Faculty of Computer Science
University of New Brunswick

Thesis Proposal for PhD Degree Program

The Integration of Para-consistent Conceptual Models Influenced by Uncertainty: A Belief-theoretic Approach

By
Ebrahim Bagheri
(3223502)

Supervisor: Dr. Ali A. Ghorbani

February 2007

Abstract

Merging and integrating different requirement specification models which have been developed by domain experts and analysts with dissimilar perspectives on the same issue has been the subject of tremendous amount of research. In this research proposal, we intend to focus on the fact that human analysts' opinions possess a degree of uncertainty which can be exploited while integrating conceptual models. We propose an underlying modeling construct which is the basis for transforming conceptual models into a manipulatable format. Based on this construct, we propose to develop methods for negotiating over and merging of conceptual models on top of an extension to the Dempster-Shafer theory of evidence called Subjective logic. The approach shall mainly focus on the formalization of uncertainty and expert reliability through the employment of belief theory. We are also interested in creating a model for pre-consensus negotiation among the involved viewpoints in the conceptual modeling process.

Contents

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 [Zave and Jackson, 1997]. Many researchers believe that requirement engineering is a process composed of two elements of requirement elicitation, and requirement modeling [Leite and Freeman, 1991]. In requirement elicitation, analysts are concerned with the revelation, communication, and validation of the facts related to the software entity. However, 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 [Andrade et al., 2004]. Conceptual modeling may be one of the most challenging aspects 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 models. 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 concern has been widely used in court investigations for many years [Leite and Freeman, 1991]. The intuition behind this practice is that various information sources, more specifically human evidences in this case, have different areas and amount of knowledge that may help in better analyzing the problem [Bagheri and Ghorbani, 2007]. They may also each use different styles of knowledge expression that can itself be helpful in beating the concern [Andrade et al., 2004].

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’ [Nuseibeh et al., 1994]. 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 [Greenspan and Feblowitz, 1993], 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 various models of viewpoint-based requirement engineering later.

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. Firstly, humans usually make conception errors due to risk aversion, short-term memory or even framing and perceptual problems [Pope and Josang, 2005]. 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 [Sentz and Ferson, 2002]. 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) [Sentz and Ferson, 2002]. 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 [Sentz and Ferson, 2002]. Here, we propose to formalize viewpoint expressions through the notion of belief from within the framework of Subjective logic [Josang, 2001] 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 requirement engineering viewpoints regarding various aspects of a software entity.

2. LITERATURE REVIEW

In this section, we review two important fields of work that have a significant affect on our proposed work. Firstly, we visit the works that have been previously performed on viewpoint-based requirement engineering and then explain the basic requirements for employing the features of belief theory and it's operators.

2.1. Viewpoint-based Requirement Engineering

As the magnitude and complexity of a software system increases, the development of its large conceptual model turns into a collaborative effort requiring the participation of various viewpoints. A requirement engineering 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 [Spanoudakis and Zisman, 2001]. There have been several attempts to formalize the process of viewpoint-based requirement engineering.

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 an important step [Sommerville and Sawyer, 1997]. 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 [Sommerville and Sawyer, 1997]. 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, fault tolerance, and etc. that need to be observed in the development of the requirement specifications [Moreira et al., 2005]. 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 engineering process which are namely interactor viewpoint, indirect interactor viewpoint, and domain feature viewpoint. In a similar attempt, Nusiebeh 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 [Nuseibeh et al., 1994]. 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 [Darke and Shanks, 1996]. 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 this issue counts as one of the key problems in conceptual modeling [Sabetzadeh and Easterbrook, 2006]. A typical merging strategy should consist of a process for identifying and evaluating discrepancies between different viewpoint specifications, and integrating the ultimate decided specification into a unique representation [Leite and Freeman, 1991].

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, or formal methods of similarity analysis [Spanoudakis and Zisman, 2001]. In the methods that use a common application vocabulary, it is assumed that the analysts are provided with a shared repository which 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 would be a rather easy task that can even be performed in an automatic or semi-automatic way.

