Extended Influence Diagrams for Enterprise Architecture Analysis

Pontus Johnson, Robert Lagerström, Per Närman and Mårten Simonsson
Royal Institute of Technology
Stockholm, Sweden
Email: {pj101, robertl, pern, martens}@ics.kth.se

Abstract

The discipline of enterprise architecture advocates the use of models to support decision-making on enterprise-wide information system issues. In order to provide such support, enterprise architecture models should be amenable to analyses of various properties, as e.g. the level of enterprise information security. This paper proposes the use of a formal language to support such analysis. Such a language needs to be able to represent causal relations between, and definitions of, various concepts as well as uncertainty with respect to both concepts and relations. To support decision-making properly, the language must also allow the representation of goals and decision alternatives. This paper evaluates a number of languages with respect to these requirements, and selects influence diagrams for further consideration. The influence diagrams are then extended to fully satisfy the requirements. The syntax and semantics of the extended influence diagrams are detailed in the paper, and their use is demonstrated in an example.

1 Introduction

During the last decade, enterprise architecture has grown into an established approach for holistic management of the information systems in an organization. Enterprise architecture is model based, in the sense that diagrammatic descriptions of the systems and their environment constitute the core of the approach. A number of enterprise architecture initiatives have been proposed, including The Open Group Architecture Framework (TOGAF) [7], the Zachman Framework [15], Intelligrid [2] and more.

What constitutes a "good" enterprise architecture model has thus far not been clearly defined. This due to the fact that the "goodness" of the model is not an inherent property, but contingent on the purpose the model is intended to fill, i.e. what kind of analyses it will be subjected to. For instance, if one seeks to employ an enterprise architecture model for evaluating the usability of an information system, the information required from the model differs radically from the case when the model is used to evaluate system interoperability.

Enterprise architecture analysis is the application of property assessment criteria on enterprise architecture models. For instance, one investigated property might be the information security of an information system and a criterion for assessment of this property might be "If the architectural model of the enterprise features an intrusion detection system, then this indicates a higher level of information security than if there is no such system." Criteria and properties such as these may be extracted from scientific theories, or from emprical measurements. In current practice, employed criteria are rarely expressed explicitly. If they are, they are frequently both vague and ambiguous in nature, making enterprise architecture analysis anything but precise and accurate.

To ameliorate such analyses, a stringent method for representing the assessment criteria would be useful. Furthermore, the analyses would benefit greatly if the considered property could be evaluated in a quantitative fashion. A notation for expressing the preference critera in an unambigous manner coupled with an inference mechanism for making deductions would fulfill these demands. This article proposes the *extended influence diagrams* as such a language. Employing the probabilistic inference mechanism of Bayesian networks, extended influence diagrams allow the analysis of a wide range of important properties of enterprise architecture models.

Section 2 presents the context of enterprise architecture-based decision making. This leads to a set of requirements on potential languages for the support of such decision making, presented in Section 3. Section 4 considers to what extent a number of existing languages fulfill these requirements. Influence diagrams turn out as the most suitable candidates, so these are presented in Section 5. However, as influence diagrams do not fulfill all requirements, Section 6 proposes a set of extensions to these in order to increase the usefulness with respect to the IT decision making context. Section 7 demonstrates the construction and use of

these extended influence diagrams. The article is concluded in Section 8.

2 Enterprise architecture analysis

The purpose of having enterprise architecture models and conducting analyses of these is to facilitate the making of rational decisions about information systems. A rational decision maker is an agent who has to choose an alternative after a process of deliberation in which he or she answers three questions: "What is feasible?", "What is desirable?" and "What is the best alternative according to the notion of desirability, given the feasibility constraint?" [9]. Translating these questions into the context of enterprise architecture analysis, architectural scenarios answer the first question of what is feasible. Answers to the second question regarding desirables, are often expressed in terms of the degree of various information system properties such as high information security, high interoperability, high availability, etc. The answer to the third question, providing the link between the feasible to the desirable, i.e. the link between the scenarios and the properties of interest, is given by architectural analysis.

Decision making, whether the decisions apply to IT or not, is rarely performed under conditions of complete certainty. One fundamental uncertainty is regarding the definition of various concepts. This is called *definitional uncertainty*. For instance, some authors define the term information security in terms of confidentiality, integrity and availability, [5] whereas others add the concept of non-repudiation and accounting to the definition [8].

Related to definitional uncertainty is *theoretical heterogeneity*. Existing knowledge regarding the nature of enterprise information systems is not consolidated within a single commonly accepted framework, as may be the case for more mature disciplines. A consequence of this is that there is a need to relate similar concepts to each other, and to relate concepts at varying levels of abstraction to each other.

In addition to definitional uncertainty, there may also be uncertainty with regard to how the world actually behaves. Knowledge of exactly how various properties causally affect one another is seldom certain. There is a level of *causal uncertainty*. An example of this would be uncertainty with respect to the causal effect the percentage of systems with updated virus protection may have on the information security.

When creating enterprise architecture models of the enterprise, the information represented in the models is normally associated with a degree of uncertainty. Perhaps the information was collected a while ago and has now become obsolete, or perhaps the information was extracted from a document that might have been incorrect. In these situations, the decision maker suffers from *empirical uncertainty*.

