Abstract

Merging and integrating different conceptual models which have been collaboratively developed by domain experts and analysts with dissimilar perspectives on the same issue has been the subject of tremendous amount of research. In this paper, we focus on the fact that human analysts’ opinions possess a degree of uncertainty which can be exploited while integrating such information. We propose an underlying modeling construct which is the basis for transforming conceptual models into a manipulatable format. Based on this construct, methods for formally negotiating over and merging of conceptual models are proposed. The approach presented in this paper focuses on the formalization of uncertainty and expert reliability through the employment of belief theory. The proposed work has been evaluated for its effectiveness and usability. The evaluators (a group of Computer Science graduate students) believed that the proposed framework has the capability to fulfill its intended tasks. The obtained results from the performance perspective are also promising.

Key words: Collaborative Modeling, Viewpoint Integration, Conceptualization, Inconsistency Management, Belief Theory

1. Introduction

Requirement engineering is an important branch of software engineering which is involved with the identification and formalization of the goals, functions and constraints of a software entity [1]. Many researchers believe that requirement engineering is a process composed of two elements namely requirement elicitation, and requirement modeling [2]. In requirement elicitation, analysts are concerned with the revelation, communication, and validation of the facts related to the software entity, while in the modeling phase, they attempt to represent and organize the identified facts in some agreed format usually known as conceptual models. Conceptual models are therefore representational abstractions of the possible model of the software entity [3]. Conceptual modeling is a challenging aspect of the requirement engineering process.

Various attempts have been made to create a firm foundation for conceptual modeling. Formal specifications such as the Z notation or models based on logical formalisms, category theory, requirement templates, state transition diagrams, conceptual graphs, and object oriented models are among the most widely used [4]. Besides the problem of developing a unified model for software conceptual models is the issue of developing and instantiating a proper conceptual model for a software entity. The principle of employing more information sources for getting a better insight into a given problem has been widely used in court investigations for many years [2]. The intuition behind this practice is that various information sources, more specifically human evidences in this case, have different areas and amounts of knowledge that may help in better analyzing the problem. They may also each use different styles of knowledge expression that can itself be helpful in beating the concern [3].

Requirement analysts have also been interested in using information from multiple sources to create a concrete, consistent and complete compilation of software requirements. Sources of information are mainly known as viewpoints. Nuseibeh defines viewpoints as “loosely coupled, locally managed distributed objects which encapsulate partial knowledge about a system and its domain, specified in a particular, suitable representation scheme, and partial knowledge of the process of development” [5]. Although not all models of viewpoint-based requirement engineering conform to this definition, most of them roughly agree on this basis. For example, Greenspan and Feblowitz [6],
have identified four viewpoints that in their understanding are useful for deriving software systems requirements. These viewpoints are service viewpoint, service workflow viewpoint, organizational model viewpoint, and capabilities and resources viewpoint. We will more specifically visit the various models that employ the viewpoint-based approach later in this paper.

The incorporation of information from multiple sources for creating a complete conceptual model of a software entity seems to be intuitively appealing; however, there are various issues that need to be addressed before viewpoint-based models can be correctly employed. First, humans usually make conception errors due to risk aversion, short-term memory or even framing and perceptual problems [7]. This implies that not all asserted information from the sources are correct or equally reliable. Furthermore, epistemic uncertainty (also known as partial ignorance) is an indispensable element of human judgments that makes them even more susceptible to inaccuracy and imprecision [8]. There is no guarantee that the received information coming from various human sources be consonant or consistent. They may be in many cases arbitrarily (with very few common elements) or disjointedly distributed (no common elements) [8]. Hence models that incorporate human expert judgment into their analysis need to consider the significant role of uncertainty and imprecision.

There are three major theoretical frameworks through which uncertainty can be handled and manipulated, namely imprecise probabilities, possibility theory, and Dempster-Shafer theory of evidence [8]. In this paper, we intend to formalize viewpoints’ role through the notion of belief from within the framework of Subjective logic [9], which is an extension to the Dempster-Shafer theory of evidence that incorporates an explicit notion of uncertainty. This will allows us to develop a unified framework for integrating the beliefs of different viewpoints regarding various aspects of a software entity through directed collaboration.

1.1. Design Concerns and Motivations

There have been various proposals for the formalization of appropriate viewpoint-based frameworks. Each of these approaches have been devised with a specific motive and to address an important concern. The model that we introduce in this paper attempts to provide a basis for collaborative conceptual model development and integration particularly with the existence of partial ignorance and uncertainty [10]. The current work strives to address the following issues:

Capturing Uncertainty: Experts’ judgements play the main role in the process of viewpoint-based modeling; therefore, we explicitly incorporate the notion of uncertainty in our proposed model. The model also supports decision making and inference under uncertainty.
belief propagation schemes that can be used as appropriate tools for pre-consensus belief recommendation.

1.2. The Overall Model

The model that we propose in this paper is based on the degree of experts’ uncertainty towards their expressed specifications. To capture the degree of uncertainty, we employ a three dimensional belief structure based on Subjective logic. This belief structure allows opinions to be expressed with varying degrees of belief, disbelief, and uncertainty. Furthermore, we propose an underlying language construct so that higher level conceptual modeling formalisms (e.g. UML, ERD, OWL, and etc.) can be translated through it onto the belief structures. This underlying language construct provides definitions, which show how higher level language concepts can be converted into a lower level belief structures; therefore, the underlying construct is employed to convert high level language concepts into declarative statements which are annotated with Subjective opinions. The exploitation of these two basic elements (the three dimensional belief model and the underlying modeling construct) would allow us to reason about different specifications that are coming from multiple sources.

One of the important phases of the collaborative conceptual model development process is viewpoint merging. In our proposed framework, this is performed on the basis of belief merging operators such as Dempster’s rule of combination. The only restriction that we apply on the expressed specifications is the use of a common application vocabulary. Similar assumption has been made in the related literature [15,16]. In our approach, the terminology repository is incrementally developed and shared as the viewpoints define and express their models. The common set of vocabulary is required so that specification overlaps are detected. We define three metrics for analyzing the effectiveness of the merging process and the quality of the obtained model. These metrics can assist the analysts in the process of decision making and planning. The proposed framework also develops and guides a formal negotiation process between the viewpoints so that a stable consensus is faster developed. Through the process of viewpoint merging and consensus building, the reliability of the experts and information sources are calculated with regards to various domains of expertise and will be used to discount the expressed information of the viewpoints.

It is important to notice that the merged views are not always syntactically or semantically consistent. Although we agree with Richards et al. [17] that lazy consistency allows the emergence of a richer final specification, it is required that the analysts perform a consistency check on the developed specifications at some point of time. Our framework provides the basis for automatic analysis of the specifications and performs the pruning of the specifications that are not significant from a collective perspective. Figure 1 shows the flow of the processes in the proposed framework.

This paper is organized as follows: In the next section, a brief overview of the related literature is given. Section 3 continues with the introduction of the theoretical developments of the framework, and is followed up by the elaboration of the model development and integration processes. In Section 5, the details of the two proposed recommendation schemes are given. Sections 6 and 7 discuss a short example and the tool support for the proposed framework. The framework is then evaluated in Section 8. The paper is concluded with some discussions and concluding remarks.

2. Background

Our proposed framework originates from the stream of research that is being pursued in the viewpoint-based conceptual model design field. The framework also borrows various operators, schemes, and ideas from belief theory; therefore, we briefly review the relevant literature to these two areas in the following subsections.

2.1. Viewpoint-based Modeling

As the magnitude and complexity of a software system increases, the development of its large conceptual model turns into a process which requires the collaborative participation of various viewpoints [18]. A conceptual modeling process with the involvement of various viewpoints is likely to wind up with many conflicting and inconsistent specifications. In the first glance these discrepancies may be considered undesirable; however, their existence can point to the aspects of the system that deserve more attention and deeper analysis [19]. There have been several attempts to formalize the process of viewpoint-based modeling.

Viewpoint-Oriented Requirement Definition (VORD) is one of the early models of requirement engineering that supports the exploitation of viewpoints. VORD allows the analysts to define software requirements in any notation. Although it does not support any sort of automated viewpoint analysis, it has defined a process in which cross viewpoint analysis for inconsistency identification has been recognized as one of its important steps [20]. As an extension to the VORD model, Preview defines a viewpoint as a concept with six slots, namely viewpoint name, viewpoint focus (boundary and scope), viewpoint concern (e.g. organizational goals, business objectives, etc.), viewpoint information sources, viewpoint requirement definitions, and the viewpoint activity history used to incorporate traceability features into the model [20]. One of the most important features of the Preview model is the notion of concerns. Concerns are high-level strategic goals and aspects of the intended software entity such as safety, and fault tolerance, that need to be observed in the development of the requirement specifications. Concerns crosscut viewpoints and can be addressed by any viewpoint. The Preview model also identifies three meta-viewpoints that the authors believe can cover most types of viewpoints in a typical software en-
In a similar attempt, Nuseibeh formalizes a viewpoint as a metamodel with five features: 1) representation style, 2) domain of interest (area of concern), 3) requirement specification, 4) work plan (requirement engineering strategy), and 5) a work record [5]. From the analysts’ perspective, this model is less flexible as compared with the previous two models since it requires a strict declaration of a representation style, and work plan for each viewpoint.

In the Controlled Requirement Expression (CORE) model the problem domain is divided into disjoint areas of concern [21]. Each viewpoint is concerned with the completion of the specifications of that certain subset of the problem domain. A consistent requirement specification is obtained if all the requirement models are merged into a single model. Merging viewpoints in this model seems to be the simplest form of conceptual model merging among all viewpoint-based models since there are no overlaps between the viewpoints and therefore very few (if any) inconsistencies will occur. It is important to mention that view merging is not always this straight forward, and counts as one of the key problems in conceptual modeling [22].

A typical merging strategy should consist of a process for identifying, evaluating and resolving discrepancies between different viewpoint specifications, and integrating the ultimate decided specifications into a unified representation [2].

The models that are currently practiced for identifying specification overlaps and discrepancies are based on one of the following schemes: shared ontologies (common application vocabulary or thesauri), human expert inspection, and formal methods of similarity analysis [19]. In the methods that use a common application vocabulary, it is assumed that the analysts are provided with a shared repository that is gradually completed and is used for expressing software specifications. If the analysts confine to this shared repository of concepts, the detection of specification overlaps can be performed in a semi-automatic fashion.