DealScribe [Robinson and Pawlowski, 1999],QARCC [Boehm and In, 1996], and Synoptic [Easterbrook, 1991], 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 information 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 [Leite and Freeman, 1991], 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 [Finkelstein et al., 1994] 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 [Dardenne et al., 1993], 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 χbel framework [Easterbrook and Chechik, 2001], and a model for similarity analysis on top of Telos ([Mylopoulos et al., 1990]) [Spanoudakis and Constantopoulos, 1995] and many other models can be found in the related literature.

In a different approach to the formalization of viewpoint models, the authors in [Fiadeiro and Maibaum, 1995] have suggested the use of Category theory [Barr and Wells, 1990]. In this approach, models are formalized as categories and connectors as functors. Similarly, Sabetzadeh and Easterbrook [Sabetzadeh and Easterbrook, 2006], propose the use of category theory and the colimit operator for merging viewpoints. They exploit Belnap's knowledge order [Belnap, 1977] 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' judgements are mixed with uncertainty and imprecision [Sandri et al., 1995]. The explicit notion of uncertainty in experts' judgments requires proper requirement analysis models that benefit from viewpoint-based models that consider the role of uncertainty in their decisions; therefore, uncertainty and imprecision need also be captured through the process of requirement elicitation in these models. Lets suppose that an expert intends to express his dis/agreement with a piece of information. This type of information can be expressed using traditional probability theory; however, this theory 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 which are able to numerically quantify the lack of expert's knowledge in an effective manner [Kulasekere et al., 2004]. 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 models of belief representation that provides appropriate means for approximate and collective reasoning under uncertainty [Stephanou and Lu, 1988],[Pope and Josang, 2005]. It is an extension to the traditional probability theory where probabilities are assigned to sets (or intervals) as opposed to singleton variables [Sentz and Ferson, 2002].

The basis of the evidence theory is based on a set of hypotheses θ called the frame of discernment defined as:

$$\theta = \{H_1, H_2, ..., H_N\}. \tag{1}$$

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

$$2^\theta = \{\phi, \{H_1\}, \{H_2\}, ..., \{H_N\}, \{H_1 \cup H_2\}, ..., \theta\}. \tag{2}$$

The base function required for the evidence theory is the basic belief assignment (bba). It is similar to the probability distribution, but differs in the fact that belief masses are distributed over the elements of the power set 2^θ and not only on the singleton elements of θ; 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:

$$m_j : 2^\theta \rightarrow [0, 1], \tag{3}$$

$$m_j(\phi) = 0, \tag{4}$$

$$\sum_{S \subseteq \theta} m_j(S) = 1. \tag{5}$$

Table 1. Sample Belief Assignment

	Peter	Paul	Mary	Paul or Mary	Paul or Peter
m_j	0.2	0.05	0.2	0.35	0.2
Bel_j	0.2	0.05	0.2	0.6	0.45
Pl_j	0.4	0.6	0.55	0.8	0.8

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 θ, 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 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:

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

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

It is straight forward to infer the relationship between the basic belief assignment function and the belief and plausibility functions.

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

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

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^θ.

Suppose as an example a case where a witness has been summoned to court to testify about a murder case. The witness expresses his 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 to assign belief masses to two composite sets. Having clearly modeled the available uncertainty in the testimony of one witness through evidential theory, the major concern now is to consolidate various testimonials 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 [Josang, 2002].

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 ($m_{1,2}$).

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

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

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 [Liu, 2006]. This approach to belief integration has been criticized due to its counter-intuitive results under highly conflicting belief expressions [Zadeh, 1984]. Several authors have proposed models to overcome this problem. For instance, Smets has proposed the assignment of the conflicting masses to ϕ. His interpretation of conflicts is that they occur when the hypothesis space is not exhaustive [Smets, 2000]. In a different approach, Yager proposes the assignment of conflict masses to θ, and interprets it as the degree of overall ignorance [Yager, 1987]. We will employ Yager's interpretation in the process of evaluating information sources' reliability.