From this description of the context of enterprise architecture analysis, requirements may be extracted on the languages used for analysis specification. Such requirements are presented in the next section.

3 Requirements on language

The previous section presented the context for which a language for enterprise architecture analysis is sought. Summarizing, this context is a decision-making situation characterized by three kinds of uncertainty: definitional uncertainty, causal uncertainty, and empirical uncertainty. Also, the language needs to have facilities for managing theoretical heterogeneity.

A language appropriate for this context needs to be able to represent a number of things. Firstly, basic decision-making support requires that the notation is able to represent the decision maker's goals, domain of control, as well as the causal relations between that which can be controlled and the desired goals. Secondly, in order to manage definitional uncertainty, the language needs to be able to represent what is meant by the constituent concepts. Thirdly, the notation needs be able to express causal and empirical uncertainty, i.e. uncertainty with respect to how concepts affect each other and with respect to the measured values of the concepts. Finally, in order to manage heterogeneous theory, the language needs to allow multiple levels of abstraction.

4 Evaluation of existing languages

A number of languages have already been proposed for various decision-making contexts. In this section, we will consider a number of those languages with respect to the requirements presented in the previous section.

For probabilistic inference, Bayesian networks are often employed [6]. Bayesian networks are composed of nodes with associated values, and arcs between the nodes. The nodes' probabilistic dependencies on each other are specified by the arcs and by so called conditional probability distributions. With respect to our requirements, Bayesian networks provide support for modeling causality, causal uncertainty and empirical uncertainty. There is, however, no support for goals, controllable variables, definitional relations, or multiple abstraction levels. Bayesian networks have been extended with utility and decision nodes in order to better support rational decision making, i.e. to represent goals and domains of control. Diagrams that feature such nodes are called influence diagrams [11, 12]. In order to easier view the world on multiple levels of abstraction, object-oriented influence diagrams have been proposed [3]. These allow concepts to be aggregated into objects, which

can be expanded and collapsed. As will be explained in Section 6, however, object-oriented influence diagrams do not offer support for avoiding definitional uncertainty.

An alternative to the abovementioned Bayesian approaches is Dempster-Shafer theory [13, 14]. Dempster-Shafer is specifically designed to allow the representation of ignorance, something that the Bayesian formalisms only are capable of to a limited degree. Dempster-Shafer Theory, however, lacks support for decision and goal representation, as well as support for managing multiple levels of abstraction or defining new concepts. The perhaps best known modeling notation in the information systems area is the Unified Modeling Language (UML) [1]. The class diagrams in this language explicitly supports the specification of definitional relations and multiple abstraction levels, but no other of the requirements presented in Section 3.

	Bayesian Networks	Influence Diagrams	Object-Oriented ID	Dempster-Shafer	UML
Goals		✓	✓		
Domain of Control		✓	✓		
Causality	✓	✓	✓	✓	
Definitional Uncertainty					✓
Causal Uncertainty	✓	✓	✓	✓	
Empirical Uncertainty	✓	✓	✓	✓	
Multiple levels of abstraction			✓		✓

Figure 1. Degree of requirement fulfillment by existing languages. The Y-axis consists of the different requirements as specified in Section 3 and the X-axis contains the various languages as described in the present chapter. Dotted "check"-notation denotes partial fulfillment.

Figure 1 summarizes the respective notations' degree of fulfillment with respect to the considered requirements. As the figure shows, influence diagrams and object oriented influence diagrams are the most suitable candidates. These are therefore presented in greater detail in the next section.

5 Influence diagrams

This section briefly describes conventional influence diagrams. For a more comprehensive treatment, the reader is referred to [11], [12] , and [6].

5.1 Formal definition of influence diagrams

An *influence diagram* is a network used for modelling uncertain variables and decisions, consisting of a directed graph $G = (N, A)$. There are three types of nodes in the set N, partitioned into the sets V, C and D. There is at most one *utility* node $v \in V$, the utility node is depicted as a rhombus (Figure 2). There are zero or more *chance* nodes $c \in C$, these are depicted as ovals. There may be zero or more *decision* nodes $d \in D$, and these are depicted as squares.

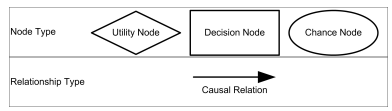

Figure 2. Utility, chance, and decision nodes in influence diagrams.

There are two types of arcs in the set A, partitioned into the sets K and J. Arcs into utility and chance nodes are *causal*, $\kappa \in K$, representing probabilistic dependence. Arcs into decision nodes are informational, $\iota \in I$, and imply time precedence. Both the causal relation arcs and the informational relation arcs are depicted as arrows. See Figure 2.

The set of causal predecessors are represented by $C(i) = \{j \in N : (j, i) \in K\}$, where $i \in \{V, C\}$. The set of informational predecessors are represented by $I(i) = \{j \in N : (j, i) \in I\}$, where $i \in D$.

