processes. Recently, the idea of using patterns has been also applied to more specific domains like workflow management [20] or inter-organizational control [21].

Regarding the reuse of (System Dynamics) models, one has to distinguish between two basic directions. On the one hand, authors like Senge [22], Eberlein and Hines [23], Liehr [24], and Myrtveit [25] introduce predefined generic structures (with slightly different semantics). All these approaches satisfy the capability of defining "components". On the other hand, Winch [26] proposes a more restrictive approach which is only based on the parameterization of generic structures (without providing standardized modeling components). Our approach picks up ideas from both directions, i.e. we address both the definition of generic components and customization.

5 Summary

This paper has presented a qualitative cost analysis methodology to investigate the complex dependencies and interactions of those factors that determine the costs of PAIS engineering projects. We have presented a formalism to design evaluation models and exemplarily discussed one evaluation model and its derivation based on the results of an empirical study. Finally, we have introduced the notion of *value-based evaluation patterns* (VBEP) as a means to enable the reuse of evaluation data in different context.

Note that the expressiveness of simulation always depends on the plausibility and resilience of the underlying simulation models. Therefore, we have additionally accomplished various empirical and experimental research activities (e.g., software experiments, online surveys, case studies) in order to put the quantifications gained from our simulation models on a more reliable basis (cf. [27] for examples).

References

1. Dehnert, J., van der Aalst, W.: Bridging the Gap between Business Models and Workflow Specification. *Int'l. Journal of Cooperative Information Systems* (2004)
2. Reichert, M., Rinderle, S., Kreher, U., Dadam, P.: Adaptive Process Management with ADEPT2. *Proc. 21th ICDE*, pp.1113-1114 (2005)
3. Dumas, M., van der Aalst, W.M.P., ter Hofstede, A.H. (eds.): *Process-aware Information Systems: Bridging People and Software through Process Technology*. Wiley, Chichester (2005)
4. Reijers, H.A., van der Aalst, W.M.P.: The Effectiveness of Workflow Management Systems - Predictions and Lessons Learned. *Int'l. J. of Inf. Manag.* 25(5), 457–471 (2005)
5. Choenni, S., Bakkera, R., Baetsa, W.: On the Evaluation of Workflow Systems in Business Processes. *Electronic Journal of IS Evaluation (EJISE)* vol.6(2) (2003)
6. Kleiner, N.: Can Business Process Changes Be Cheaper Implemented with Workflow-Management-Systems?. In: *Proc. IRMA 2004*, pp. 529–532 (2004)
7. Mutschler, B., Reichert, M., Bumiller, J.: Designing an Economic-driven Evaluation Framework for Process-oriented Software Technologies. In: *Proc. 28th ICSE*, pp. 885–888 (2006)
8. Yu, E.: *Modelling Strategic Relationships for Process Reengineering*. PhD Thesis, University of Toronto (1995)
9. Mutschler, B., Zarvic, N., Reichert, M.: A Survey on Economic-driven Evaluations of Information Technology. Technical Report, TR-CTIT-07, University of Twente (2007)
10. Mutschler, B., Reichert, M., Bumiller, J.: An Approach for Evaluating Workflow Management Systems from a Value-Based Perspective. In: *Proc. 10th IEEE EDOC*, pp. 477–482 (2006)
11. Richardson, G.P., Pugh, A.L.: *System Dynamics - Modeling with DYNAMO* (1981)
12. Ogata, K.: *SD*. Prentice-Hall, Englewood Cliffs (2003)
13. Mutschler, B., Reichert, M.: A Survey on Evaluation Factors for Business Process Management Technology. Technical Report, TR-CTIT-06-63, University of Twente (2006)
14. Forrester, J.W.: *Industrial Dynamics*. Productivity Press, Cambridge, London (1961)
15. Vangheluwe, H., de Lara, J., Mosterman, P.J.: An Introduction to Multi-Paradigm and Simulation. In: *Proc. AIS 2002*, pp. 9–20 (2002)
16. Mutschler, B., Reichert, M.: Simulation Models for Analyzing the Dynamic Costs of Process-aware IS. Technical Report, TR-CTIT-07-14, University of Twente (2007)
17. Boehm, B., Abts, C., Brown, A.W., Chulani, S., Clark, B.K., Horowitz, E., Madachy, R., Reifer, D., Steece, B.: *Software Cost Estimation with Cocomo 2*. Prentice-Hall, Englewood Cliffs (2000)
18. Vensim: Ventana Systems (2006) <http://www.vensim.com/>
19. Boehm, B., Abts, C., Chulani, S.: Software Development Cost Estimation Approaches - A Survey. Technical Report, USC-CSE-2000-505 (2000)
20. van der Aalst, W.M.P., ter Hofstede, A.H.M., Kiepuszewski, B., Barros, A.P.: Advanced Workflow Patterns. In: *Proc. 7th CoopIS, LNCS 1901*, pp. 18–29 (2000)
21. Kartseva, V., Hulstijn, J., Tan, Y.H., Gordijn, J.: Towards Value-based Design Patterns for Inter-Organizational Control. In: *Proc. 19th Bled Electronic Commerce Conference* (2006)
22. Senge, P.M.: *The 5th Discipline: The Art and Practice of the Learning Organization* (1990)
23. Eberlein, R.J., Hines, J.H.: Molecules for Modelers. In: *Proc. 14th System Dyn. Conf.* (1996)
24. Liehr, M.: A Platform for SD Modeling – Methodologies for the Use of predefined Model Components. In: *Proc. 20th System Dynamics Conf.* (2002)
25. Myrtveit, M.: Object-oriented Extensions to SD. In: *Proc. 18th System Dynamics Conf.* (2000)
26. Winch, G.: User-parameterized generic Models: A Solution to the Conundrum of Modelling Access for SMEs? *SD Review* 18(3), 339–357 (2003)
27. Mutschler, B., Reichert, M., Bumiller, J.: Why Process-Oriented is Scarce: An Emp. St. of Process-oriented IS in the Automotive Industry. In: *Proc. 10th IEEE EDOC*, pp. 433–438 (2006)