DealScribe [23], QARCC [24], and Synoptic [25], are a few of the models that employ expert-centric identification of discrepancies. For example, in Synoptic, experts are required to fill in conflict forms in cases where they find a conflict in the specifications. Similarly in DealScribe, experts are asked to evaluate the degree of conflict between the viewpoint specifications, and based on the provided in-
formation, the contention and the average potential conflict metrics are calculated.

The majority of the models that perform a formal inconsistency analysis require the specifications to be in a unified representation style. In [2], the authors propose a special language to represent viewpoint specifications called VWPI. Consistency between the viewpoints of this model is calculated by a static analyzer. In a different attempt, the authors in [26] have developed a model that converts viewpoint specifications into first-order logic. Consistency of the viewpoints are then analyzed through the manipulation of the first-order logic clauses. In the KAOS framework [27], divergence and inconsistency has been analyzed through goal regression which identifies boundary conditions that maybe the source of specification discrepancies. This model is based on backward chaining and formalizes software system goals and domain information in the form of \( A \rightarrow B \). Backward chaining starts by taking a negation of some asserted fact and continues until it unifies with another given fact in the aggregated specification knowledge base. If no unification is reached it can be inferred that inconsistencies do not exist. Several other interesting formal models for inconsistency identification such as the \( \chi \) bel framework [28], and a model for similarity analysis on top of Telos ([29]) [30], a model for manipulating multiple goal oriented requirement engineering diagrams called AGORA [31] and many other models can be found in the related literature.

In a different approach to the formalization of viewpoint models, the authors in [32] have suggested the use of Category theory [33]. In this approach, models are formalized as categories and connectors as functors. Similarly, Sabetzadeh and Easterbrook [22] propose the use of category theory and the colimit operator for merging viewpoints. They exploit Belnap’s knowledge order [34] to define knowledge degrees which partially allows the identification of conflicts and uncertainty in the merged models. In this model, the value of the conflicting or uncertain knowledge degrees are calculated using the least upper bound operator.

### 2.2. Belief Theory

Many of the experts’ judgments are mixed with uncertainty and imprecision [35]. The exploitation of the implicit notion of uncertainty in experts’ judgments requires models that benefit from viewpoint-based approaches that consider the role of uncertainty in their decisions; therefore, uncertainty and imprecision need also be captured through the elicitation process in these models. Lets suppose that an expert intends to express his dis/agreement with a piece of information. This piece of information can be partially expressed using traditional probability theory. The main deficiency of the application of probability theory here is that it is not capable of representing and dealing with uncertainty. Traditional probability theory respects the principle of additivity, and therefore, the probability mass not assigned to a variable will be assigned to its complement which is not applicable to the judgement of human experts. In human judgments, although a certain degree of agreement is assigned to a variable \((P_i)\), it cannot be inferred that the expert believes in the complement of that variable with the rest of the probability mass not assigned to the variable \((1 - P_i)\). It is possible that the rest of the probability mass be left unassigned and be interpreted as the degree of expert’s ignorance or uncertainty.

Belief calculus is of the theoretical models that are able to numerically quantify the lack of experts’ knowledge in an effective manner [36]. It is a potentially useful tool for the evaluation of the reliability of experimental measurements of the factors that have been evaluated with the involvement of human experts. Dempster-Shafer theory of evidence is one of the most widely employed models of belief representation that provides appropriate means for approximate and collective reasoning under uncertainty [37,7]. It is an extension to the traditional probability theory where probabilities are assigned to sets (or intervals) as opposed to singleton variables [8].

The basis of the evidence theory is based on a set of hypotheses \( \theta \) called the frame of discernment defined as:

\[
\theta = \{H_1, H_2, ..., H_N\}.
\]  

(1)

The frame of discernment consists of \( N \) exclusive hypotheses. Based on the frame of discernment \( \theta \), the power set \( 2^\theta \) composed of all the possible propositions can be created such that:

\[
2^\theta = \{\phi, \{H_1\}, \{H_2\}, ..., \{H_N\}, \{H_1 \cup H_2\}, ..., \theta\}.
\]  

(2)

The base function required for the evidence theory is the basic assignment function (bba). It is similar to probability distribution, but differs in the fact that belief masses are distributed over the elements of the power set \( 2^\theta \) and not only on the singleton elements of \( \theta \); therefore, composite subsets of the power set can also receive a degree of the belief mass. The belief \( m_j \) assigned by an information source \( j \) is defined as a function that maps the power set to the \([0, 1]\) interval and should observe the following conditions:

\[
m_j : 2^\theta \rightarrow [0, 1],
\]  

(3)

\[
m_j(\phi) = 0,
\]  

(4)

\[
\sum_{S \subseteq \theta} m_j(S) = 1.
\]  

(5)

The belief mass \( m_j(S) \) shows how firmly the information source \( j \) believes in the hypothesis presented in \( S \). In cases where \( S \) is a composite subset of \( \theta \), the belief mass assignment has been attributed to \( S \) due to uncertainty and the lack of information over the truthfulness of the subsets of \( S \). We will later discuss that the finer the belief assignments are, the lower uncertainty will be. From the definition of the basic belief assignment function, two important functions can be defined: belief function \( Bel_j \) and the plausibility function \( Pl_j \). The belief function shows the amount of
belief that information source $j$ has assigned to any proposition such as $S$. The plausibility function represents the total amount of potential belief that can be assigned to a proposition like $S$. The belief and plausibility functions are defined as:

$$Bel_j(S) = \sum_{B \subseteq S} m_j(B).$$  \hfill (6)

$$Pl_j(S) = \sum_{S \cap \overline{B} \neq \phi} m_j(B).$$  \hfill (7)

It is straightforward to infer the relationship between the basic belief assignment function and the belief and plausibility functions shown below:

$$m_j(S) = \sum_{B \subseteq S} (-1)^{|S-B|} Bel_j(B).$$  \hfill (8)

$$Pl_j(S) = 1 - Bel_j(\overline{S}).$$  \hfill (9)

where $|S-B|$ is the difference of the cardinality of the two sets and $\overline{S}$ is the typical complement of the set $S$ with regards to $2^X$.

Suppose as an example a case where a witness has been summoned to court to testify about a murder case. The witness expresses his/her opinion (guess) about the suspects in the form of basic belief assignments. It is possible to see from Table 1 that evidential theory has been able to capture the uncertainty present in the belief expression of the witness regarding the guilt of the suspects, by enabling him/her to assign belief masses to two composite sets. Having provided the tool to model the available uncertainty in the testimony of one witness through evidential theory, the major question is how to consolidate various testimonial evidence from different witnesses. Dempster-Shafer theory provides the suitable means for integrating various belief assignments into one belief in a fair and equal way through the application of Dempster’s rule of combination [38].

Dempster’s rule of combination merges multiple belief functions expressed by various independent sources of information. The combination operator functions over two basic belief assignments $m_1$ and $m_2$. The result is a compilation of the collective belief of the two information sources ($m_{1,2}$).

$$m_{1,2}(S) = \frac{\sum_{A \cap B = S} m_1(A)m_2(B)}{1 - K}; S \neq \phi,$$  \hfill (10)

$$K = \sum_{A \cap B = \phi} m_1(A)m_2(B).$$  \hfill (11)

$K$ is a normalization factor that represents the degree of conflict between the expressed belief of the information sources. Dempster’s rule redistributes the conflicting masses over the non-conflicting masses and therefore insists on the mutual agreements and removes conflicts [39]. This approach to belief integration has been criticized due to its counter-intuitive results under highly conflicting belief expressions [40]. Several authors have proposed models to overcome this problem. For instance, Smets has proposed the assignment of the conflicting masses to $\phi$. His interpretation of conflicts is that they occur when the hypothesis space is not exhaustive [41]. In a different approach, Yager proposes the assignment of conflict masses to $\theta$, and interprets it as the degree of overall ignorance [42]. We will employ Yager’s interpretation in the process of evaluating information sources’ reliability.

3. Theoretical Developments

In this section, we discuss the theoretical basis of our approach. The belief foundation for expert opinion expression are formalized and proper methods for converting opinion expressions from linguistic terms into belief structures and vice versa are discussed. The underlying modeling construct is also developed and the possibility of converting conceptual models developed based on UML through the proposed modeling construct form and onto the belief structures is investigated. The process of modeling experts’ reliability from the perspective of various domains of concern is also formalized.

3.1. Belief Formalization

In the elicitation process involving various experts, each piece of gathered information can be represented in the form of a declarative expression. As an example, consider a case in the design of an electronic learning system where one of the analysts ($A_i$) has defined the concept of ‘course’ as a UML class. This can be expressed as $bel_i(f(A_i, class(course)))$; which means that expert $A_i$ believes that a course should be modeled as a class in the conceptual model. Based on this model, since the analyst does not have any doubt (uncertainty) about the expressed specification, it can be inferred that $s/he does not believe in modeling a course as an attribute of a larger class. Lets suppose that expert $A_i$’s belief about the course concept being modeled as a class be $x$, then we have:

$$bel_i(x) = 1, disbel_i(x) = 0.$$  \hfill (12)

$$bel_i(\overline{x}) = 0, disbel_i(\overline{x}) = 1.$$  \hfill (13)

It can be seen from the example that the set of hypotheses only consists of $x$ and its complement $\overline{x}$; therefore, the frame of discernment is binary which means that there are only two hypotheses here that can receive belief masses in

<table>
<thead>
<tr>
<th>Sample Belief Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peter Paul Mary Paul</td>
</tr>
<tr>
<td>$m_j$</td>
</tr>
<tr>
<td>$Bel_j$</td>
</tr>
<tr>
<td>$Pl_j$</td>
</tr>
</tbody>
</table>
the framework of belief calculus. It is logical to employ Subjective logic an extension of the Dempster-Shafer theory of evidence that supports belief representation and reasoning in binary frames of discernment. Subjective logic explicitly defines uncertainty as a separate dimension that is actually implicit in the definition of belief in the Dempster-Shafer theory. This is a major advantage for our purpose since we intend to capture experts’ uncertainty about their expressions.