Associated with each node i is a variable X_i and a set Ω_i of possible values it may assume. If i is the utility node, then X_i represents the expected utility and its domain Ω_i is a subset of the real line. If i is a chance node, then Ω_i is the sample space for the random variable X_i. Finally, decision node i has alternative X_i chosen from the set i. The utility node $v \in V$ has an associated utility function $U : \Omega_{C(v)} \rightarrow \Omega_v$, which represents the expected utility as a function of the values of the conditioning predecessors of the utility node. There is a conditional probability distribution π_i for every chance node i, given the values of its causal predecessors, $\pi(x_i|x_{C(i)}) = Pr\{x_i|x_{C(i)}\}$. The notation $\Pr\{x_i|x_{C(i)}\}$ means that $\Pr\{X_i = x_i|X_{C(i)} = x_{C(i)}\}$.

The probability distributions for each chance node are represented in conditional probability matrices. Table 1 shows the conditional probability matrix for a chance node y dependent on a node z.

Object-oriented influence diagrams [3] follow the same rules that influence diagrams do, except that they also provide support for composition of sub-diagrams. The main diagram may thus consist of a set of sub-diagrams, all consisting in turn of nodes or sub-sub-diagrams. Nodes may be related between sub-diagrams.

Table 1. A conditional probability matrix.

Z		$z1$	$z2$
Y	$y1$	$Pr(y1\|z1)$	$Pr(y1\|z2)$
	$y2$	$Pr(y2\|z1)$	$Pr(y2\|z2)$

5.2 Using influence diagrams for enterprise architecture analysis

Although not as potent as the extended version presented below, conventional influence diagrams can be used for architecture analysis. Recalling the decision making approach described in Section 2, the question "What is desirable?" may be specified in the influence diagram utility node, $v \in V$ (see Figure 3). An example of an utility node concept for enterprise architecture analysis is "information security".

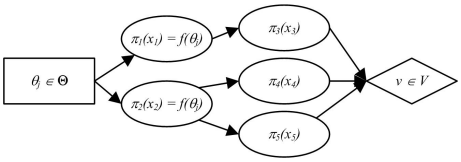

Figure 3. The relation between scenario selection, content of the architecture scenarios, intermediary chance nodes, and the utility node.

The answer to "What is feasible?" is in the case of architecture-based decision making answered by a set of enterprise architecture scenarios,$\Theta = \{\theta_1, ...\theta_n\}$. Values of entities in the architectural scenarios may then be used to determine the conditional probability distributions for chance nodes in the influence diagram, $\pi_i(x_i) = f(\theta_i)$. The decision maker's choice between architecture scenarios is represented by a decision node, $\theta \in \Theta$, which is connected by causal relations to all those chance nodes that the scenario selection might potentially affect.

Finally and most importantly, the chance nodes that are affected by the decision node now need to be causally related to the utility node so that the utility of potential scenarios can be calculated. Exactly how these chance nodes may be related to the utility node is an important issue and will be adressed later in this paper. Typically, this is performed by the use of intermediary chance nodes. The resulting influence diagram is a representation of how we believe that the real world functions; it is the representation of a causal theory of, for instance, information security. The resulting influence diagram is the answer to the decision maker's third question, "What is the best alternative according to the notion of desirability, given the feasibility constraint?"

6 Extended influence diagrams

Although influence diagrams may be used for enterprise architecture analysis in their conventional form, there are, as mentioned in Section 4 and Section 5, some requirements that are not sufficiently addressed. This section therefore presents a set of extensions to influence diagrams for the purposes of architecture analysis.

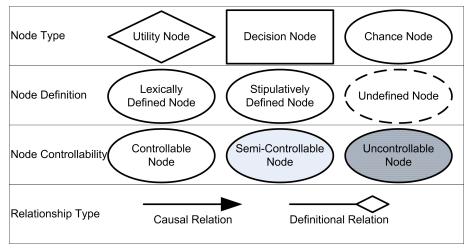

Figure 4. The syntax of extended influence diagrams.

6.1 Lexically defined nodes

Causality means that one phenomenon in the real world somehow affects another. Because causality is a concept of the real world, it is important that there is a mapping between the real world and the influence diagrams, i.e. that the concepts presented in the influence diagrams are well-defined. If they are not well-defined, it will be theoretically impossible to determine whether there is in fact any truth to the causal relation between the phenomena.

The definitions of some nodes in influence diagrams are deemed uncontroversial. For instance, in the IT-community, we might assume that there is common agreement on the definition of the concept of memory size; it is typically measured in terms of bytes, etc. Nodes that we deem uncontroversial are called *lexically defined nodes*, $L \subset N$ (see Figure 4) [10].

6.2 Stipulatively defined and undefined nodes

In the considered decision-making context, we have assumed that there is considerable confusion as to the meaning of concepts, such as information security. There are thus many potential nodes that are not lexically defined. In order to manage these, we introduce the possibility to define nodes within the influence diagram. This is done by relating a node, directly or indirectly, to lexically defined nodes with a new kind of arc, called a definitional relation, $\delta \in \Delta$, where $\Delta \subset A$ (see Figure 4). Nodes that are defined in

the diagram using the definitional relation are called *stipulatively defined nodes* $S \subset N$ [10]. Those nodes that are neither lexically nor stipulatively defined are simply *undefined*, $U \subset N$.

