On Modeling and Analyzing Cost Factors in Information Systems Engineering

Bela Mutschler[1,2] and Manfred Reichert[2,3]

[1] Daimler AG, Group Research, P.O. Box 2360, 89013 Ulm, Germany
bela.mutschler@daimler.com
[2] Information Systems Group, University of Twente, The Netherlands
m.u.reichert@utwente.nl
[3] Institute of Databases and Information Systems, University of Ulm, Germany
manfred.reichert@uni-ulm.de

Abstract. Introducing *enterprise information systems* (EIS) is usually associated with high costs. It is therefore crucial to understand those factors that determine or influence these costs. Though software cost estimation has received considerable attention during the last decades, it is difficult to apply existing approaches to EIS. This difficulty particularly stems from the inability of these methods to deal with the dynamic interactions of the many technological, organizational and project-driven cost factors which specifically arise in the context of EIS. Picking up this problem, we introduce the EcoPOST framework to investigate the complex cost structures of EIS engineering projects through qualitative cost evaluation models. This paper extends previously described concepts and introduces design rules and guidelines for cost evaluation models in order to enhance the development of meaningful and useful EcoPOST cost evaluation models. A case study illustrates the benefits of our approach. Most important, our EcoPOST framework is an important tool supporting EIS engineers in gaining a better understanding of the critical factors determining the costs of EIS engineering projects.

Keywords: Information Systems Engineering, Cost Analysis, Evaluation Models, Simulation.

1 Introduction

While the benefits of *enterprise information systems* (EIS) are usually justified by improved process performance [1], there exist no approaches for systematically analyzing related cost factors and their dependencies. Though software cost estimation has received considerable attention during the last decades [2] and has become an essential task in information systems engineering, it is difficult to apply existing approaches to EIS, particularly if the considered EIS shall provide active business process support. 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 process-aware EIS (and which do only partly exist in data- or function-centered information systems) [3]. As an example consider the significant costs for redesigning business processes. Another challenge deals with the many dependencies existing between different cost factors. Activities for *business process*

redesign, for example, can be influenced by intangible impact factors like available *process knowledge* or *end user fears*. These dependencies, in turn, result in dynamic effects which influence the overall costs of EIS engineering projects. Existing evaluation techniques [4] are typically unable to deal with such dynamic effects as they rely on too static models based upon snapshots of the considered software system.

What is needed is an approach that enables project managers and EIS engineers to model and investigate the complex interplay between the many cost and impact factors that arise in the context of EIS. This paper presents the EcoPOST methodology, a sophisticated and practically validated, model-based methodology to better understand and systematically investigate the complex cost structures of EIS engineering projects. Specifically, this paper extends our previous work on EcoPOST [5] by introducing model design rules and modeling guidelines, which enhance the development of meaningful and useful evaluation models.

Section 2 summarizes the EcoPOST methodology. Section 3 introduces rules for designing evaluation models and Section 4 describes modeling guidelines. Section 5 summarizes results from one of our case studies in order to illustrate the benefits of the EcoPOST approach. It further discusses issues related to validation research from a more general perspective. Section 6 concludes with a summary.

2   The EcoPOST Cost Analysis Methodology

Our EcoPOST methodology was designed to ease the realization of process-aware EIS in the automotive industry (and was, consequently, also validated and piloted in several EIS engineering projects in this domain). The EcoPOST methodology comprises seven steps (cf. Fig. 1). *Step 1* concerns the comprehension of an evaluation scenario. This is crucial for developing problem-specific evaluation models. The following two steps (Steps 2 and 3) deal with the identification of two different kinds of *Cost Factors* representing costs that can be quantified in terms of money (cf. Table 1): *Static Cost Factors* (SCFs) and *Dynamic Cost Factors* (DCFs).

Step 4 deals with the identification of *Impact Factors* (ImFs), i.e., intangible factors that influence DCFs and other ImFs. We distinguish between organizational, project-specific, and technological ImFs. ImFs cause the value of DCFs (and other ImFs) to

Table 1. Cost Factors

	Description
SCF	*Static Cost Factors* (SCFs) represent costs whose values do not change during an EIS engineering project (except for their time value, which is not further considered in the following). Typical examples: software license costs, hardware costs and costs for external consultants.
DCF	*Dynamic Cost Factors* (DCFs), in turn, represent costs that are determined by activities related to an EIS engineering project. The (re)design of business processes prior to the introduction of EIS, for example, constitutes such an activity. As another example consider the performance of interview-based process analysis. These activities cause measurable efforts which, in turn, vary due to the influence of intangible *impact factors*. The DCF "Costs for Business Process Redesign" may be influenced, for instance, by an intangible factor "Willingness of Staff Members to Support Process (Re)Design Activities". Obviously, if staff members do not contribute to a (re)design project by providing needed information (e.g., about process details), any redesign effort will be ineffective and result in increasing (re)design costs. If staff willingness is additionally varying during the (re)design activity (e.g., due to a changing communication policy), the DCF will be subject to even more complex effects. In the EcoPOST framework, intangible factors like the one described are represented by *impact factors*.

change, making their evaluation a difficult task to accomplish. As examples consider factors such as "End User Fears", "Availability of Process Knowledge", or "Ability to (re)design Business Processes". Also, ImFs can be static or dynamic (cf. Table 2).