A belief expression in Subjective logic is defined as a 3-tuple $\omega^A = (b^A_x, d^A_x, u^A_x)$ also known as the opinion of expert $A$ about hypothesis $x$ ($\omega^A_x$). It should be noted that the elements of opinion in Subjective logic can be converted to functions of the Dempster-Shafer theory by allowing $Bel_A(x) = b^A_x$ and $Pl_A(x) = b^A_x + u^A_x$. It can be shown with this definition that belief ($b^A_x$), disbelief ($d^A_x$), and uncertainty ($u^A_x$) elements of an opinion should satisfy:

$$b^A_x + d^A_x + u^A_x = 1. \quad (14)$$

The above condition restricts the possible values that can be expressed as an opinion by an expert only to the points placed in the interior surface of an equal-sided triangle. The three constituent elements determine the position of an opinion within the triangular space. Figure 2 shows the three axis that can be used to identify the position of an opinion point in the triangle. In the opinion triangle, the line connecting absolute belief and absolute disbelief corners (right and left corners) is called the probability axis. This is because the removal of uncertainty from Subjective logic will result in a pure probabilistic interpretation of belief (i.e. $b^A_x + d^A_x = 1$). The opinions which are situated on this axis are named dogmatic opinions since they do not contain any degree of uncertainty. Among dogmatic beliefs, the two opinions located on the extreme ends of the probability axis are called absolute opinions and represent inflexible agreement or disagreement with the hypothesis ($b^A_x = 1, d^A_x = 1$). In the electronic learning example, the opinion expressed by the expert about the course class is regarded as an absolute opinion (complete agreement-dogmatic), and is placed on the right end of the probability axis. Suppose that another analyst ($B$) has expressed its opinion about the same concept ($x$) with the following values $\omega^B_x = (0.5, 0, 0.5)$. This means that the analyst is rather uncertain and ignorant of the correct model for $x$, but prefers to model the concept of course as a class. The location of this opinion ($\omega^B_x$) is shown on Figure 2.

3.1.1. Linguistic Opinions

Domain experts and analysts are generally uncomfortable with expressing their opinions in an exact probabilistic form. They prefer to use common linguistic terms to articulate their opinions in a rough manner [43]. For this reason, it is required to convert linguistic expressions of the experts into a numeric format so that calculations can be performed and then re-convert the numeric values into linguistic terms for expert comprehension.

As it can be seen in Figure 2, the uncertainty axis and the probability axis have been divided into four and five sections, respectively. Each of these divisions represent a linguistic term. For example, the partitions on the uncertainty axis have developed the absolutely uncertain, very uncertain, slightly certain, and absolutely certain terms that can be used by the experts. On the other hand, the divisions on the probability axis has created five linguistic terms, namely firm disagreement, slight disagreement, either way, slight agreement, and firm agreement. If we consider the spaces that are created by the intersection of these two divisions, we will have various sub-spaces within the triangle that roughly represent certain linguistic expressions. For example, the gray area in Figure 2 represents the kind of opinions by the experts where the expert is rather uncertain or ignorant of the correct model for $x$ and is interpreted as ‘expert $C$ is very uncertain about $x$; however, at the same time, prefers to slightly agree with it’.

Once an opinion has been expressed in a linguistic form, it can be converted into its numeric representation by first finding its correct location in the triangle, and then taking the value of the belief elements of the center of gravity of the corresponding sub-space as the representative of the expressed opinion. As an example, suppose that a different expert ($D$) has a very strong feeling that the course concept should be modeled as an attribute of the Department class and strictly disagrees with the fact that it should be modeled as a separate class ($x$); therefore, s/he expresses his disagreement with $x$ as ‘I firmly disagree with $x$ and I am absolutely certain about this’. This expression of opinion about $x$ can be converted to $\omega^D_x = (0.1, 0.8, 0.1)$ by using the center of gravity of the formed trapezoid in the far left side. The very low degree of belief mass that has been assigned to the proposition is due to the fact that the selected divisions for the linguistic terms are rather coarse.
Analysts can decide on the degree of granularity that they deem appropriate for their purpose. It should be noted that very fine linguistic terms can cause ambiguity themselves. The re-conversion of numeric opinion values to linguistic terms is also performed similarly.

3.2. The Underlying Modeling Construct

The integration of conceptual models is the issue of many different application domains such as software conceptualization, database schema development, and ontology creation, to name a few. For this reason it would be enticing to create an integration model that is independent of the actual conceptual modeling formalism that has been used to create the final specifications. To achieve this, a lower level modeling construct is required so that the higher level conceptual models can be mapped through it onto the belief structure and the reasoning process be performed based on the lower belief layer. The results of the operation on the lower level models can then be mapped back to the higher level conceptual models. Figure 3 depicts how viewpoints can use any known conceptual model for expressing their idea without having to worry about the underling formalisms of belief modeling and reasoning.

Construct is a low level modeling notion that we propose for mapping higher level models. It has the capability to be decorated with belief elements that can be used for reasoning and integration. Construct has four segments: Attributes, Opinion, Pre-condition, and Post-condition. Attributes are the set of elements that are needed in the higher level model. Construct has three default attributes that can be extended by analysts. These attributes are Name, Cardinality, and Contributors. The name attribute allows each construct to have a unique name, cardinality defines the number of identical instances that are permissible, and the contributors attribute is used to identify the list of experts (viewpoints) that have affected the definition of this instance of the construct, which is used for traceability purposes.

The belief segment consists of three elements namely belief, disbelief, and uncertainty that are employed to assign subjective opinions to each instance of the construct. Pre-conditions and post-conditions are the set of circumstances that need to hold before and after the creation of an instance of a specific construct. Suppose that a set of analysts each representing a different viewpoint have agreed upon using the class diagrams of the unified modeling language as the high level conceptual modeling language. In order for them to be able to use our proposed framework, the underlying construct model should be customized for the class diagram of the unified modeling language. Here, we show how the class, and aggregation notions from the set of all concepts in the class diagrams can be defined in the construct format. The definition of the rest of the concepts in the unified modeling language, entity-relationship diagrams, OWL, etc. trivially follow the same path and are very similar to what is shown in Figure 4.

As it can be seen in the definition of both constructs, the name of each instance is to be assigned by the analyst, and only one instance of each construct with the same specification is permitted. Opinions are assigned complete belief by default that will be over-ruled by the belief values that are provided by the analysts. If no belief value is assigned to a concept, a full degree of belief is assigned to it. The most important section of the definitions are the pre and post-conditions. In the Class construct, it should be made sure that if a different construct instance has been created with the same name, the belief of the analyst is lower than the sum of the uncertainty and the disbelief for that instance. This pre-condition makes sure that there exists a certain degree of belief that can be assigned to this construct. In the Aggregation construct, beside the same condition that should hold, it should also be made sure that the base constructs that are used as aggregation source and destination be believed to be more of class construct nature than any other construct. This means that the source and destination of an aggregation cannot be of type property, composition, or others. In the definitions above, \( X \) is a variable that can be unified with any defined construct.

Post-conditions are propositions that should hold having expressed an instance of the construct. For example, the post-conditions for the Class (Aggregation) construct are that if the belief value assigned to an instance construct is higher than 0.5, all other constructs instances with a similar signature other than the Class (Aggregation) construct
should be assigned the complement of the belief value assigned to this instance of the construct. This condition does not apply to constructs with a belief lower than 0.5, since a person may disagree with multiple construct instances which does not mean that s/he agrees with the other constructs and therefore the other constructs should not receive the complement of the belief. To clarify this point, consider a case where an analyst has stated that a given concept should not be modeled as a Class. This statement does not mean that s/he agrees with the proposition that the concept should be modeled as an attribute. However on the contrary, if the analyst states that a given concept should be modeled as a class, it can be inferred that s/he does not believe in it being modeled as an attribute. It is important to observe the condition that only construct instances that do not possess a prior directly assigned belief value can be automatically assigned a degree of belief.

Let’s consider the simple example given in Figure 5. In this example, a single analyst has expressed his opinion about a problem, with a certain degree of belief. Initially we have to convert the linguistic values into numeric values. Therefore, using the underlying construct layer definitions the following belief values will be automatically developed:

\[ \omega_{\text{car-class}} = (0.85, 0.05, 0.1), \]
\[ \omega_{\text{tire-class}} = (0.85, 0.05, 0.1), \]
\[ \omega_{\text{car-tire-aggregation}} = (0.65, 0.25, 0.1). \]

Based on these belief values we can see that the preconditions of all the propositions are true; therefore, the addition of all propositions to the model is permissible. Furthermore, the post-conditions should all hold, and hence the following propositions can be added to the set of propositions.

\[ \omega_{\text{car-attribute}} = (0.85, 0.05, 0.15), \]
\[ \omega_{\text{tire-attribute}} = (0.85, 0.05, 0.15), \]
\[ \omega_{\text{car-tire-composition}} = (0.65, 0.05, 0.35). \]

The intuition behind these rules is that if an analyst believes for example that the relationship between the car and its tires is of aggregation type, then s/he does not believe in it being of type composition. Similarly, if an analyst expresses his opinion concerning the suitability of a class for the representation of the concept of car and tire, it can be inferred that s/he does not believe in them being an attribute of a class. The reason that the attribute construct has been added to car and tire and not to car-tire is that only constructs with similar signatures are considered while negation of beliefs are automatically generated. This
can be also seen in the addition of the composition construct to car-tire, and not to car and tire. As it can be seen from this example, pre and post-conditions are analogous to the consistency rules that have been employed in various papers such as [44] and [45]. Another important issue that needs to be mentioned is that the underlying language construct is restricted to handling higher level conceptual models that are convertible into a declarative form (See examples above); however, this is not a restrictive condition, since several researchers such as Möller [46] have shown that domain knowledge, whatever level it belongs to, can be represented declaratively. Therefore, an underlying language construct that provides support for declarative languages should be sufficient for our purpose.

3.3. Viewpoint Reliability Structure

In our perception of viewpoint-based models, concerns are one of its most important aspects. The definition of concern in our model is similar to what has been defined in [20]. Based on this definition, each viewpoint is responsible for considering the information for some specific concerns of the system. Since not all analysts have the same degree of expertise in all of the different concerns that are considered in the elicitation process, a reliability metric needs to be defined. This reliability metric can be used to discount the information which is expressed by that viewpoint. According to the fact that we employ concerns (n concerns), each viewpoint can be assigned a set of different reliability measures (one for each concern).