6.3 Definitional relations

This subsection details the semantics of the definitional relations, Δ. The set of defining nodes is given by $D(i) = \{j \in N : (j,i) \in \Delta\}$. The definitional relation implies simple aggregation; the defined node is nothing more than its constituent parts. It is represented mathematically using the same conditional probability distributions as are used for causal relations, $\pi_i(x_i|x_{D(i)}) = Pr\{x_i|x_{D(i)}\}$. Since definitional relations are simple aggregations, a node that has defining nodes cannot also in the same graph feature causal predecessors.

The concept of definitional relations is fairly closely related to the aggregation mechanism provided by object-oriented influence diagrams. However, aggregated objects in object oriented influence diagram's are to be viewed more as placeholders for diagrams than as nodes. They are therefore not associated with any variable X_i. As is considered below, the association of a variable with the aggregated node is of significant importance for our purposes.

6.4 Controllability of nodes

A utility node may be affected by many chance nodes, and it may be the case that only some of these chance nodes are affected (directly or indirectly) by the decision node. Those nodes that lie in a causal path from decision node to utility node or are stipulatively defined by such nodes are relevant for decision making. Other nodes are not. Nodes that are completely dependent on the decision node are denoted *controllable nodes*, $F \subset N$. Nodes that are completely independent of the decision nodes are denoted *uncontrollable nodes*, $W \subset N$. Finally, nodes that are partially dependent on the decision node are denoted *semi-controllable nodes*, $SC \subset N$. The graphical rendering of these node types is given in Figure 4. Furthermore, it may sometimes be useful to distinguish between *directly controllable* nodes, $DC \subset F$ and *indirectly controllable nodes*, $IDC \subset F$. Directly controllable nodes are targets of a causal relation originating in a decision node, $DC(i) = \{j \in D : (j,i) \in K\}$. Indirectly controllable nodes are targets of a causal relation originating in a directly controllable node or an indirectly controllable node, $IDC(i) = \{j \in \{DC, IDC\} : (j,i) \in K\}$.

6.5 A set of related influence diagrams

One major benefit of definitional relations is that it is possible to treat a topic on different levels of abstraction. It is thus possible to speak abstractly of the causal effect that for instance firewall protection has on information security, and more concretely about the effect it has on confidentiality, integrity and availability. These two levels of abstraction are presented in Figure 5 in terms of two different graphs.

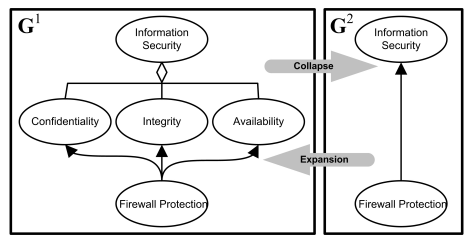

Figure 5. An extended influence diagram is a set of conventional influence diagrams linked by node collapse and node expansion transformations.

We define an extended influence diagram as a set of graphs, $E = \{G^1, ..., G^n\}$, related by expansions and collapses of the definitional structures, where the expansions and collapses are dictated by specific transforms, $G^j = coll(G^i)$ and $G^i = exp(G^j)$. In Figure 5, $G^2 = coll(G^1)$ and $G^1 = exp(G^2)$. The transforms are required to ensure that the graphs do not contradict each other. In the rest of this subsection, these transforms are detailed.

Firstly, the effects of node collapses on the affected conditional probability distributions are considered. In Figure 6, given a graph G^1 with an expanded node i, we need to know how to represent the conditional probability distributions of the nodes affected by the collapse, namely node i and nodes $k_1, ..., k_q$ in G^2.

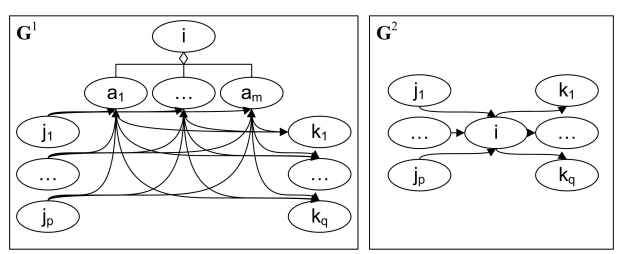

Figure 6. Node collapse.

Using the law of total probability, $P(A) = P(A|B)P(B) + P(A| \sim B)P(\sim B)$, we find that the conditional probability distribution for node i in the initial expanded diagram is

$$
\begin{aligned}
&Pr\{x_i|x_{j1},...,x_{jp}\} \\
&\quad = \sum_{x_{a1},...,x_{am}} Pr\{x_i|x_{a1},...,x_{am}\} \\
&\quad \cdot \prod_{\alpha=a1..am} Pr\{x_\alpha|x_{j1},...,x_{jp}\}
\end{aligned}
$$

The notation $\sum_{x_1,...,x_n}$ means the sum as the variables $x_1,, x_n$ go through all possible values in their corresponding spaces. For node $\kappa \in \{k_1, ..., k_q\}$, we have

$$
\begin{aligned}
&\pi_\kappa(x_\kappa|x_i) = Pr\{x_\kappa|x_i\} \\
&\quad = \sum_{x_{a1},...,x_{am}} Pr\{x_\kappa|x_{a1},...,x_{am}\} \\
&\quad \cdot \prod_{\alpha=a1..am} Pr\{x_\alpha|x_i\}
\end{aligned} \tag{1}
$$