Table 2. Impact Factors

	Description
Static ImF	Static ImFs do not change, i.e., they are assumed to be constant during an EIS engineering project; e.g., when there is a fixed degree of user fears, process complexity, or work profile change.
Dynamic ImF	Dynamic ImFs may change during an EIS engineering project, e.g., due to interference with other ImFs. As examples consider process and domain knowledge which is typically varying during an EIS engineering project (or a subsidiary activity).

It is important to mention that – unlike SCFs and DCFs – the values of ImFs are not quantified in monetary terms. Instead, they are "quantified" by experts[1] using qualitative scales describing the degree of an ImF. As known from software cost estimation models, such as Boehm's COCOMO [2], the qualitative scales we use comprise different "values" (typically ranging from "very low" to "very high"). These values are used to express the strength of an ImF on a given cost factor (just like in COCOMO).

Generally, dynamic evaluation factors (i.e., DCFs and dynamic ImFs) are difficult to comprehend. In particular, intangible ImFs (i.e., their appearance and impact in EIS engineering projects) are not easy to follow. When evaluating the costs of EIS engineering projects, therefore, DCFs and dynamic ImFs constitute a major source of misinterpretation and ambiguity. To better understand and to investigate the dynamic behavior of DCFs and dynamic ImFs, we introduce the notion of *evaluation models* as basic pillar of the EcoPOST methodology (*Step 5*; cf. Section 2.2). These evaluation models can be simulated (*Step 6*) to gain insights into the dynamic behavior (i.e., evolution) of DCFs and dynamic ImFs (*Step 7*). This is important to effectively control the design and implementation of EIS as well as the costs of respective projects.
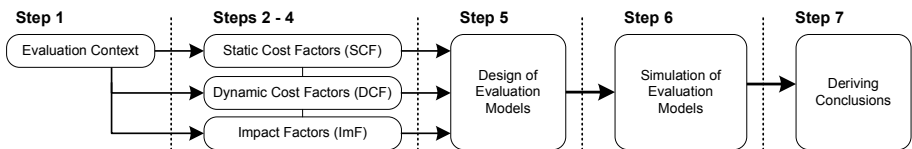

Fig. 1. Main Steps of the EcoPOST Methodology

2.1   Evaluation Models

In EcoPOST, dynamic cost/impact factors are captured and analyzed by evaluation models which are specified using the System Dynamics [6] notation (cf. Fig. 2). An evaluation model comprises SCFs, DCFs, and ImFs corresponding to model variables. Different types of variables exist. *State variables* can be used to represent dynamic factors, i.e., to capture changing values of DCFs (e.g., the "Business Process Redesign

---

[1] The efforts of these experts for making that quantification is not explicitly taken into account in EcoPOST, though this effort also increases information system development costs.

Costs"; cf. Fig. 2A) and dynamic ImFs (e.g., "Process Knowledge"). A state variable is graphically denoted as rectangle (cf. Fig. 2A), 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 poured into it in the past. Typically, state variables are connected to at least one *source* or *sink* which are graphically denoted as cloud-like symbols (except for state variables connected to other ones) (cf. Fig. 2A). 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. 2A). Picking up again the bathtub image, an *inflow* is a pipe that adds water to the bathtub, i.e., inflows increase the value of state variables. An *outflow*, by contrast, is a pipe that purges water from the bathtub, i.e., outflows decrease the value of state variables. The DCF "Business Process Redesign Costs" shown in Fig. 2A, 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. 2A). In particular, a rate variable controls the inflow/outflow it is assigned to based on those SCFs, DCFs, and ImFs which influence it. It can be considered as an interface which is able to merge SCFs, DCFs, and ImFs.
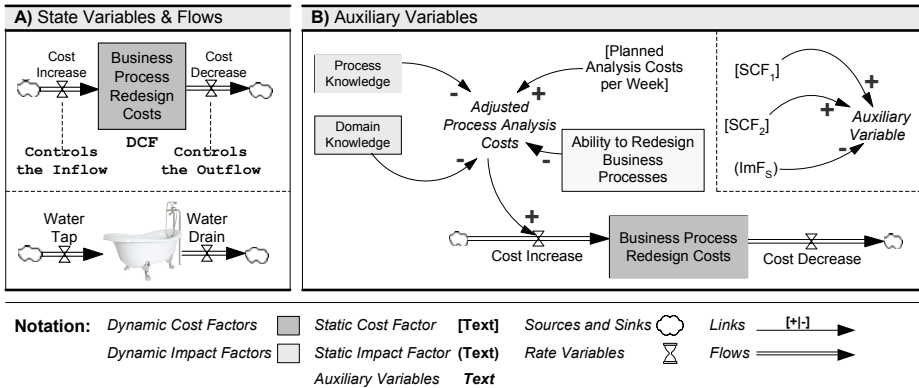