The set of reliability values attributed to each viewpoint has two faces. First, the analysts involved in the creation of the specifications for each viewpoint are asked to express their own degree of confidence in their understanding and knowledge of that area of concern. Second, a third-party understanding of the reliability of each viewpoint in a specific concern is taken into consideration. The third-party reliability values can be equally set for all viewpoints if no extra information about the viewpoints are available. We will explain how third-party reliability beliefs are calculated later in this paper. Figure 6 shows how the two reliability values can be assigned to each viewpoint.

Viewpoint analysts are initially asked for the perception of their own reliability. They can state their opinion using linguistic opinion terms. Once the opinions are expressed, the amount of ascribed belief is extracted and assigned to that viewpoint for the specific concern. As it can also be seen in Figure 6a, the sum of the reliability values assigned to all concerns for a certain viewpoint should add up to one; therefore, a normalization process is needed here. The same applies to the reliability values assigned from the perspective of the third-party. The issue now is to integrate these two sources of reliability information into a single value.

It is rather intriguing to interpret the reliability values assigned to each viewpoint as the amount of belief mass that has been assigned to each viewpoint. The ascribed mass (assigned either by the viewpoint itself or the third-party) represents the degree of belief in the fact that the viewpoint is going to reveal the correct specification; therefore, the combination of the two reliability values reduces to the problem of combining two belief mass assignments. We propose the use of Yager’s rule of combination which is actually the application of Dempster’s rule of combination without normalization. Yager’s rule can be considered as an epistemologically honest interpretation of the belief masses since it does not change the value of the belief masses through normalization. Instead of normalization, Yager’s rule assigns the conflicting belief mass to the universal set (θ).

According to Yager’s rule, the mass assigned to each hypothesis 1 shows the degree of belief in that hypothesis in cases where the hypothesis is a singleton; however, since we are not totally ignorant of the reliability hypothesis space, we can re-distribute the ignorance mass assigned to θ over the belief of the set of viewpoints. The re-distribution is performed proportionally to the degree of belief mass that has been assigned to each viewpoint by Yager’s rule. Consider Figure 7 where the reliability of each viewpoint is being analyzed with regard to concern1 based on the information from Figure 6. According to Equations 10 and 11, the degree of conflict is equal to 0.612. The initial degree of reliability assigned to each viewpoint based on Yager’s rule of combination would be 0.275, 0.081, and 0.032, respectively. We now need to proportionally re-distribute the conflicting mass (0.612) onto the viewpoints. The ratio of the conflicting masses that each viewpoint is going to receive is 0.612 × 0.71(0.275/0.388), 0.612 × 0.21(0.081/0.388), and 0.612 × 0.08(0.032/0.388), respectively; therefore, the three viewpoints will roughly receive 0.705, 0.211, and 0.082 of belief assigned to them (∑ ≈ 1).

These values are then normalized by dividing them by the largest value (0.705). In order for the reliability values to be usable as a discounting measure for the expressed specifications by each viewpoint, the reliability values need to be converted into the format compatible with subjective logic. Subjective logic requires the explicit definition of all belief, disbelief, and uncertainty values. Through the employment of the method based on Yager’s rule of combination, the belief in each viewpoint can be calculated but the degree of disbelief and uncertainty remain unidentified. For this purpose, the remaining belief mass can be attributed to uncertainty, and since we believe each viewpoint to be honest, no degree of disbelief is ascribed to them. This means that a viewpoint does not try to negatively affect the design process and possible deviations and errors are due to its neglect or ignorance towards the domain. The reliability belief for each of the viewpoints regarding Concern1 would be $R_{V1P1} = (1, 0, 0)$, $R_{V1P2} = (0.3, 0, 0.7)$, and $R_{V1P3} = (0.12, 0, 0.88)$. These reliability beliefs are used to discount

---

1 The hypothesis is that the related viewpoint is going to reveal the most useful information in that concern.
the information expressed by each viewpoint in that certain concern (e.g. Concern1).

4. The Integration Process

The provided information by each viewpoint should be in the form of an agreed conceptual model. These models need to be consolidated into one single model that correctly represents the perspective of all the viewpoints. There are various issues such as creating consensus among viewpoints, conflict resolution, inconsistency management, viewpoint reliability analysis, and decision making that need to be addressed before a successful viewpoint integration process can be performed. In this section, we will describe how various conceptual models originating from several viewpoints can be merged on the basis of negotiation, consensus building, and model pruning.

4.1. Model Merging

Conceptual models need to be mapped through the underlying construct layer before they can be merged. As was explained in the previous sections, the conversion process can be performed automatically based on the defined pre and post-conditions of the related constructs. Upon conversion of conceptual models through the underlying construct onto the belief structures, merging can take place. There have been various proposals for combining belief masses such as Inagaki’s combination rule [47], Zhang’s center of combination [49] that is used for propagating probabilities through logical links between two frames of discernment, Dubois and Prade’s disjunctive rule of combination [49] and the related conjunctive form. Although all these rules of combination seem to be useful for our purpose, we choose the consensus operator [38] to merge various conceptual models [50]. The choice of the consensus operator has been due to three facts. First, the consensus operator has been designed specifically for binary frames of discernment; which is the type of frames that we have also chosen for our model. Second, the operator has been shown to have a stable behavior under various conditions and even while merging conflicting dogmatic beliefs [51]. Finally, it is an appealing choice since it satisfies two important algebraic properties i.e. commutativity (A ⊕ B = B ⊕ A), and associativity (A ⊕ [B ⊕ C] = [A ⊕ B] ⊕ C). These two properties are of great significance since the merging process of conceptual models should neither be affected by the order used for the merge process nor the order of models while being manipulated by the merging operator.

Let \( \omega^A_x = (b^A_x, d^A_x, u^A_x) \) and \( \omega^B_x = (b^B_x, d^B_x, u^B_x) \) be two opinions about a common fact \( x \) stated by two different viewpoints \( A \) and \( B \), and let \( \kappa = u^A_x + u^B_x - u^A_x u^B_x \). When \( u^A_x \rightarrow 0 \), and \( u^B_x \rightarrow 0 \), the relative dogmatism between the two opinions are defined using \( \gamma = u^B_x / u^A_x \). Now let \( \omega^A_{x,B} = (b^A_{x,B}, d^A_{x,B}, u^A_{x,B}) \) be a fair representative of both opinions such that:

\[
\begin{align*}
  b^A_{x,B} &= (b^A_x + b^B_x u^A_x) / \kappa \\
  d^A_{x,B} &= (d^A_x u^B_x + d^B_x u^A_x) / \kappa \\
  u^A_{x,B} &= (u^A_x u^B_x) / \kappa
\end{align*}
\]

\( \omega^A_{x,B} \) is the opinion resulting from the application of the consensus operator on two opinions from different viewpoints about a common hypothesis. The behavior of the consensus operator when \( u^A_x = 1 - d^A_x \) or \( u^B_x = 1 - b^B_x \) (employed to reduce one of the dimensions for visualization purposes) is shown in Figure 8.

The employment of the consensus operator can provide a basis for merging conceptual models that are in essence uncertain. We employ the consensus operator along with a widely used model called discount and combine (aka the tradeoff model) [52]. This model discounts the information provided from each source with a certain value signifying the importance of the information source and then combines the obtained results. More specifically we discount the information provided by each viewpoint related to a certain concern with the reliability value of that viewpoint in that specific concern, which informally means that we
put more weight for the expressions of more reliable viewpoints and less weight for the statements of the less reliable viewpoints. Figure 9 shows this process. The employed discounting operator \([9]\) is defined as:

\[
d^{A \rightarrow C_i}_x = b^{A}_x - c_i b^{C_i}_x
\]  

(15)

\[
d^{A \rightarrow C_i}_x = d^{C_i}_x
\]  

(16)

\[
a^{A \rightarrow C_i}_x = a^{A}_x - c_i A^{C_i}_x
\]  

(17)

where \(b^{A \rightarrow C_i}_x\) is the belief of viewpoint \(A\) about a hypothesis \(x\) related to Concern \(i\) \((x \rightarrow c_i)\) discounted with the viewpoint’s reliability in that specific concern \((R^{C_i}_x)\). The rest of the notations can be interpreted similarly.

4.2. Consensus Building

It is perceivable that merging specifications coming from various viewpoints is going to produce major inconsistencies and conflicts. These inconsistencies appear as conflicting belief masses in the merge process. Although tolerating conflicts and inconsistencies are a recommended practice [17], guiding the merge process towards low inconsistency among the viewpoints and hence more stable final product is a more desirable outcome. It is based on this desire that the participating viewpoints should be gradually guided towards a common understanding, and as a result a shared and agreed model.

In order to move towards a consensus among the viewpoints, each viewpoint should be made aware of the collective understanding of the viewpoints towards an appropriate conceptual model, so that they update their current conceptual model based on the new understanding that they have obtained. In this situation analysts should decide on how they are willing to update their conceptual models. They may want to totally change their initial statements since they have now reached a new conclusion based on the new information, or they may still insist on their previous statements and disagree with the common understanding. These two extreme cases are representative of credulous and dogmatic analysts. In their update strategy these analysts consider zero and full inertia towards change [36]. There are also analysts that may partially lose their belief in what they had stated before, or gain some insight into the problem and therefore, slightly increase their belief in some hypothesis.

One way to help analysts reach a consensus is to allow them to review the merged model together and discuss the issues with each other and create an informal negotiation process [53]. This approach has its own advantages that cannot be neglected; however, the existence of a formal method for assisting viewpoint convergence is highly desirable. Viewpoints convergence can be facilitated by recommending appropriate belief masses to the specifications expressed by each viewpoint, that is, recommending to each viewpoint the assignment of a degree of belief mass for each viewpoint that if adhered to, can accelerate the process of consensus formation.