3. MOTIVATIONS AND CONTRIBUTIONS

There have been various proposals for the formalization of appropriate viewpoint-based requirement engineering frameworks. Each of these approaches have been devised with a specific motive and to address an important concern. Different from the other models, we intend to introduce a model that attempts to provide a basis for conceptual model integration particularly with the existence of partial ignorance and uncertainty. The work strives to address the following issues:

Addressing Uncertainty: Experts' judgements play the main role in the process of viewpoint-based requirement engineering; therefore, we intend to explicitly incorporate the notion of uncertainty in our proposed model. The model will also need to support decision making and inference under uncertainty.

Creating Underlying Modeling Constructs: Merging conceptual models is closely related to other domains such as schema merging [Pottinger and Bernstein, 2003], and ontology merging [Noy and Musen, 2000]. It is based on the relevancy of these problem domains that it is desirable that a single model be applicable to multiple domains. This is achievable if the proposed model is independent of the conceptual modeling language. We propose an underlying construct that allows the mapping of higher level conceptual modeling languages onto its elements, and therefore supports model integration in various modeling environments.

Consensus Building and Negotiation: The models received from different perspectives will contain varying degrees of discrepancy. To remove these discrepancies, information sources need to discuss the reasons of these inconsistencies and settle on a set of agreed requirement specifications. We propose the development of a formal overlay consensus building process on top of the merging framework to enhance the consensus building process.

Employing Linguistic Terms: Domain experts are usually uncomfortable with expressing precise mathematical values. In our proposed method we allow the experts to state their opinions using linguistic terms [Zadeh, 1996]; while all reasoning is performed by converting these opinions into mathematical representations.

Evaluating Merging Effectiveness: Assessing the efficacy of the model created through the merging process is an important issue. Merging effectiveness can be measured through appropriate metrics that calculate the degree of model stability. We intend to develop various metrics for this purpose mainly based on the Generalized Entropy Criterion [Stephanou and Lu, 1988]. However, 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 model.

Information Source Reliability Assignment: Many authors have pointed out the fact that not all information sources are equally reliable [Rogova and Nimier, 2004]. This is also true in requirement engineering. The notion of concerns that has been employed by different researchers shows that some analysts may be more dependable in their opinions in a specific concern as compared with the others; therefore, it is logical to consider the reliability of various experts while merging viewpoint information. We intend to discount the expressed opinions of the experts with their degree of reliability before they are integrated (See Figure 1).

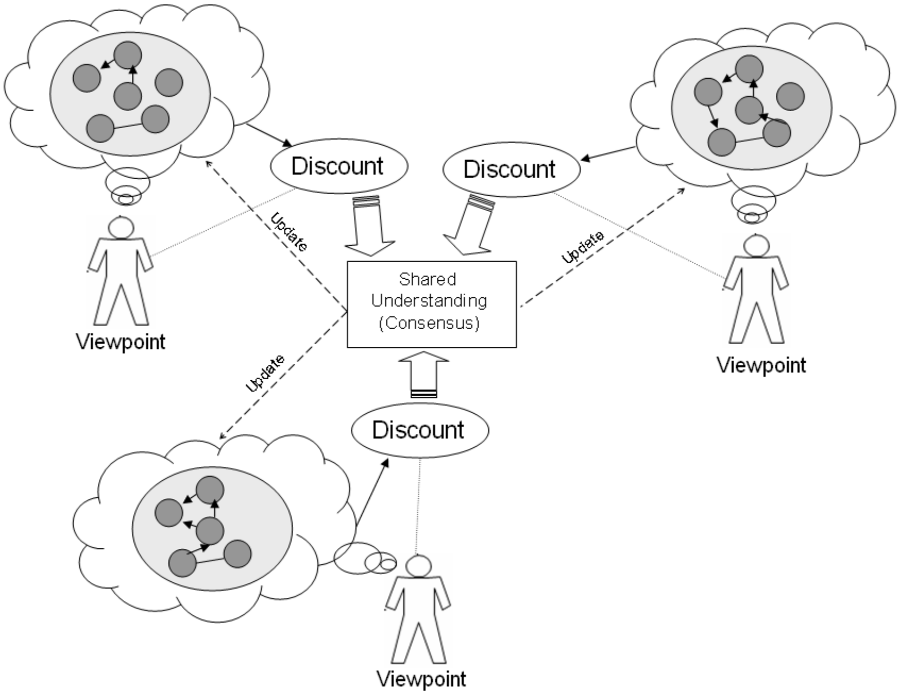


Figure 1. The Process of Viewpoint Integration.