The first term in equation 1 is given. With Bayes' theorem, $P(A|B) = P(B|A)P(A)/P(B)$, the second term can be calculated,

$$
Pr\{x_\alpha|x_i\} = \frac{Pr\{x_i|x_\alpha\}Pr\{x_\alpha\}}{Pr\{x_i\}} \tag{2}
$$

All three terms in this equation need to be expanded; the law of total probability can be used for this purpose.

$$
\begin{aligned}
&Pr\{x_i|x_\alpha\} \\
&\quad = \sum_{\substack{x_{a1},...,x_{\alpha-1} \\ x_{\alpha-1},...,x_{am}}} Pr\{x_i|x_{a1},...,x_{am}\} \\
&\quad \cdot \prod_{\substack{\chi=a1,...,am \\ \chi\neq\alpha}} Pr\{x_\chi\}
\end{aligned}
$$

$$
\begin{aligned}
&Pr\{x_i\} \\
&\quad = \sum_{x_{a1},...,x_{am}} Pr\{x_i|x_{a1},...,x_{am}\} \\
&\quad \cdot \prod_{\alpha=a1..am} Pr\{x_\alpha\}
\end{aligned}
$$

$$
\begin{aligned}
&Pr\{x_\alpha\} = \\
&\quad = \sum_{x_{j1},...,x_{jp}} Pr\{x_\alpha|x_{j1},...,x_{jp}\} \\
&\quad \cdot \prod_{\eta=j1..jp} Pr\{x_\eta\}
\end{aligned}
$$

$Pr\{x_\eta\}$ for nodes $\eta \in \{j_1, ..., j_p\}$ need to be provided as prior probabilities.

The most expanded graph is considered as the "true" graph, with which the more collapsed versions need to comply according to the node collapse transformations above. In the case where there is already an expanded graph in the graph set E, a node expansion is simply a return to that graph. During diagram construction, it may however also be convenient to have access to a node expansion transformation, $G^1 = exp(G^2)$, cf. Figure 6. In this case, we assume that $\pi_i(x_i|x_{a1}, ..., x_{am})$ is given. We then need to know how to represent the conditional probability distributions of the nodes affected by the expansion, namely nodes $a_1, ..., a_m$ and nodes $k_1, ..., k_q$. For $\alpha \in \{a_1, ..., a_m\}$,

$$
\begin{aligned}
&Pr\{x_\alpha|x_{j1},...,x_{jp}\} = \\
&\quad = \sum_{x_i\in\Omega_i} Pr\{x_\alpha|x_i\} \cdot Pr\{x_i|x_{j1},...,x_{jp}\}
\end{aligned}
$$

$Pr\{x_i|x_{j1}, ..., x_{jp}\}$ is given, and $Pr\{x_\alpha|x_i\}$ was calculated in Equation 2. For $\kappa \in \{k_1, ..., k_q\}$,

$$
\begin{aligned}
&Pr\{x_\kappa|x_{a1},...,x_{am}\} = \\
&\quad = \sum_{x_i\in\Omega_i} Pr\{x_\kappa|x_i\} \cdot Pr\{x_i|x_{a1},...,x_{am}\}
\end{aligned}
$$

Summarizing, if we have $\pi_{kp}(x_{kp}|x_{a1}, ..., x_{am})$, $\pi_i(x_i|x_{a1}, ..., x_{am})$, $\pi_\alpha(x_\alpha|x_{j1}, ..., x_{jp})$, and $\pi_\eta(x_\eta), \eta \in \{j_1, ..., j_p\}$, for graph G^1, we can calculate $\pi_\kappa(x_\kappa|x_i), \kappa \in \{k_1, ..., k_q\}$ and $\pi_i(x_i|x_{j1}, ..., x_{jp})$ for graph G^2. And vice versa (in the case where G^1 does not already exist), given $\pi_\kappa(x_\kappa|x_i)$, $\pi_i(x_i|x_{j1}, ..., x_{jp})$, and $\pi_\eta(x_\eta)$ for graph G^2 we can calculate $\pi_\kappa(x_\kappa|x_{a1}, ..., x_{am})$, $\pi_i(x_i|x_{a1}, ..., x_{am})$, and $\pi_\alpha(x_\alpha|x_{j1}, ..., x_{jp})$ for graph G^1.

7 Using extended influence diagrams for information security analysis

This chapter describes the construction of extended influence diagrams and the use of them for enterprise architecture analysis. The former is described in Section 7.1 and aims at demonstrating not only the development process in itself, but also the expressive possibilities of the extended influence diagram language. The user perspective is described in Section 7.2 below, where the developed extended influence diagram is used by a decision maker. The context is that of Section 2, so the decision maker in question is an IT decision maker. The example relates to information security and to the ISO 17799 standard [4].

7.1 Development of extended influence diagrams

A process for the development of extended influence diagrams is described in this section and in Figure 7 below.