Fig. 2. Evaluation Model Notation and initial Examples

Besides state variables, evaluation models may comprise *constants* and *auxiliary variables*. Constants are used to represent static evaluation factors, i.e., SCFs and static ImFs. Auxiliary variables, in turn, represent intermediate variables and typically bring together – like rate variables – cost and impact factors, i.e., they merge SCFs, DCFs, and ImFs. As an example consider the auxiliary variable "Adjusted Process Analysis Costs" in Fig. 2B, which merges the three dynamic ImFs "Process Knowledge", "Domain Knowledge", and "Ability to Redesign Business Processes" and the SCF "Planned Analysis Costs per Week". 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. Altogether, we define:

Definition 2.1 (Evaluation Model). *A graph EM = (V, F, L) is denotes as evaluation model, if the following holds:*

- $V := S \mathbin{\dot\cup} X \mathbin{\dot\cup} R \mathbin{\dot\cup} C \mathbin{\dot\cup} A$ *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 representing 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 representing links with Lab := {+, −} being the set of link labels:*
  - $(q_i, q_j, +) \in L$ *with* $q_i \in (S \mathbin{\dot\cup} A \mathbin{\dot\cup} C)$ *and* $q_j \in (A \mathbin{\dot\cup} R)$ *denotes a positive link,*
  - $(q_i, q_j, -) \in L$ *with* $q_i \in (S \mathbin{\dot\cup} A \mathbin{\dot\cup} C)$ *and* $q_j \in (A \mathbin{\dot\cup} R)$ *denotes a negative link.*

The EcoPOST evaluation models presented so far are already useful for EIS engineers and project managers. However, the evolution of DCFs and dynamic ImFs is still difficult to comprehend. Thus, we have added a simulation component to our evaluation framework for analyzing this evolution (cf. Step 6 in Fig. 1).

2.2 Understanding Model Dynamics through Simulation

To enable simulation of an evaluation model we need to formally specify its behavior by means of a *simulation model*. We use *mathematical equations* for this purpose. Thereby, the behavior of each model variable is specified by one equation (cf. Fig. 3), which describes how a variable is changing over time from simulation start.

Fig. 4A shows a simple evaluation model.[2] Assume that the evolution of the DCF "Business Process Redesign Costs" (triggered by dynamic ImF "End User Fears") shall be analyzed. End user fears can lead to emotional resistance of users and, in turn, to a lack of user support when redesigning business processes (e.g., during an interview-based process analysis). For model variables, which represent an SCF or static ImF, the equation specifies a constant value for the model variable; i.e., SCFs and static ImFs are specified by single numerical values in *constant equations*. As example consider EQUATION A in Fig. 4B. For model variables representing DCFs, dynamic ImFs, or rate/auxiliary variables, the corresponding equation describes how the value of the model variable evolves over time (i.e., during simulation). Thereby, the evolution of DCFs and dynamic ImFs is characterized by *integral equations* [7]. This allows us to

---

[2] It is the basic goal of this toy example to illustrate simulation of evaluation models. Generally, evaluation models are much more complex. Due to lack of space we do not provide a more extensive example.
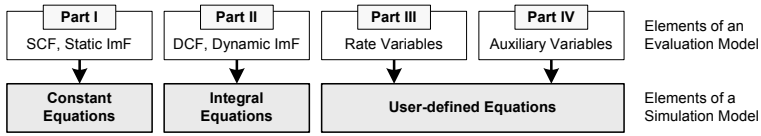
Fig. 3. Elements of a Simulation Model

capture the accumulation of DCFs and dynamic ImFs from the start of a simulation run (t_0) to its end (t):

Definition 2.2 (Integral Equation). *Let EM be an evaluation model (cf. Definition 2.1) and S be the set of all DCFs and dynamic ImFs defined by EM. An integral equation for a dynamic factor $v \in S$ is defined as follows:*

$$v(t) = \int_{t_0}^{t} [inflow(s) - outflow(s)]ds + v(t_0) \text{ where}$$

- *t_0 denotes the starting time of the simulation run,*
- *t represents the end time of the simulation run,*
- *$v(t_0)$ represents the value of v at t_0,*
- *$inflow(s)$ represents the value of the inflow at any time s between t_0 and t,*
- *$outflow(s)$ represents the value of the outflow at any time s between t_0 and t.*
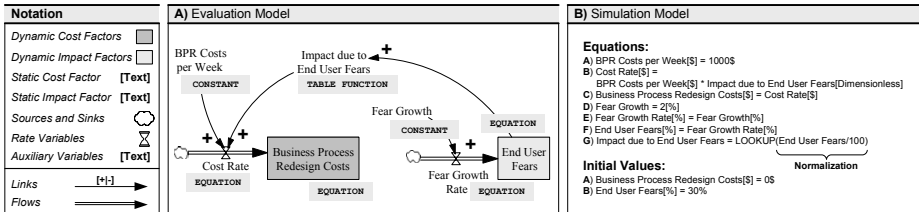

Fig. 4. Dealing with the Impact of End User Fears

As example consider EQUATION C in Fig. 4B which specifies the increase of the DCF "Business Process Redesign Costs" (based on only one inflow). Note that in Fig. 4B the equations for the DCF "Business Process Redesign Costs" and the dynamic ImF "End User Fears" are presented in the way they are specified in Vensim [8], the tool we use in EcoPOST, and not as real integral equations.