The degree of change that is expected from a viewpoint for converging towards consensus is inversely proportional to the degree of belief ascribed to its reliability. With this definition, although a viewpoint with a very high degree of reliability in a certain concern is not expected to change its belief and agree with less reliable peers, it is expected to significantly change its belief in cases where it’s opinions are highly conflicting with that of the others while it is highly unreliable in that certain concern. In the process of consensus building, the reliability ascribed by a fair third-party should only be considered as the trustworthy and fair reliability measure, since an analyst may be unaware of its own lack of expertise which will result in low flexibility towards change and hence divergence of viewpoint beliefs. To that end, in the case of conflict, the participating viewpoints should adjust their opinions. The viewpoints with a lower perceived reliability value are expected to change their opinion more than those with a higher reliability value.

To formalize this notion, the third-party reliability value will be directly ascribed to each viewpoint. This will show the degree of trust that can be assigned to the statements of this viewpoint within a certain concern. The complement of the reliability belief will be assigned to the opinions that have been obtained from the model merging process based on the consensus operator. With this approach if the reliability of a user is low, a high degree of belief will be assigned to the fact that adjustments to the viewpoint’s opinions should be made. Let \(R^{V_P}_i = (b^{C_j}_i, b^{C_j}_i, u^{C_j}_i)\) be the third-party reliability belief ascribed to viewpoint \(i\) for concern \(j\), the corresponding degree of belief ascribed to the consensus model would be \(R^{C_j}_i\). Returning to the definitions of Section 3.3, the third-party reliability value assigned to each viewpoint for a specific concern is only a single number in \([0,1]\). This value does not represent the degree of disbelief and uncertainty. We choose to represent this value as the belief in the viewpoint and its complement as the degree of uncertainty about the true reliable viewpoint. Hence, in \(R^{C_j}_i\), \(u^{C_j}_i\) is equal to \(1 - b^{C_j}_i\) and the mass assigned to disbelief is equal to zero. This interpretation of reliability seems to render correct results.

<table>
<thead>
<tr>
<th>Third-party</th>
<th>Self Evaluation</th>
<th>Yager’s Combination</th>
</tr>
</thead>
<tbody>
<tr>
<td>(V_{P1})</td>
<td>0.5</td>
<td>0.277</td>
</tr>
<tr>
<td>(V_{P2})</td>
<td>0.3</td>
<td>0.277</td>
</tr>
<tr>
<td>(V_{P3})</td>
<td>0.2</td>
<td>0.16</td>
</tr>
<tr>
<td>(\theta)</td>
<td>0</td>
<td>0.612</td>
</tr>
</tbody>
</table>

Fig. 7. The Reliability Values of each Concern for the Viewpoints are Combined using Yager’s Rule of Combination.
Once the reliability values of each viewpoint and its complement assigned to the consensus model is calculated, each of the belief bases should be discounted with its corresponding reliability belief. The result would be two belief bases normalized with appropriate reliability values. The combination of these values based on the consensus operator will compile a suitable opinion that can be recommended to the corresponding viewpoint. Let \( \otimes \) and \( \oplus \) be the discount and consensus operators, and \( B_{\text{Red}}^{i} \), \( B^{V_{P_{i}}} \), and \( B^{\text{Cns}} \) be the recommended belief base, viewpoint \( i \)'s belief base, and the consensus belief base, respectively, the recommendation process can be expressed as:

\[
B_{\text{Red}}^{i} = \left[ \otimes \left( R_{C_{i}}^{c_{j}}, B_{V_{P_{i}}}^{j} \right) \right] \oplus \left[ \otimes \left( R_{C_{\text{Cns}}}, B^{\text{Cns}} \right) \right].
\]  \( (18) \)

Let's consider the suitability of the recommendation process for facilitating consensus building by analyzing the two extreme cases where a viewpoint is believed to be fully reliable and not reliable at all. In the situation where a viewpoint is assigned the value of zero as its reliability for concern \( i \), the corresponding reliability belief would be \((0, 0, 1)\). If we discount the expressions of this viewpoint with its reliability belief, all expressions will become totally uncertain and therefore, the combination of such belief base with any other belief base (such as the consensus belief base) will return the second belief base as the result. This shows that uncertain belief bases act as identity elements for the consensus operator. Now consider the second scenario where a viewpoint has been assigned a full reliability belief. In such a case the degree of belief assigned to the consensus belief base is equal to total ignorance, hence the result of the combination process of these two models will be the opinions expressed by the viewpoint which is a desirable result. It can be seen from these two cases that the recommendation belief base developed by this method is both conceptually correct and convergent.

### 4.3 Consensus Effectiveness

The process of belief revision and combination in conceptual model integration is repeated iteratively until a desirable degree of stability is reached. Measuring the degree of stability requires some quantitative measure for evaluating the combined model. Stephanou and Lu [37] have proposed a quantitative metric called the generalized entropy criterion which measures the degree of consensus effectiveness obtained as a result of combining evidences from multiple sources through the application of the Dempster’s rule of combination.

The generalized entropy criterion \( (H) \) is a linear combination of two sub-metrics namely partial ignorance \( (I) \) and total entropy \( (H_{t}) \) and is defined as:

\[
H = H_{t} + \beta I.
\]  \( (19) \)

where \( \beta \) is a problem dependent scaling factor. The total entropy metric is itself a summation over the belief and core entropy metrics. The belief entropy \( (H_{b}) \) is defined as the degree of confusion about the exact fraction of belief that should be allocated to each hypothesis. Core entropy \( (H_{c}) \) is the degree of confusion in distinguishing the correct subsets of the hypothesis space that should receive belief masses. Different from these two metrics, partial ignorance \( (I) \) is the degree of inability to restrict the belief mass to small subsets of the hypothesis space.

It has been shown that the generalized entropy criterion decreases with consensus; therefore, the amount by which it decreases as a result of consensus can be considered as the degree of consensus effectiveness. There are various obstacles to the direct application of the generalized entropy criterion in our application domain. Firstly, it seems that the generalized entropy criterion is most suitable for large frames of discernment, and hence some of its metrics such as partial ignorance do not make sense in a binary frame with singleton elements. Furthermore, since the entropy model is designed for measuring consensus effectiveness based on Dempster’s rule of combination, it has assumed that the conflict mass has been normalized out in combination, which is not a correct assumption in our case. It also does not directly address uncertainty and disbelief dimensions that are present in our belief formalization. For these reasons and inspired by the generalized entropy criteria, we propose three similar metrics for measuring consensus effectiveness, namely ambiguity \( (\zeta) \), indecisiveness \( (\psi) \), and conflict \( (\delta) \).

Each of the metrics are calculated separately for all of the present concerns in the analysis process. The metric values for each concern are superscripted with the corresponding concern indicator. For instance, \( \zeta^{k} \) depicts the ambiguity...
value with regards to concern \( k \). The overall metric values are calculated by a weighted aggregation scheme shown in Equation 25. The metrics are defined as follows.

**Definition** Ambiguity \( (\zeta^k) \) provides the basis to calculate the degree of confusion in the overall viewpoints knowledge about the exact fraction of belief that should be assigned to each hypothesis and is defined as:

\[
\zeta^k = - \left( \frac{1}{e^{1-(b_x + d_x)} - 1} \right). \tag{20}
\]

\[
\zeta^k = \sum_{i=1}^{n_k} \zeta_i^k. \tag{21}
\]

Ambiguity is similar to the belief entropy metric in the generalized entropy criterion. Ambiguity provides the basis to calculate the degree of confusion of the viewpoint with regards to the exact fraction of belief that should be assigned to a given statement. An ambiguous statement shows that the viewpoint is still unsure about the accuracy and correctness of the statement. For instance, a statement such as I have no idea whether part-time students can borrow books is completely ambiguous. This statement can be transformed into \((\text{PartTimeStudent(Person)}, \chi\) where \( \chi = (0,0,1) \); therefore, \( \zeta(\text{PartTimeStudent(Person)}, \chi) = 1 \). Furthermore, a statement such as \((\neg\text{StudentUnionMember(Person)} \lor \text{PartTimeStudent(Person)}, \chi_2\) where \( \chi_2 = (1,0,0) \) is not ambiguous at all, since the viewpoint has been able to express total belief and certainty in the information that it has expressed.

**Definition** Indecisiveness \( (\psi^k) \) is a measure of the ability of the viewpoints to firmly state a given proposition. The further away the degree of belief and disbelief of a given hypothesis are, the stronger and more decisive the hypothesis is. Indecisiveness is defined as:

\[
\psi_i^k = \left( \frac{2}{e^{(b_x - d_x)} + e^{(d_x - b_x)} - \frac{2e}{e^2 + 1}} \right). \tag{22}
\]

\[
\psi^k = \sum_{i=1}^{n_k} \psi_i^k. \tag{23}
\]

Indecisiveness is a measure of the ability of the viewpoint to firmly assert a given statement. The further away the degrees of belief and disbelief for a given statement are, the stronger and more decisive the statement is. A completely decisive statement is one that either possesses \( b_x = 1 \) or \( d_x = 1 \), which means that the viewpoint either completely agrees with this statement or fully disagrees with it. As an example, a statement such as either part-time students can or cannot borrow books from the library is completely indecisive. The difference between indecisiveness and ambiguity is quite subtle in that ambiguity measures the ignorance of the viewpoint towards the possibilities of the stated requirement, while indecisiveness evaluates the inability of the viewpoint to select the best of the possibilities in a tradeoff situation.

The third metric that we define for consensus stability analysis is the conflict metric.

**Definition** Conflict \( (\delta^k) \) defines the degree of inconsistency between the beliefs of the different viewpoints and is modeled by:

\[
\delta^k = \sum_{j=1}^{n_k} \left( b_j, d_j \right)_\text{consensus} / 2, \tag{24}
\]

where \( n_k \) is the number of propositions in viewpoint \( k \). The quantity of conflict between two statements of two viewpoints shows the amount of disagreement between the beliefs of two viewpoints with regards to the statements.

Analogous to the previous two metrics, conflict should also be minimized for reaching a more stable conceptual model as a result of consensus. To create a unified view for each of the three metrics, the value of the metrics in each concern should be aggregated. The aggregation process is a weighted sum of each metric. The applied weight is proportional to the number of propositions in the corresponding concern. In cases where the analysts can assign an importance value to each concern \( (\gamma^k) \), that is also taken into consideration, else all concerns are considered equally important \( (\gamma^k = 1) \). Let \( m \), \( |\text{proposition}^k| \), and \( |\text{proposition}| \) be the number of concerns, number of expressed propositions in concern \( k \), and the overall number of expressed concerns, the overall conflict metric will be:

\[
\delta = \sum_{i=1}^{m} \left( \frac{\text{proposition}^k}{\text{proposition}} \right) \times \gamma^i \times \delta^i. \tag{25}
\]

Ambiguity and Indecisiveness metrics are also similarly aggregated. As mentioned earlier, the process of belief revision and integration is repeated until a stable and agreeable model is reached. In this process, each epoch should be analyzed based on the consensus effectiveness metrics. The desirable state is when the value of all the three metrics has decreased significantly enough as a result of the application.
of the consensus operator. Figure 10 depicts the behavior of the proposed consensus effectiveness metrics.