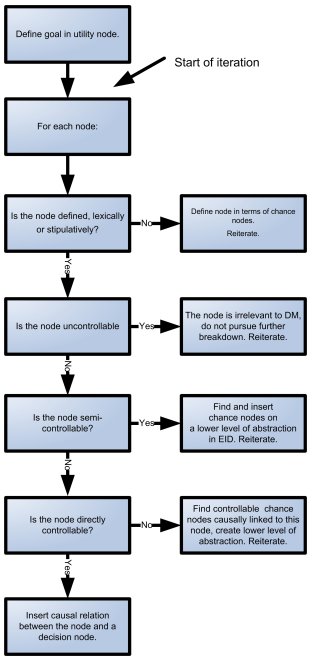

Figure 7. The process of developing extended influence diagrams.

When developing an extended influence diagram, the first thing to decide upon is the utility node. The process then consists of the iterative identification of nodes that either define or causally affect the parent node. This process continues until the identified nodes' relation to the domain of control, i.e. to the decision node, is made clear.

7.1.1 The utility node

We start with the assumption that the general goal of the IT decision maker is to maximize the company profit. *Profit* thus constitutes the utility node, see Figure 8.

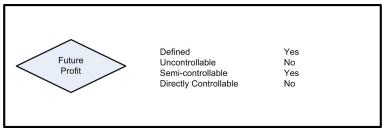

Figure 8. The utility node.

Following the flowchart of Figure 7, we assume that the term profit is lexically defined; for the purpose of this example the ambiguousness inherent in the concept will not cause any misunderstandings. The flowchart thus directs us to the subsequent questions, regarding controllability.

We posit that *Profit* is not uncontrollable; the actions taken by IT decision makers at a company causally affects the profits of the company. Moving on in the flowchart, we find that the concept is, however, semi-controllable in the sense that nodes not controlled by the IT decision maker have some level of impact on the utility node.

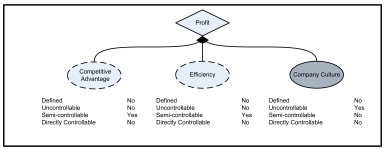

Figure 9. Profit is assumed causally affected by Competitive Advantage, Efficiency and Company Culture.

7.1.2 Causally related nodes

It is not satisfactory to use a semi-controllable variable as a basis for decision making, since its value may be affected by phenomena that are unrelated to the decision situation at hand. The flowchart therefore suggests the identification and incorporation of new chance nodes with a causal impact on *Profit*. For the purpose of this example, the chance nodes *Competitive Advantage*, *Efficiency* and *Company Culture* are introduced. See Figure 9. As presented in Section 5.1, the utility function, U, detailing the relation between the *Profit* node and its causal predecessors must also be specified.

Restarting at the top of the flowchart for the new nodes, none of these concepts would normally be considered lexically defined, and as mentioned earlier, such nodes need to be defined stipulatively. To avoid an overly complicated example, we will assume that these nodes are lexically defined.

Still following the flowchart, we posit that the chance node *Company Culture* is uncontrollable; an IT decision maker is typically not in a position to significantly influence or change company culture. Therefore no more chance nodes will be added below this particular chance node.

The chance nodes *Competitive Advantage* and *Efficiency* are examples of chance nodes that are semi-controllable, i.e., just as for the node *Profit*, the decision maker has control over some predecessors that causally influence the behavior of these chance nodes, but not all of them.

7.1.3 Definitionally related nodes

A chance node that does not suffer from semi-controllability, yet has causal effects on *Competitive Advantage* and *Efficiency*, is *Information Security*. The node and the resulting conditional probability distributions for *Competitive Advantage* and *Efficiency* are thus introduced. In a more complete example, one would imagine many more

nodes similar to *Information Security*, such as *Performance*, *Interoperability*, etc.

Information security is not lexically defined, so we stipulatively define it in terms of *Confidentiality*, *Integrity* and *Availability*. We will, for the sake of the example, consider these new nodes as lexically defined. Availability could be defined in terms of annual uptime. Integrity could be defined as percentage of business-critical information destroyed or altered in an unauthorized manner. Confidentiality could be defined as percentage of business-critical information disclosed to unauthorized persons or processes.

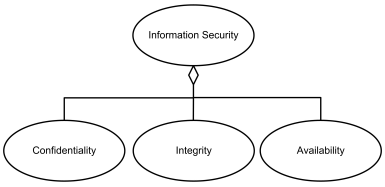

Figure 10. The chance node Information Security is stipulatively defined as Confidentiality, Integrity, and Availability.

With an eye on the flow chart, we believe that the three concepts are not only defined, but also (indirectly) controllable; the IT decision maker has considerable influence on their values. We have thus finally found concepts that are both defined and controllable. However, we do not think that any one of these three concepts are direct controllable, which is the stopping criterion of the flowchart.