Rate and auxiliary variables are both specified in the same way, i.e., as user-defined functions defined over the variables preceding them in the evaluation model. In other words, rate as well as auxiliary variables are used to merge static and dynamic cost/impact factors. During simulation, values of rate and auxiliary variables are dynamic, i.e., they change along the course of time. Reason is that they are not only influenced by SCFs and static ImFs, but also by evolving DCFs and dynamic ImFs. The behavior of rate and auxiliary variables is specified in the same way:

Definition 2.3 (User-defined Equation). *Let EM be an evaluation model (cf. Def. 2.1) and X be the set of rate/auxiliary variables defined by EM. An equation for $v \in X$ is a user-defined function $f(v_1, ..., v_n)$ with $v_1, ..., v_n$ being the predecessors of v in EM.*

As example consider EQUATION B in Fig. 4B. The equation for rate variable "Cost Rate" merges the SCF "BPR Costs per Week" with the auxiliary variable "Impact due to End User Fears". Assuming that activities for business process redesign are scheduled for 32 weeks, Fig. 5A shows the values of all dynamic evaluation factors of the evaluation model over time when performing simulation. Fig. 5B shows the outcome of the simulation. As can be seen there is a significant negative impact of end user fears on the costs of business process redesign.
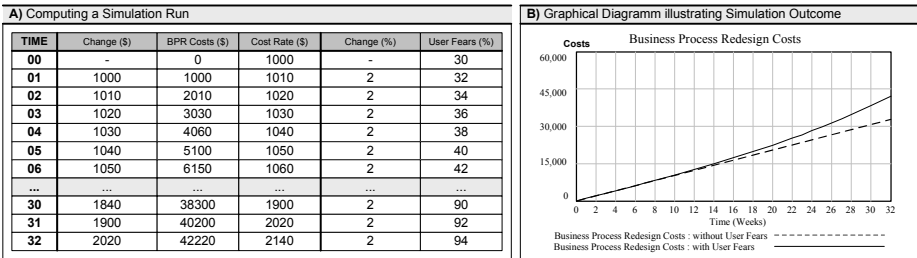
A) Computing a Simulation Run					
TIME	Change ($)	BPR Costs ($)	Cost Rate ($)	Change (%)	User Fears (%)
00	-	0	1000	-	30
01	1000	1000	1010	2	32
02	1010	2010	1020	2	34
03	1020	3030	1030	2	36
04	1030	4060	1040	2	38
05	1040	5100	1050	2	40
06	1050	6150	1060	2	42
...	...	...	...	...	...
30	1840	38300	1900	2	90
31	1900	40200	2020	2	92
32	2020	42220	2140	2	94

Fig. 5. Dealing with the Impact of End User Fears

2.3   Sensitivity Analysis and Reuse of Evaluation Information

Generally, results of a simulation enable EIS engineers to gain insights into causal dependencies between organizational, technological, and project-specific factors. This helps them to better understand resulting effects and to develop a concrete "feeling" for the dynamic implications of EcoPOST evaluation models. To investigate how a given evaluation model "works" and what might change its behavior, we simulate the dynamic implications described by it – a task which is typically too complex for the human mind. In particular, we conduct "behavioral experiments" based on series of simulation runs. During these simulation runs selected simulation parameters are manipulated in a controlled manner to systematically investigate the effects of these manipulations, i.e., to investigate how the output of a simulation will vary if its initial condition is changed. This procedure is also known as *sensitivity analysis*. Simulation outcomes can be further analyzed using graphical charts.

Designing evaluation models can be a complicated and time-consuming task. Evaluation models can become complex due to the high number of potential cost and impact factors as well as the many causal dependencies that exist between them. Taking the approach described so far (cf. Section 2), each evaluation and simulation model has to be designed from scratch. Besides the additional efforts, this results in an exlusion of existing modeling experience, and prevents the reuse of both evaluation and simulation models. In response to this problem, in [5,9] we have introduced a set of reusable *evaluation patterns* (EP). EPs do not only ease the design and simulation of evaluation

models, but also enable the reuse of evaluation information. This is crucial to foster the practical applicability of the EcoPOST framework.

3   Model Design Rules

Overall benefit of EcoPOST evaluation models depends on their quality. The latter, in turn, is determined by the syntactical as well as the semantical correctness of the evaluation model. Maintaining correctness of an evaluation model, however, can be a difficult task to accomplish. This section picks up this problem.

3.1   Modeling Constraints for Evaluation Models

Rules for the correct use of flows and links are shown in Fig. 6A and Fig. 6B. By contrast, Fig. 7A – Fig. 7F show examples of *incorrect* models.
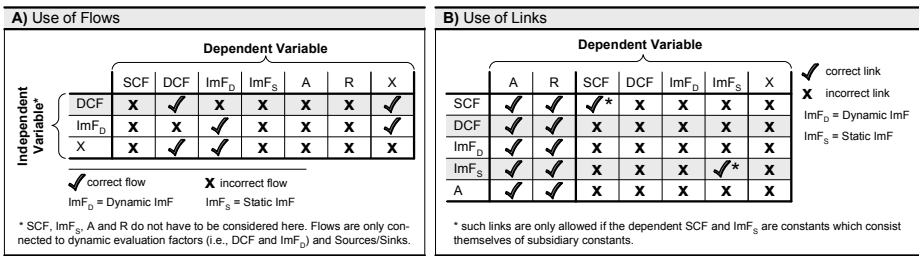