4.4. Reliability Re-visited

The third-party reliability beliefs assigned to a viewpoint is based on that viewpoint’s activities within each concern. The utility degree of each viewpoint’s belief statements are analyzed using the three metrics introduced in the previous section. The higher the degree of contribution of a viewpoint is to each of these metrics, the more its reliability will increase, and vice versa. Table 2 shows the results of the evaluation of four different viewpoints based on the three metrics for Concern i. We will use this information to explain the process of the third-party reliability belief adjustment.

To identify the degree of change that must be applied to the reliability belief of a given viewpoint, three metrics need to be calculated which are derived from the factors introduced in the previous section, namely average conflict, ambiguity change, and indecisiveness change. Average conflict is measured by integrating the belief propositions of a viewpoint with the consensus belief base developed in the previous cycle. Based on the integration, the average degree of conflict between the expressed opinions of the viewpoint and that of the consensus belief base is measured. Ambiguity change is determined similarly to average conflict, but instead of calculating the average conflict, the degree of change in the ambiguity factor of the consensus belief base is measured. Analogous to the other two metrics, indecisiveness change computes the degree of change in the indecisiveness factor of the consensus belief base. It should be noted that the lower these metrics are, the more effective and successful the viewpoint has been in that certain concern.

The utility degrees are each split into three segments, namely effective (+), neutral (0), and ineffective (−). As it can be seen in Table 2, the average ($T_j$) and standard deviation ($\sigma_{f_j}$) of each metric ($f_j$) is calculated over all of the viewpoints. The viewpoints that have at least two of their metrics within $[T_j - \sigma_{f_j}, T_j + \sigma_{f_j}]$ are considered as neutral. Subsequently, the viewpoints with at least two metrics located in $(-\infty, T_j - \sigma_{f_j})$, or $(T_j + \sigma_{f_j}, +\infty)$ are considered to be effective or ineffective, respectively. According to this classification, the reliability of the viewpoints that are located in the effective zone is increased by a specific amount ($\epsilon$). To keep the balance of the reliability beliefs, the same amount of belief is reduced from the ineffective viewpoints. In cases where there are no ineffective viewpoints, the belief mass assigned to the effective viewpoints is reduced from the neutral viewpoints. Let $n$ be the number of viewpoints in the effective zone and $m$ be the number of beliefs in either the ineffective or neutral zones, and $\alpha$ be a moratorium factor which defines the degree of applied penalty, $\epsilon$ is defined as:

$$\epsilon = \sum_{i=1}^{m} (\frac{B_i^C}{n} \times \alpha).$$

In Table 2, $\alpha$ has been set to 0.2, and only $VP_1$ has been located in the effective zone, and the rest of the viewpoints have been placed in the neutral zone. Since there are no viewpoints in the ineffective zone, the belief mass transferred to $VP_1$ will be reduced from the viewpoints in the neutral zone which is equal to 0.1. The updated reliability belief values for each viewpoint in that certain concern has been shown in the last column of the table.

4.5. Decision Making and Model Pruning

The integration of various conceptual models can result in inconsistencies that may have not been present in the initial conceptual models. It is recommended that these inconsistencies be tolerated until the situation under which a final decision must be made on the ultimate conceptual model. This means that inconsistencies, and conflicts should be resolved at some point in time; however, consistency enforcement is usually one of the last activities that are performed in the model integration process.

In the proposed model, inconsistencies and conflicts arise as a result of the difference in the belief masses assigned to different hypotheses by various viewpoints. For instance, consider the example in Figure 5, where a viewpoint has defined the relationship between the tires of a car and the car itself through aggregation. It is possible that some other viewpoint believes that the tires of a car should be modeled as an attribute within the car class. This means that the belief masses assigned to each of these propositions are in conflict. The conflict may be resolved through the consensus building process, but if it still remains, a final decision must be made on the ultimate conceptual model. This means that inconsistencies, and conflicts should be resolved at some point in time; however, consistency enforcement is usually one of the last activities that are performed in the model integration process.

In the proposed model, inconsistencies and conflicts arise as a result of the difference in the belief masses assigned to different hypotheses by various viewpoints. For instance, consider the example in Figure 5, where a viewpoint has defined the relationship between the tires of a car and the car itself through aggregation. It is possible that some other viewpoint believes that the tires of a car should be modeled as an attribute within the car class. This means that the belief masses assigned to each of these propositions are in conflict. The conflict may be resolved through the consensus building process, but if it still remains, a final decision must be made on the ultimate conceptual model. This means that inconsistencies, and conflicts should be resolved at some point in time; however, consistency enforcement is usually one of the last activities that are performed in the model integration process.

5. Pre-consensus Negotiation

The process explained thus far assumes that the conceptual model of each viewpoint has been readily developed,
and rather ignores the individual viewpoint’s conceptual model development process. This is only acceptable if we assume that each viewpoint is completely aware of the target software system’s needs and requirements, which is not a correct assumption. In real-world scenarios, the viewpoints are incomplete with regards to the domain of discourse and hence are unbiased and willing to receive new information to calibrate their expressed specifications prior to the integration of the specifications with that of the other viewpoints. To address this necessity, we propose two recommendation models that make use of the belief mass assignments of the participating viewpoints prior to model integration and consensus to recommend suitable specifications to the others.

5.1. Belief Recommendation

Human judgements are usually affected by other people’s perception of the same issue. This means that analysts are likely to be influenced by the opinions of the people that they trust and consider reliable. Based on this fact, it seems rational to recommend possible proposition belief mass assignments to analysts derived from the expressions of the others. Returning to the Car example, suppose that an analyst has decided to define a Tire as an attribute, but is rather uncertain about the degree of belief that it should assign to it or even if this is a correct decision. In such cases, an estimate of a proper belief assignment can be inferred from the expressions of the other participating analysts. For instance, the estimate may suggest that the other viewpoints have assigned a high degree of disbelief, which is a sign of disagreement with this proposition. With more information, the analyst can make appropriate decisions that can lead to a faster consensus formation and lower degree of conflict and discrepancy between the viewpoints. The analysts are not compelled to conform to the recommendation.

The belief recommendation process is performed at the time when one of the viewpoints inserts a new proposition into his/her specifications. The specifications of the other viewpoints are checked for a similar proposition (In the example, the rest of the viewpoints are visited for the concept of Tire modeled as an attribute.). If the proposition is found in the other viewpoints, it will be considered in the recommended belief after being discounted with the reliability of that viewpoint. The recommendation process can be formalized as follows:

$$B_{x,Belief.Rec} = \bigoplus_{i=1}^{n} \left( B_{j,C_{j},VP_{i}}^{VP_{i},C_{j}} \right)$$  \hspace{1cm} (28)

where $B_{x,Belief.Rec}$, $B_{x,VP_{i}}$, $C_{j}$ and $n$ denote the recommended opinion for proposition $x$, the opinion of the $i^{th}$ viewpoint towards proposition $x$, the concern that the given proposition belongs to, and the number of viewpoints that have proposition $x$ in their specifications, respectively. $B_{x,Belief.Rec}$ will be eventually recommended to the viewpoint. In the belief recommendation process, third-party

Table 2

<table>
<thead>
<tr>
<th>Initial</th>
<th>3\textsuperscript{rd}-party Rel.</th>
<th>Conflict</th>
<th>Change</th>
<th>Ambiguity</th>
<th>Change</th>
<th>Indecisiveness</th>
<th>Change</th>
<th>Updated 3\textsuperscript{rd}-party Rel.</th>
</tr>
</thead>
<tbody>
<tr>
<td>VP(_1)</td>
<td>0.5</td>
<td>-4°</td>
<td>-2°</td>
<td>-0.7°</td>
<td>0.6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VP(_2)</td>
<td>0.3</td>
<td>-1°</td>
<td>-2°</td>
<td>0.5°</td>
<td>0.26</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VP(_3)</td>
<td>0.1</td>
<td>2°</td>
<td>0.4°</td>
<td>-0.5°</td>
<td>0.07</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VP(_4)</td>
<td>0.1</td>
<td>0°</td>
<td>0°</td>
<td>-0.5°</td>
<td>0.07</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>-</td>
<td>-0.75</td>
<td>-0.66</td>
<td>0.175</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>-</td>
<td>2.5</td>
<td>1.56</td>
<td>0.53</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 10. The Behavior of the proposed Consensus Effectiveness Metrics.
reliability beliefs are employed, since the process should only consider the result of the experts interactions and not their claims in order to make reliable recommendations.

5.2. Proposition Recommendation

As the complexity of a system increases, the design of its conceptual model also becomes an overwhelming task; therefore, an analyst may fail to remember a certain minutiae of the software system. The neglect towards this small issue may cause further ripple effects in the design of the conceptual model of the system. To avoid such a situation, to familiarize the participating viewpoints with each other’s opinions, and broaden the perspective of the viewpoints, the expressed propositions of each viewpoint can be recommended to the others. In this way, as a viewpoint expresses a proposition, the other viewpoints are also notified of the new proposition.