Pre-consensus Belief Recommendation: The other aspect of this work that seems to be quite interesting is to devise a pre-consensus negotiation model between the viewpoints. In contrast to the negotiation model which can be based on the merged model, it would be helpful to create an argumentation and recommendation framework which propagates the opinions of each viewpoint to the others and allows all the viewpoints to consider facts from the others prior to integration and merge. The tools on calculating epistemic necessity and possibility from parent distribution in uncertain fuzzy knowledge bases developed by Magrez and Smets [Magrez and Smets, 1989] seem to be a good candidate for the basis of this work. The authors propose methods for estimating the possibility that a certain proposition is true given a partially uncertain proposition.

Addressing Sequence and Time: We are also interested in working on analyzing the suitability of the underlying construct model for its application to the conceptual models that have a notion of time or sequence such as UML collaboration diagrams, sequence diagrams, or statechart diagrams [Nejati et al., 2007]. It is perceivable that with a Markovian interpretation of those diagrams, the construct proposed in the coming sections is able to fully support such models, but we still need to make sure that they do provide intuitively correct results after merging and integration.

3.1. The Overall Model

The integration model that we propose is founded on the degree of experts' uncertainty towards their expressed requirement 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 into it. The exploitation of these two basic elements (the three dimensional belief model and the underlying modeling construct) would allow us to reason about different requirement specifications that are coming from multiple sources.

The viewpoint merging process 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 requirement engineering specifications is the use of a common application vocabulary. This is required so that specification overlaps are detected. We intend to define various 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 merging process 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. We agree with [Richards and Menzies, 1997] that lazy consistency allows the emergence of a richer final requirement specification product; however, it is required that the analysts perform a consistency check on the developed specification at some point of time. Our framework will need to provide the basis for automatic syntactical correctness analysis of the specifications and allow the pruning of the specifications that are not significant from a collective perspective. The overall flow of the proposed model can be seen in Figure 2.

4. THEORETICAL DEVELOPMENTS

In this section, we will discuss the theoretical basis of our proposed approach. The belief foundation for expert opinion expression will be formalized and proper methods for converting opinion expressions from linguistic terms into belief structures and vice versa will be discussed. The underlying modeling construct will also be developed and the possibility of converting conceptual models developed based on UML into the proposed modeling construct form will be investigated. The process of modeling experts' reliability from the perspective of various domains of concern will also be formalized. However, the model for assigning third-party reliability belief masses to each viewpoint is yet to be developed.