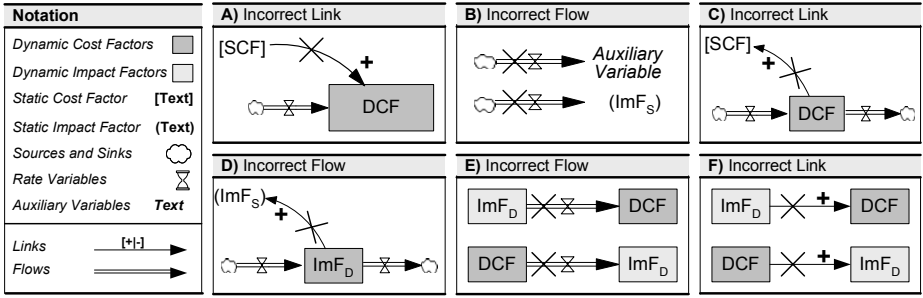**Fig. 6.** Using Flows and Links in our Evaluation Models

Dynamic evaluation factors, for example, may be only influenced by flows and not by links as shown in Fig. 7A. Likewise, flows must be not connected to auxiliary variables or constants (cf. Fig. 7B). Links pointing from DCFs (or auxiliary variables) to SCFs or static ImFs (cf. Fig. 7C and Fig. 7D) are also not valid as SCFs as well as static ImFs have constant values which cannot be influenced. Finally, flows and links connecting DCFs with dynamic ImFs (and vice versa) are also not considered as correct (cf. Fig. 7E and Fig. 7F).

Several other constraints have to be taken into account as well when designing evaluation models. In the following let $EM = (V, F, L)$ be an evaluation model (cf. Definition 2.1). Then:

***Design Rule 1* (Binary Relations).** Every model variable must be used in at least one binary relation. Otherwise, it is not part of the analyzed evaluation context and can be omitted:

$$\forall v \in (S \,\dot{\cup}\, X) : \exists q \in (S \,\dot{\cup}\, X) \wedge ((v,q) \in F \vee (v,q) \in F) \tag{1}$$

$$\forall v \in (A \,\dot{\cup}\, C) : \exists q \in (A \,\dot{\cup}\, R) \wedge \exists\, (q,v,[+|-]) \in L \tag{2}$$

Fig. 7. Examples of Incorrect Modeling

***Design Rule 2* (Sources and Sinks).** Every state variable must be connected to at least one source, sink or other state variable. Otherwise it cannot change its value and therefore would be useless:

$$\forall v \in S : \exists q \in (S \cup X) \wedge ((v,q) \in F \vee (q,v) \in F) \tag{3}$$

***Design Rule 3* (Rate Variables).** Every rate variable is influenced by at least one link; otherwise the variable cannot change and therefore is useless:

$$\forall v \in R : \exists q \in (S \,\dot\cup\, A \,\dot\cup\, C) \wedge \exists \, (q,v,[+|-]) \in L \tag{4}$$

***Design Rule 4* (Feedback Loops).** There are no cycles consisting only of auxiliary variables, i.e., cyclic feedback loops must at least contain one state variable (cycles of auxiliary variables cannot be evaluated if an evaluation model is simulated):

$$\neg \exists < q_0, q_1, ..., q_r > \in A^{r+1} \; with \; (q_i, q_{i+1}, [+|-]) \in L \; for$$
$$i = 0, ..., r-1 \wedge q_0 = q_r \wedge \; q_k \neq q_l \; for \; k,l = 1, ..., r; k \neq l \tag{5}$$

***Design Rule 5* (Auxiliary Variables).** An auxiliary variable has to be influenced by at least two other static or dynamic evaluation factors or auxiliary variables (except for auxiliary variables used to represent table functions [9]):

$$\forall v \in A : \exists p,q \in (A \,\dot\cup\, S \cup C) \wedge ((q,v,[+|-]) \in L \wedge (p,v,[+|-]) \in L) \tag{6}$$

These modeling constraints provide basic rules for EcoPOST users to construct syntactically correct evaluation models.

3.2   Semantic Correctness of Evaluation Models

While syntactical model correctness can be ensured, this is not always possible for the semantical correctness of evaluation models. Yet, we can provide additional model design rules increasing the meaningfulness of our evaluation models.

Design Rule 6 (**Transitive Dependencies**). Transitive link dependencies (i.e., indirect effects described by chains of links) are restricted. As example consider Fig. 8. Fig. 8A reflects the assumption that increasing end user fears result in increasing emotional resistance. This, in turn, leads to increasing business process costs. Consequently, the modeled transitive dependency between "End User Fears" and "Business Process Redesign Costs" is not correct, as increasing end user fears do not result in decreasing business process (re)design costs. The correct transitive dependency is shown in Fig. 8B. Fig. 8C illustrates the assumption that increasing process knowledge results in an increasing ability to (re)design business processes. An increasing ability to (re)design business processes, in turn, leads to decreasing process definition costs. The modeled transitive dependency between "Process Knowledge" and "Process Definition Costs", however, is not correct, as increasing process knowledge does not result in increasing process definition costs (assuming that the first 2 links are correct). The correct transitive dependency is shown in Fig. 8D.
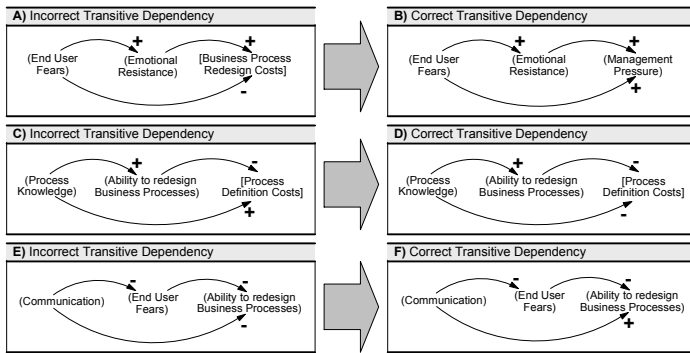