7.1.4 An abstract extended influence diagram

One commonly proposed positive causal factor with respect to the Information Security in an enterprise is the company's ISO 17799 *Compliance*, i.e. the degree to which the company follows the rules specified by the international standard ISO/IEC 17799 "Information Technology - Security Techniques - Code of Practice for Information Security Management" [4]. However, it may be difficult to find detailed information on exactly how the constituent parts of *Information Security*, i.e. *Confidentiality*, *Integrity*, and *Availability*, are affected by *ISO 17799 Compliance*. In order to represent such abstract knowledge, the node *Information Security* can be collapsed, as represented in Figure 11. Continuing, assuming that the node *ISO 17799 Compliance* is both lexically defined and directly controllable, the flowchart prescribes the introduction of a decision node. A first, abstract version of the extended influence diagram is thus complete, cf. Figure 11.

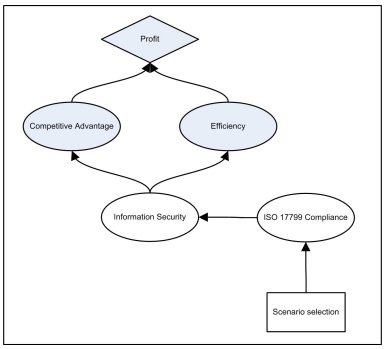

Figure 11. The information security chance node is defined and collapsed, the node is causally affected by the company's ISO 17799 compliance, which is directly controllable.

7.1.5 Refinement by node expansion

The assumption that *ISO 17799 Compliance* is lexically defined may be criticized, so we instead choose to refine this concept with a stipulative definition. Compliance can be measured by the degree to which the company fulfills requirements stated in individual chapters of the standard. In Figure 12 (which by no means aims for a complete description of the standard) *ISO 17799 compliance* is defined by four branches: The first is *Human Resources Security* which in turn is defined by the existence of a *User Training Process* to improve system user's security awareness. The second is *Communications and Operations Management*, which is definitionally linked to both the existence of a *Back-up Process* and *Anti-Virus Applications*. The third is *Information Systems Acquisition, Development and Maintenance*, definitionally linked to the existence of *Cryptographic Control Applications* and *Intrusion Detection Applications*. The fourth is *Information Security Incident Management*, defined by the existence of a formal *Incident Management Process*. As discussed in Section 6, the abstract causal effect of *ISO 17799 Compliance* on *Information Security* may now, after node expansion, be refined in terms of these more concrete defining nodes' causal effects on the *Confidentiality*, *Integrity*, and *Availability*.

Direct controllability and lexical definition is now attributed to the bottom nodes in the ISO 17799 Compliance branches. The refined version of the extended influence diagram is presented in Figure 12.

7.2 Enterprise architecture analysis with extended influence diagrams

In this section, the extended influence diagram introduced above is employed to illustrate the use of an extended

influence diagram for the purposes of enterprise architecture analysis.

Typically, the connection between corporate profit and more tangible concepts, such as information security, is a product of a corporate and IT strategic planning process. In order to separate concerns, it is common to address the more concrete aspects - information security, modifiability, performance, etc. - individually. This example considers the architectural analysis of information security. *Information Security* is therefore viewed as the utility node.

7.2.1 The Enterprise architecture metamodel

When enterprise architecting, a metamodel details what parts of the organization are to be modeled. In this example, the metamodel is limited to two entities, *processes* and *applications*. Here, it is assumed that each metamodel entity is associated with a probability representing the likelihood that the entity will in fact be implemented if the scenario is selected.

7.2.2 Enterprise architecture scenarios

In this example, the decision maker is faced with the choice between two architectural scenarios (Figure 13). The scenarios are similar insofar as they both include anti-virus and intrusion detection applications in addition to the company's incident management process. The scenarios differ as follows. In the first scenario, a user training process is introduced to increase the overall user security awareness. In the second scenario, cryptographic control applications are implemented to increase the confidentiality of the systems and a back-up process is introduced to improve the availability.

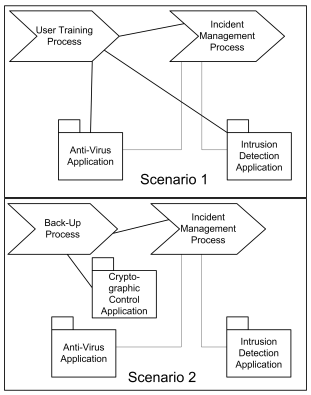

Figure 12. The two scenarios. The first introduces user training; the second introduces a back-up process and cryptographic control applications.

7.2.3 Conditional probability distributions of directly controllable nodes

If the metamodel is constructed to support the analysis, information from the scenarios can be used to determine the conditional probability distributions (matrices) for the directly controllable nodes of the extended influence diagram. Table 2, presenting the conditional probability matrix of the chance node *User Training Process*, exemplifies how the impact differs between Scenario 1 and 2 with respect to the existence of the user training process.

Table 2. The conditional probability distribution of the node User Training Process.

Scenario Selection		Scenario 1	Scenario 2
User Training Process	Yes	1	0.1
	No	0	0.9

In this example, the decision maker is confident that the user training process will be implemented if Scenario 1 is selected. However, if Scenario 2 is selected, there is a chance that the process will be implemented anyhow. Perhaps the decision maker has uncertain information about the existence of other user training initiatives within the company.