We impose two restrictions on the proposition recommendation process. Firstly, the propositions are initially discounted with the reliability of the expressing viewpoint. The discounting process will put more emphasis on the propositions from more reliable viewpoints, and will assist the filtration of less reliable propositions. Secondly, a proposition is only recommended if after the discounting process, its degree of uncertainty is lower than the sum of its belief and disbelief masses \((d_x + b_x > u_x)\). This condition will prevent uncertain propositions from being recommended to the other viewpoints. Let \(B^\text{Prop-Rec}_j\) and \(VP_i\) be the recommended opinion for the proposition \(x\), and the original viewpoint expressing proposition \(x\), the recommendation process can be defined as:

\[
B^\text{Prop-Rec}_x = \bigoplus_{i=1}^{n} \left( \bigotimes \left( R_{VP_i}^C, B^VP_i \right) \right) .
\]

Similar to belief recommendation, analysts are open to accept or reject the recommendations. Furthermore, through this process, an analyst can become aware of propositions in the other viewpoints that s/he may consider incorrect. The analyst can then assign a total disbelief to that proposition in his/her own specifications which will be propagated (recommended) to the other viewpoints. The belief propagation can act as a medium for opinion exchange and cooperation. This process can be employed as a pre-consensus formal negotiation process that can facilitate conceptual model creation and integration. The experts can gradually soften their belief expressions and reach a compromise.

6. An Example

Consider a simple example where three analysts John, Bob, and Mary representing three different viewpoints collaborate to create a complete model of a transportation system. We partially show how these viewpoints can create a unified conceptual model. For simplicity purposes, the analysts consider the system based a single shared concern and we further assume that the analysts are equally reliable. Figure 11 shows a part of the conceptual models designed by each of the viewpoints (The letters in the gray ovals represent viewpoint opinions in linguistic terms. Refer to Figure 2). As it can be seen the models are annotated with opinion values that help the integration process.

In this example, Bob and John are concerned with the design of the car itself, while Mary is aiming to design the external relationships of the car with the other elements. Before the integration of the models, we have to transform them using the underlying construct. While transforming the conceptual models through the underlying construct, we can infer that there is an inconsistency between the models designed by Bob and John which is the result of the difference in the definitions of the concept of tire. John has modeled the tires of a car as its attributes, while Bob has defined them as a separate stand-alone class.

After the models have been turned into belief representation they can be merged by first discounting each of the propositions of the models with the reliability of the viewpoint and then combining them using the consensus operator. The models are not yet checked for inconsistency, and hence the existence of conflict is permitted. At this stage, the process of proposition recommendation for consensus formation and viewpoint reliability re-calculation can take place. For example, since Mary has disagreed with the definition of tire as an attribute and Bob has defined a tire as a class, John will be informed (proposition recommendation) that the definition of a tire being an attribute is not a suitable choice. Furthermore, based on the assertions of Bob, a suitable amount of belief for assignment to the tire attribute proposition is recommended to John (belief recommendation). The viewpoints can gradually adjust their belief values and use the belief and proposition recommendation procedures to form a formal negotiation process and finally settle on an agreed set of specifications.

As it can be seen in Figure 12, the merged model can contain partial inconsistency that needs to be resolved. Based on the potency metric, we can infer that the model that has defined the tire of a car as a stand-alone class has a higher degree of believability (potency) and therefore, the notion that has modeled tire as a class attribute is removed from the final conceptual model.

It is worth noting that each of the viewpoints had the possibility to insist on its opinion by explicitly adding the conflicting proposition stated by the other viewpoints to its belief base and assigning it a very high degree of disbelief. For example, John could have added a tire class to its model and assigned it an ‘Absolutely Certain/Firm Disagreement’ opinion (Mary has done this for the concept of tire). This would have significantly affected the final merged model. On the contrary, since s/he had already expressed some doubt on his first proposition by assigning an ‘Either Way/Slightly Certain’ opinion to the tire attribute, s/he could have overly softened his opinion (or even removed his initial proposition) so that consensus could have been
achieved more easily and the degree of confidence in the final product would have been higher.  

7. Tool Support

To support the collaborative conceptual model design process, we have designed two separate Eclipse plug-ins,
The plug-ins are developed on top of the Eclipse Modeling Framework (EMF). The viewpoints can install the integration client plug-in into Eclipse and develop their conceptual models. The required information regarding the various types of recommendations, available concerns, common application vocabulary, etc. is sent to the clients from the Integration Server that needs to be set up for the viewpoints to connect to.

The Integration Client plug-in provides the viewpoints with the functionality to assign linguistic beliefs to conceptual model elements (see Figure 13), check for any element without an explicit belief value assigned to it, assert self-assessment of the viewpoint's own reliability, visually observe the change of the integration effectiveness metrics (see Figure 14), view and employ the common terminology that is incrementally developed by all viewpoints, and also receive belief and proposition recommendations. Using the Integration Server plug-in, the moderator of the conceptual modeling process can define the participating viewpoints and their network location, the set of possible concerns and their significance weights, perform versioning on the integrated conceptual models, and visually track the reliability values of the participating viewpoints.

The belief and proposition recommender modules of the plug-ins provide a means for the viewpoints to formally negotiate over their expressed specifications. Besides this type of facility for the viewpoints to communicate, there are also cases where the viewpoints require direct informal communication with the other viewpoints in the form of natural language texts. In such a case, the viewpoints can send textual messages to other viewpoints to clarify their intentions. These messages are attached to model elements so that the track of the issues discussed in the modeling process can be kept. Furthermore, when a viewpoint receives a recommendation from the Integration Server, it can appeal to the proposed recommendation by sending a message to the server. The server will then re-route the rebuttal to those viewpoints that had an influence in the recommendation. With this feature the viewpoints can both formally and informally collaborate through the proposed framework towards building a unified conceptual model.

The tool also provides features for tracking the origins of the elements of the final conceptual model. The viewpoints can observe the history of changes, discussions, and belief adjustments that have been made on the elements of the conceptual model. This feature is useful for understanding the intuition behind the existence of each model element in the integrated conceptual model.
8. Performance Evaluation

The proposed framework was evaluated from two perspectives: effectiveness, and usability. In the set of experiments which were carried out to evaluate the effectiveness of the framework, a group of Computer Science graduate students were asked to design two conceptual models for the Pet store application from a given textual description. The development process of these conceptual models was facilitated by the proposed framework. Furthermore, the participating students were also asked to complete the Computer System Usability Questionnaire (CSUQ) whose analysis was employed for examining the usability aspects of the proposed framework and its supporting tool. In the next two subsections the effectiveness and usability facets of the proposed framework are evaluated and discussed.

8.1. Framework Effectiveness

Sommerville and Sawyer [20] have proposed three main viewpoints, namely domain, interactor, and indirect stakeholder viewpoints that in their belief can cover most of the aspects of a software design process. The domain viewpoint is the advocate for the set of requirements and design needs imposed from the application domain. The interactor viewpoint is the representative of the things that directly interact with the target system and finally, the indirect stakeholder viewpoint is the understanding and requirements of the parties that do not have direct interaction with the target system, but have some sort of interest in the system. They also propose four major concerns that need to be considered which are safety, availability, functionality, and maintainability. Following the same setting, we selected different viewpoints that represented these three types of viewpoints. Each viewpoint was given the same textual definition of the Pet Store application. The viewpoints were required to identify the requirements of the Pet Store application from the textual information. According to their own requirement definitions, each viewpoint then designed two different conceptual models in the form of UML class and state diagrams. They were also asked to initially specify the degree of their own reliability in each of the four concerns.

To support the negotiation, and model integration process, the developed client-side plug-in were provided to the viewpoints. The server-side plug-in was also employed to manage the processing of information that were gathered from each individual viewpoint. The conceptual models developed by each viewpoint were stored on the server-side plug-in in Ecore format. Using these tools, each of the viewpoints started to specify their requirements and then design their conceptual models. In this process, belief and proposition recommendations were constantly provided to the viewpoints. Furthermore, based on the agreement of the viewpoints, the developed conceptual models were merged automatically and stored on the server and the individual models were versioned and stored. This process was repeated until the final result of the collaborative development of the conceptual models was considered acceptable by the viewpoints. Epoches were periodic forty-five minute long working sessions. The development of the class dia-
gram and state chart required ten and eight epoches, respectively. In each epoch, the final merged model was compared with a reference conceptual model that were developed by a group of experienced modelers for the Pet Store application (one for the state chart and one for the class diagram). These reference models were considered as sound models designed based on the given textual description to be compared to (the participating viewpoints were blind to the reference conceptual models).

As has been shown in Figure 14, the integration client plug-in provides the modelers with the feature to observe the behavior of the integration effectiveness metrics. In the two experiments the behavior of these metrics were also observed. Figures 15 and 16 show the average value of these metrics in every epoch for all the viewpoints. The Conflict metric increases until the middle of the experiment and then gradually decreases towards the end. This can be interpreted as a result of the behavior of the modelers. In the collaborative design procedure, the viewpoints start off by adding those elements of the conceptual model that they think is correct from their own stand point. This causes conflicts and discrepancies between their designs. Gradually as the viewpoints observe the proposition and belief recommendations, they consider adjusting their expressions in order to reach a final conclusion. Throughout this process, the degree of conflict will gradually decrease. Similarly, in the other two metrics the Ambiguity and Indecisiveness of the opinions of the viewpoints decreased as a result of their collaboration.

In each epoch, the individual conceptual models were merged and stored. The comparison of the merged model of each epoch with the reference conceptual model yielded the competency of the proposed model. For the UML class diagram, two main metrics were observed in each epoch: Precision and Recall. To calculate precision and recall, four base metrics were measured, namely True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). TP represents the number of design entities that existed in the reference model and also did exist in the merged model, FP denotes the number of model elements that did not exist in the reference model, but did exist in the merged model. On the other hand, TN shows the number of model elements that did not exist in either the merged model or the reference model, and FN depicts the number of elements that did exist in the reference model but were omitted (or did not appear) in the merged model. Using these four metrics, Precision and Recall can be calculated.

Precision (quality of the merged model) is the ratio of correctly labeled elements in the merged model over all of the elements in the merged model, and recall (coverage of the merged model) is the average number of correctly classified elements over all elements. These metrics are determined as follows:

\[
\text{Precision} = \frac{TP}{TP + FP} \quad \text{Recall} = \frac{TP}{TP + FN} \frac{TN}{TN + FN}.
\]

The weighted harmonic mean of precision and recall (F-score) was also measured. In addition to precision and recall that were calculated for the UML class diagram conceptual model, an additional metric was required for the
state diagrams. This metric is needed so that not only the correct addition or removal of an element from the final conceptual model is calculated, but also the correct transition from one state to the other is also considered. This is because in some cases the correct state exists in the final conceptual model, but the transitions from/to that element are not proper. The metric used for this purpose is named Accuracy and is measured by calculating the total number of correct transitions over all of the correct (CT), incorrect (IT) and missing transitions (MT):

\[ \text{Accuracy} = \frac{\text{CT}}{\text{IT} + \text{CT} + \text{MT}}. \]

The experiments showed that the degree of conflict between the viewpoints has a direct influence on the precision, recall and accuracy of the final conceptual models. As it can be seen in Figure 17, the precision and recall in the first experiment decreases as the degree of conflict between the viewpoints increases; however, as the degree of conflict decreases, these two measures increase. Analogously in the second experiment, accuracy decreases as a consequence of conflict between the viewpoints, which gradually rises as conflict is resolved. Furthermore, Figure 17 shows that in the final epoch of both experiments, the viewpoints were able to reach a consensus through the employment of the proposed framework and develop an acceptable final conceptual model (viewpoint convergence).