Fig. 8. Transitive Dependencies (Simplified Evaluation Models)

Finally, Fig. 8E deals with the impact of communication (e.g., the goals of an EIS project) on the ability to redesign business processes. Yet, the transitive dependency shown in Fig. 8E is not correct. The correct one is shown in Fig. 8F.

Altogether, two causal relations ("+" and "-") are used in the context of our evaluation models. Correct transitive dependencies can be described based on a *multiplication operator*. More precisely, transitive dependencies have to comply with the following three *multiplication laws* for transitive dependencies (for any $x, y \in \{+, -\}$):

$$+ * y = y \tag{7}$$
$$- * - = + \tag{8}$$
$$x * y = y * x \tag{9}$$

The evaluation models shown in Fig. 8A and Fig. 8C violate Law 1, whereas the model shown in Fig. 8E violates the second one. Law 3 states that the "*" is commutative.

Design Rule 7 (**Dual Links I**). A constant cannot be connected to the same auxiliary variable with both a positive and negative link:

$$\forall v \in C, \forall q \in A : \neg \exists\, l_1, l_2 \in L \; with\, l_1 = (v, q, -) \wedge l_2 = (v, q, +) \tag{10}$$

***Design Rule 8* (Dual Links II).** A state variable cannot be connected to the same auxiliary variable with both a positive and negative link:

$$\forall v \in S, \forall q \in A : \neg \exists\, l_1, l_2 \in L \; with\, l_1 = (v, q, -) \wedge l_2 = (v, q, +) \tag{11}$$

***Design Rule 9* (Dual Links III).** An auxiliary variable cannot be connected to another auxiliary variable with both a positive and negative link:

$$\forall v \in A, \forall q \in A : \neg \exists\, l_1, l_2 \in L \; with\, l_1 = (v, q, -) \wedge l_2 = (v, q, +) \tag{12}$$

Finally, there exist two additional simple constraints:

***Design Rule 10* (Representing Cost Factors).** A cost factor cannot be represented both as SCF and DCF in one evaluation model.

***Design Rule 11* (Representing Impact Factors).** An impact factor cannot be represented both as static and dynamic ImF in one evaluation model.

Without providing model design rules, incorrect evaluation models can be quickly modeled. This, in turn, does not only aggravate the derivation of plausible evaluations, but also hampers the use of the modeling and simulation tools [5] which have been developed as part of the EcoPOST framework.

4   Modeling Guidelines

To further facilitate the use of our methodology, governing guidelines and best practices are provided. This section summarizes two categories of EcoPOST governing guidelines: (1) guidelines for evaluation models and (2) guidelines for simulation models.

In general, EcoPOST evaluation models can become large, e.g., due to a potentially high number of evaluation factors to be considered or due to the large number of causal dependencies existing between them. To cope with this complexity, we introduce guidelines for designing evaluation models (cf. Table 3). Their derivation is based on experiences we gathered during the development of our approach, its initial use in practice, and our study of general System Dynamics (SD) guidelines [10]. As example consider guideline EM-1 from Table 3. The distinction between SCFs and DCFs is a fundamental principle in the EcoPOST framework. Yet, it can be difficult for the user to decide whether a cost factor shall be considered as static or dynamic. As example take an evaluation scenario which deals with the introduction of a new EIS "CreditLoan" to support the granting of loans at a bank. Based on the new EIS, the entire loan offer process shall be supported. For this purpose, the EIS has to leverage internal (i.e., within the bank) and external (e.g., a dealer) trading partners as well as other legacy applications for customer information and credit ratings. Among other things, this necessitates the integration of existing legacy applications. In case this integration is done by external suppliers, resulting costs can be represented as SCFs as they can be clearly quantified

Table 3. Guidelines for Designing Evaluation Models

GL	Description
EM-1	Carefully distinguish between SCFs and DCFs.
EM-2	If it is unclear how to represent a given cost factor represent it as SCF.
EM-3	Name feedback loops.
EM-4	Use meaningful names (in a consistent notation) for cost and impact factors.
EM-5	Ensure that all causal links in an evaluation model have unambiguous polarities.
EM-6	Choose an appropriate level of detail when designing evaluation models.
EM-7	Do not put all feedback loops into one large evaluation model.
EM-8	Focus on interaction rather than on isolated events when designing evaluation models.
EM-9	An evaluation model does not contain feedback loops comprising only auxiliary variables.
EM-10	Perform empirical and experimental research to generate needed data.

based on a contract or service agreement. If integration is done in-house, however, integration costs should be represented as DCFs as costs might be influenced by additional ImFs in this case. Other guidelines are depicted in Table 3.