7.2.4 Calculating the results

Given the conditional probability distributions of the directly controllable nodes in Figure 12, $DC = \{\pi_{dc1}, ..., \pi_{dcn}\}$, the probability distributions of nodes *Confidentiality*, *Integrity* and *Availability*, e.g. $Pr\{x_{Cred} \mid x_{dc1}, ..., x_{dcn}\}$, may be calculated using the law of total probability. In a similar manner, the expected utility of the node *Information Security* may be assessed. Assuming that the domain of the node is $\Omega_{is} = \{x \in R : 0 < x < 5\}$, we may present the results on the form of Figure 14. The diagram indicates the level of information security as well as the uncertainty associated with the assessment. The diagram provides support for the decision maker in the decision between three alternatives: a) choose scenario 1, b) choose scenario 2, or c) increase certainty of results in order to provide better decision support for the selection between the scenarios. In order to increase the certainty of the results, we would need to engage either in more data collection, i.e. increasing the precision of the enterprise architecture model, or in more theory development, i.e. increasing the precision of the extended influence diagram.

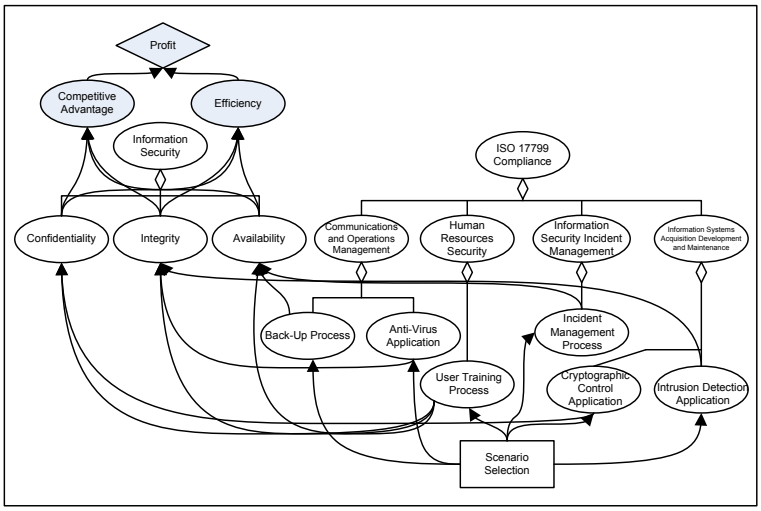

Figure 13. The Information Security and the ISO 17799 Compliance nodes are expanded.

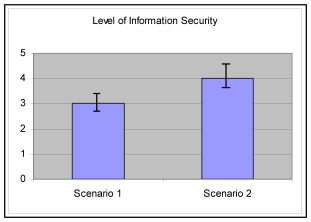

Figure 14. Grey bars show the utility expressed in terms of level of information security. The narrow black bars indicate degree of uncertainty of the assessments.

8 Conclusions

This article proposes the use of a language for the analysis of enterprise architectures. A number of candidate languages are considered but neither one is deemed satisfactory. The best candidate, the *influence diagram*, is therefore extended. The proposed *extended influence diagrams* differ from the conventional ones in their ability to cope with definitional uncertainty, i.e. the uncertainty associated with the use of language and in their ability to represent multiple levels of abstraction. The syntax and semantics of extended influence diagrams is detailed, and their use is demonstrated in an example. The proposed language supports quantitative analysis of important properties of enterprise architecture models, such as information security, performance, availability, and interoperability, providing both an assessment of the utility of the architecture and the uncertainty associated with that assessment.

References

[1] M. Fowler and K. Scott. *UML Distilled: Applying the Standard Object Modeling Language*. Addison-Wesley, 1997.

[2] J. Hughes. *The Integrated Energy and Communication Systems Architecture*. Electric Power Research Institute, 2004.

[3] Hugin Expert A/S. *HUGIN API Reference Manual, Version 6.3*, 2004.

[4] International Organization for Standardization. *ISO/IEC International Standard 17799:2000 Information Technology - Security Techniques - Code of Practice for Information Security Management*, 2000.

[5] P. Liu, P. Ammann, and S. Jajodia. Rewriting histories: Recovering from malicious transactions. *Distributed and Paralell Databases*, 8, 2000.

[6] R. Neapolitan. *Learning Bayesian Networks*. Pearson Education, 2004.

[7] The Open Group. *The Open Group Architecture Framework, Version 8*, 2005.

[8] S. Poslad and M. Calisti. Towards improved trust and security in fipa agent platforms. In *Autonomous Agents 2000 Workshop on Deception, Fraud and Trust in Agent Societies*, 2000.

[9] A. Rubenstein. *Modeling Bounded Rationality*. The MIT Press, 1998.

[10] M. Scriven. Definitions in analytical philosophy. *Philosophical Studies*, 5(3), 1954.

[11] R. Shachter. Evaluating influence diagrams. *Operations Research*, 34(6), 1986.

[12] R. Shachter. Probabilistic inference and influence diagrams. *Operations Research*, 36(4), 1988.

[13] G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.

[14] J.-B. Yang. Rule and utility based evidential reasoning approach for multiattribute decision analysis under uncertainties. *European Journal of Operations Research*, 133, 2001.

[15] J. Zachman. A framework for information systems architecture. *IBM Systems Journal*, 26(3), 1987.