The other aspect of the system that requires evaluation is the belief and proposition recommender modules. In order to evaluate these two modules, the number of correct recommendations over the total number of recommendations made by each recommender in each epoch was considered as a benchmark for the performance of the modules; therefore, the definition of correctness is important. In the proposition recommender module, if the proposed element existed in the reference conceptual model, the recommendation was considered as correct. In the belief recommendation module if the degree of belief recommended to an item conveyed the correct decision (it being either disbelief or belief), it was considered as correct. Furthermore, if the recommended belief consisted of more belief/disbelief values as opposed to uncertainty it was also considered to be strong. Figures 18 and 19 show the performance of the belief and proposition recommender modules, respectively. The results show that both recommenders have a positive performance and effect on the overall process. An interesting observation can be made in the behavior of the proposition recommender module in the second experiment around the fifth epoch where the percentage of correct recommendations drop in the fifth epoch. To explain this, if we overlay Figures 19 and 16, it can be seen that as the degree of conflict rises, the accuracy of the recommender module decreases. This effect is not significant in the first experiment since the development and integration of class diagrams does not incorporate the notion of sequence or time; therefore, the complexity involved in detecting the correct proposition for recommendation is reduced. In the second experiment where state diagrams are developed, due to the involvement of sequence and time, a mere recommendation of the existence of a state is not enough and the recommendation of the correct transitions is also needed, which is a difficult task to perform while the expressions of the viewpoints are in high conflict.

In both of the experiments the behavior of the third party reliability evaluation method was also observed. To evaluate the final reliability value, two different settings for each viewpoint was tested. In the first setting, the third party reliability value was initially set equal to the value given by the viewpoints (Initialized Reliability). In the second setting, the third party reliability values were initially set to one (Non-Initialized Reliability). The changes made to the third party reliability values were recorded and showed that the reliability values in both settings converged to a similar value with a maximum deviation of 0.06. Figure 20 shows the adaptation process of the reliability values of each viewpoint in the first experiment for the safety concern. Other than the close convergence of the reliability values under both settings, it can be inferred from the overall performance of the proposed framework that the reliability adaptation module behaves suitably such that a unique final conceptual model can be developed.

8.2. Framework Usability

In order to evaluate the usability/applicability of the proposed framework under real world scenarios and its supportive Eclipse plugins, the participating students were asked to complete the computer system usability questionnaire. CSUQ [54] is an instrument for measuring user satisfaction with computer system usability in the context of scenario based usability studies. This questionnaire has been developed and evaluated by the IBM Corporation. CSUQ is made up of four sub-scales, each consisting of items ranked on a 7-point scale: the overall satisfaction score (OVERALL: all 19 Questions), the system usefulness score (SYSUSE: Questions 1-8), the information quality score (INFOQUAL: Questions 9-15), and the interface quality score (INTERQUAL: Questions 16-18). This questionnaire has been chosen because of its acceptable reliability: a coefficient alpha (\(\alpha\)) exceeding 0.89 for all of its sub-scales has been proved. Seven-point rating scales (1=totally disagree, 7=totally agree) were used in the questionnaire because they allow three levels of either positive or negative ratings.

The initial results obtained from the evaluators showed that the proposed framework received a score over the average of the scale score (3.5) in all 19 questions (See Figure 21). Only in two of the questions (9 and 11), the system received a value of less than 4. However, these questions are not directly related to the performance of the proposed framework and address the lack of online help and error messaging in the integration client plug-in tool. Further analysis shows that the values assigned to each question in
the range of [5, 7] dominate the other ranges and for some questions the percentage is similar to the [3, 4] range, which is also acceptable. From the perspective of the sub-scales, the framework received a significant score. The score of all sub-scales were higher than 4.6. The proposed framework received 4.804, 4.633, 4.745, and 5.277 in the OVERALL, SYSUSE, INFOQUAL, and INTERQUAL sub-scales, respectively.

Other than the 19 questions, the evaluators were asked to point to three of the positive and negative aspects of the system. One of the aspects that was highly liked by the analysts was the possibility of assigning belief to the model...
elements; however, there were suggestions that a group selection feature be added to the integration client plug-in so that several elements can be selected at once and a similar belief be assigned to them. The two proposed recommender modules were also appreciated by the evaluators. Here, the only drawback reported by the evaluators was that since the framework is reliant on a common terminology some times different wordings are used for the same concepts, which causes problems for the recommender modules. To overcome this deficiency a thesauri, WordNet, or even a unique application vocabulary can be incorporated into the system. We will address this issue in our future work.

Besides the positive and negative aspects of the system, two of the evaluators had specific recommendations for the improvement of the current work. First, based on the understanding that conceptual modeling is a process that is widely used in different domains, the opportunity for crafting a syntax independent model was found to be precious. A low-level construct that allowed high-level modeling language concepts to be mapped and employed by the reasoning elements of the individual conceptual models that did not appear in the merged model but had the chance of being a part of the final model in the later epochs will be lost in this approach.

9. Discussion

In Section 1.1, we elaborated on the issues that are to our high interest and need to be addressed in the proposed model. Here, we re-visit those points and discuss the contributions of our proposed model. First, based on the understanding that conceptual modeling is a process that is widely used in different domains, the opportunity for crafting a syntax independent model was found to be precious. A low-level construct that allowed high-level modeling language concepts to be mapped and employed by the reasoning elements is used to specify the elements that are disputed. Based on these values and with the help of the least upper bound operator, the result of the integration of several annotated elements can be computed. For instance, the result of the integration of three elements with \( \top \), \( \bot \), and \( \% \) as their annotations values will be an element classified as disputed \( \downarrow \). This case is similar to a case where the analysts employ the model proposed model do not provide any explicit opinion about the propositions that they express. In this case, extreme opinions are assigned to the statements (total belief, or total disbelief). The application of the consensus operator on these elements has a similar result as what has been proposed in [22] and hence reduces to the least upper bound operator under boundary conditions. Even though in such a case the two models show similar behavior for integration, our model provides the analysts with proper tools to determine the degree of integration (consensus) effectiveness. These measures can be employed as indicators to show if further negotiation is needed between the viewpoints.

Easterbrook [55] describes three strategies for conflict resolution namely, cooperative, competitive, and the third party method. In our proposed framework, the viewpoints can benefit from all three of these strategies. The belief and proposition recommender schemes develop a cooperative platform for gradually resolving conflicts and removing discrepancies. On the other hand, each viewpoint can insist on its own design by ignoring the recommendations or explicitly expressing its disagreement with the other viewpoints design which is actually a competitive strategy. Ultimately, if the employment of these two models of conflict resolution does not yield an acceptable result, the potency metric is used to prune inconsistent and conflicting propositions. The potency metric can be considered as a third party resolution strategy which is indirectly based on belief structures and viewpoint reliability.

The other important aspect of the proposed model is the consideration of viewpoint reliability in the model integration process. Each viewpoint is assigned different reliabil-
ity belief values for each of the concerns that it is involved in. The reliability belief is calculated based on the information from the recent activity of the viewpoint and the viewpoint’s belief about its own reliability. The reliability metric plays a major role in the consensus building process, since dogmatic (highly conflict causing) viewpoints can be identified, and in cases where they do not comply with the overall belief, will be automatically ignored in the integration process.

In addition, one of the significant issues that needs to be considered is the extra burden that the proposed model imposes on the viewpoints by requiring them to annotate their conceptual model elements. It should be noted here that the viewpoints are not obliged to fully annotate their conceptual model; therefore, if the propositions are not annotated, total belief will be automatically assigned to the given proposition. In such a case the process is still meaningful. Even more, the viewpoints are even free to only annotate a subset of their conceptual model that they think is required (or that may need more detailed negotiation.) In this way only very little extra work is needed from the participating viewpoints.

Finally, it should be mentioned that the merge process in this paper does not use any form of complex model structure matching/mapping for finding the similarities between two conceptual models. The merge process only considers two model elements identical, if it is able to unify their descriptions. More complex model merging/matching algorithms can be found in the related literature [56,57], which would be employed in our work at a later stage. We have not focused on this aspect of the work yet, since our intention in this paper is to provide suitable methods for handling and representing uncertainty, algorithms for belief and proposition recommendation and also evaluating the reliability of the participating viewpoints.

10. Concluding Remarks

In this paper, we have proposed a model for the collaborative development and integration of para-consistent conceptual models which are tainted with partial uncertainty. The proposed model is based on concepts from belief theory and mainly derives its operators from Subjective logic [9]. It provides features and various tools for reasoning and negotiating over the conceptual models developed by the viewpoints in a collaborative setting and hence formally building and measuring consensus among the viewpoints. The framework incorporates methods for measuring the reliability of the viewpoints based on their conceptual model design behavior and uses the calculated reliability values throughout its decision making procedure, by depending mainly on the reliable viewpoints. The proposed framework has been evaluated from two perspectives: effectiveness and usability. The evaluations show that the various aspects of the framework are suitable for performing the intended tasks. Furthermore, the experts participating in the experiments felt comfortable using and evaluating the system and believed the framework to be well suited for its intentions and expressed their interest in using it in the future.

Our immediate future work is to incorporate a module into the proposed framework so that different viewpoints be free to use dissimilar vocabulary. The module should then enable the viewpoints to identify the possible matches between the employed vocabulary and employ these corre-
spondences in the integration process. Moreover, we are interested in performing controlled experiments to evaluate the suggestions of the evaluators regarding the replacement of the individual viewpoints specifications with the merged model after each epoch.

References