To simulate EcoPOST evaluation models constitutes another complex task. The guidelines from Table 4 are useful to deal with it. Guideline SM-7, for example, claims to assess the usefulness of an evaluation model and related simulation results always in comparison with mental or descriptive models needed or used otherwise. In our experience, there often exists controversy on the question whether an evaluation model meets reality. However, such controversies miss the first purpose of a model, namely to provide insights that can be easily communicated.

Table 4. Guidelines for Developing Simulation Models

GL	Description
SM-1	Ensure that all equations of a simulation model are dimensionally consistent.
SM-2	Do not use embedded constants in equations.
SM-3	Choose appropriately small time steps for simulation.
SM-4	All dynamic evaluation factors in a simulation model must have initial values.
SM-5	Use appropriate initial values for model variables.
SM-6	Initial values for rate variables need not be given.
SM-7	The validity of evaluation models and simulation outcomes is a relative matter.

The sketched governing guidelines and best practices represent a basic set of clues and recommendations for users of the EcoPOST framework. They support the modeler in designing evaluation models, in building related simulation models, and in handling dynamic evaluation factors. Yet, it is important to mention that the consideration of these guidelines does not automatically result in better evaluation and simulation models or in the derivation of more meaningful evaluation results. Notwithstanding, taking the guidelines increases the probability of developing meaningful models.

5   Case Study

In previous work, we already showed how experimental [**?**] and empirical research [**?**,13,14,15] contributes to the derivation of good quality evaluation models. This section summarizes results from an additonal case study, this time focusing on the overall applicability of the EcoPOST framework. We also discuss model validation from a more

general viewpoint. Due to space limitations we cannot decsribe the complete case study in detail (for details see [9]).

5.1 Research Design

We apply the EcoPOST framework to a complex EIS engineering project from the automotive domain in which we participated. We investigate cost overruns observed during the introduction of a large information system for supporting the development of *electrical and electronic (E/E) systems* (e.g., a multimedia unit in the car). Based on real project data, interviews with project members (e.g., requirements engineers, software architects, software developers), online surveys among end users, and practical experiences gathered in the respective EIS engineering project, we develop a set of EcoPOST evaluation models and analyze these models using simulation.

An initial business case for the considered EIS engineering project is developed prior to project start in order to convince senior management to fund the project. This business case is based on data about similar projects provided by competitors (evaluation by analogy) as well as on rough estimates on planned costs and assumed benefits of the project. The business case comprises six main cost categories: (1) *project management*, (2) *process management*, (3) *IT system realization*, (4) *specification and test*, (5) *roll-out and migration*, and (6) *implementation of interfaces*.

In a first project review (i.e., measurement of results), it turns out that originally planned project costs are not realistic, i.e., cost overruns are observed – particularly concerning cost categories (2) and (3). In our case study, we analyze cost overruns in three cost categories in detail using the EcoPOST methodology (cf. Table 5).

To be able to build evaluation and simulation models for the three analyzed cost categories, we need to collect data. This data is based on four information sources (cf. Table 6), which allow us to identify relevant cost and impact factors, i.e., evaluation factors that need to be included in the evaluation models to be developed. Likewise, the information sources also enable us to spot important causal dependencies between cost and impact factors and to derive evaluation models.

Table 5. Analyzed Cost Categories

	Description
1	**Process Management Costs**: This category deals with costs related to the (re)design of the business processes to be supported. This includes both the definition of new and the redesign of existing processes. As example of a process to be newly designed consider an E/E data provision process to obtain needed product data. As example of an existing process to be redesigned consider the basic E/E release management process. Among other things, process management costs include costs for performing interview-based process analysis and costs for developing process models.
2	**IT System Realization Costs**: This category deals with costs for implementing the new EIS on top of process management technology. In our case study, we focus on the analysis of costs related to the use of the process management system, e.g., costs for specifying and implementing the business functions and workflows to be supported as well as costs for identifying potential user roles and implementing respective access control mechanisms.
3	**III. Online Surveys among End Users**: We conduct two online surveys among two user groups of the new EIS (altogether 80 survey participants). The questionnaires are distributed via a web-based delivery platform. They slightly vary in order to cope with the different work profiles of both user groups. Goal of the survey is to confirm the significance of selected ImFs like "End User Fears" and "Emotional Resistance of End Users".
IV	**Specification and Test Costs**: This cost category sums up costs for specifying the functionality of the EIS as well as costs for testing the coverage of requirements. This includes costs for eliciting and documenting requirements as well as costs for performing tests on whether requirements are met by the EIS.

Table 6. Data Collection

	Description
I	*Project Data*: A first data source is available project data; e.g., estimates about planned costs from the initial business case. Note that we did not participate in the generation of this business case.
II	*Interviews*: We interview 10 project members (2 software architects, 4 software developers, 2 usability engineers, and 2 consultants participating in the project). Our interviews are based on a predefined, semi-structured protocol. Each interview lasts about 1 hour and is accomplished on a one-to-one basis. Goal of the interviews is to collect data about causal dependencies between cost and impact factors in each analyzed cost category.
III	*Online Surveys among End Users*: We conduct two online surveys among two user groups of the new EIS system (altogether 80 survey participants). The questionnaires are distributed via a web-based delivery platform. They slightly vary in order to cope with the different work profiles of both user groups. Goal of the survey is to confirm the significance of selected ImFs like "End User Fears" and "Emotional Resistance of End Users".
IV	*Practical Experiences*: Finally, our evaluation and simulation models also build upon practical experiences we gathered when participating in the investigated EIS engineering project. We have worked in this project as requirement engineers for more than one year and have gained deep insights during this time. Besides the conducted interviews, these experiences are the major source of information when designing our evaluation models.

5.2   Lessons Learned

Based on the derived evaluation models and simulation outcomes, we have been able to show that costs as estimated in the initial business case are not realistic. The simulated costs for each analyzed cost category exceed the originally estimated ones. Moreover, our evaluation models provide valuable insights into the reasons for the occurred cost overruns, particularly into causal dependencies and resulting effects on the costs of the analyzed EIS engineering project.

Table 7. Lessons Learned

LL	Description
LL-1	Our case study confirms that the EcoPOST framework enables EIS engineers to gain valuable insights into causal dependencies and resulting cost effects in EIS engineering projects.
LL-2	EcoPOST evaluation models are useful for domain experts and can support IT managers and policy makers in understanding an EIS engineering project and decision-making.
LL-3	EIS engineering projects are complex socio-technical feedback systems which are characterized by a strong nexus of organizational, technological, and project-specific parts. Hence, all evaluation models include feedback loops.
LL-4	Our case study confirms that evaluation models can become complex due to the large number of potential SCFs, DCFs and ImFs as well as the many causal dependencies existing between them. Governing guidelines (cf. Section 4) help to avoid too complex evaluation models.
LL-5	Though our simulation models have been build upon data derived from four different data sources, it has turned out that it is inevitable to rely on hypotheses to build simulation models.

Regarding the overall goal of the case study, i.e., the investigation of the practical applicability of the EcoPOST framework and its underlying evaluation concepts, our experiences confirm the expected benefits. More specifically, we can summarize our experiences by means of five lessons learned (cf. Table 7).

6   Summary

This paper summarizes the EcoPOST cost analysis methodology, a practically approved, model-based methodology to better understand and systematically investigate the complex cost structures of EIS engineering projects. We sketch our qualitative

EcoPOST methodology, introduce model design rules and describes modeling guidelines. We also summarize a case study illustrating the benefits of our approach.

Currently, our methodology is used in various information system engineering projects, mainly in the automotive domain. In future, we want to further validate our approach and aim at increasing the number of EcoPOST evaluation patterns [9].

References

1. Reijers, H.A., van der Aalst, W.M.P.: The Effectiveness of Workflow Management Systems - Predictions and Lessons Learned. Int'l. J. of Inf. Mgmt. 25(5), 457–471 (2005)
2. 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)
3. 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)
4. Mutschler, B., Zarvic, N., Reichert, M.: A Survey on Economic-driven Evaluations of Information Technology. Technical Report, TR-CTIT-07, University of Twente (2007)
5. Mutschler, B., Reichert, M., Rinderle, S.: Analyzing the Dynamic Cost Factors of Process-aware Information Systems: A Model-based Approach. In: 19th CAiSE, pp. 589–603 (2007)
6. Richardson, G.P., Pugh, A.L.: System Dynamics - Modeling with DYNAMO (1981)
7. Forrester, J.W.: Industrial Dynamics, Industrial Dynamics. Productivity Press (1961)
8. Ventana Systems, Inc.: Vensim (2006), http://www.vensim.com/
9. Mutschler, B.: Analyzing Causal Dependencies on Process-aware Information Systems from a Cost Perspective. PhD Thesis, University of Twente (2008)
10. Sterman, J.D.: Business Dynamics. McGraw-Hill, New York (2000)
11. Mutschler, B., Weber, B., Reichert, M.: Workflow Management versus Case Handling: Results from a Controlled Software Experiment. In: Proc. ACM SAC, Special Track on Coordination Models, Languages and Architectures, pp. 82–89 (2008)
12. Mutschler, B., Reichert, M., Bumiller, J.: Unleashing the Effectiveness of Process-Oriented Information Systems: Problem Analysis, Critical Success Factors, and Implications. IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews 38(3), 280–291 (2008)
13. Mutschler, B., Rijkpema, M., Reichert, M.: Investigating Implemented Process Design: A Case Study on the Impact of Process-aware Information Systems on Core Job Dimensions. In: Proc. 8th Int'l. BPMDS Workshop, pp. 379–384 (2007)
14. Mutschler, B., Reichert, M.: A Survey on Evaluation Factors for Business Process Management Technology. Technical Report, TR-CTIT-06-63, University of Twente (2006)
15. Mutschler, B., Reichert, M., Bumiller, J.: Why Process-Orientation is Scarce: An Empirical Study of Process-oriented Information Systems in the Automotive Industry. In: Proc. 10th IEEE EDOC, pp. 433–438 (2006)