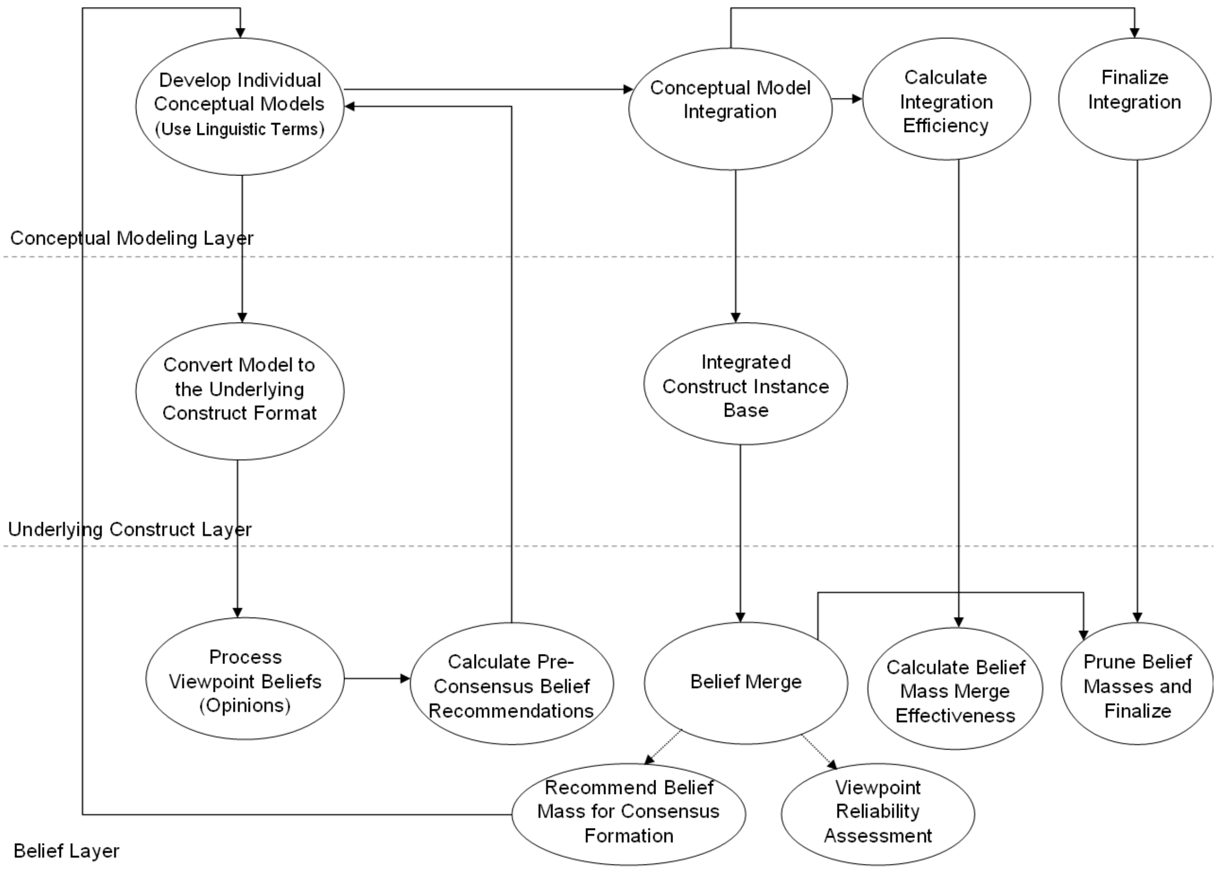

Develop Individual Conceptual Models (Use Linguistic Terms)
Conceptual Model Integration
Calculate Integration Efficiency
Finalize Integration
Conceptual Modeling Layer
Convert Model to the Underlying Construct Format
Integrated Construct Instance Base
Underlying Construct Layer
Process Viewpoint Beliefs (Opinions)
Calculate Pre-Consensus Belief Recommendations
Belief Merge
Calculate Belief Mass Merge Effectiveness
Prune Belief Masses and Finalize
Recommend Belief Mass for Consensus Formation
Viewpoint Reliability Assessment
Belief Layer

Figure 2. The Overall Flow of the Proposed Model

4.1. Belief Formalization

In the process of requirement elicitation from 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 '$belief(A_i, class(course))$'; 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 requirement specification, he is completely certain that 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:

$$belief(x) = 1, disbelief(x) = 0. \tag{12}$$

$$belief(\overline{x}) = 0, disbelief(\overline{x}) = 1. \tag{13}$$

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

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

$$b_x^A + d_x^A + u_x^A = 1. \tag{14}$$

The above equation 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 3 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. $b_x^A + d_x^A = 1.$). 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 a hypothesis ($b_x^A = 1$, $d_{x'}^A = 1$). 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 his opinion about the same concept (x) with the following values $\omega_x^B = (0.5, 0, 0.5)$. 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. This opinion has been shown in Figure 3.

4.1.1. 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. For this reason, it is required to convert linguistic expressions of the experts into a mathematical format so that calculations can be performed and then re-convert the mathematical values into linguistic terms for expert comprehension.

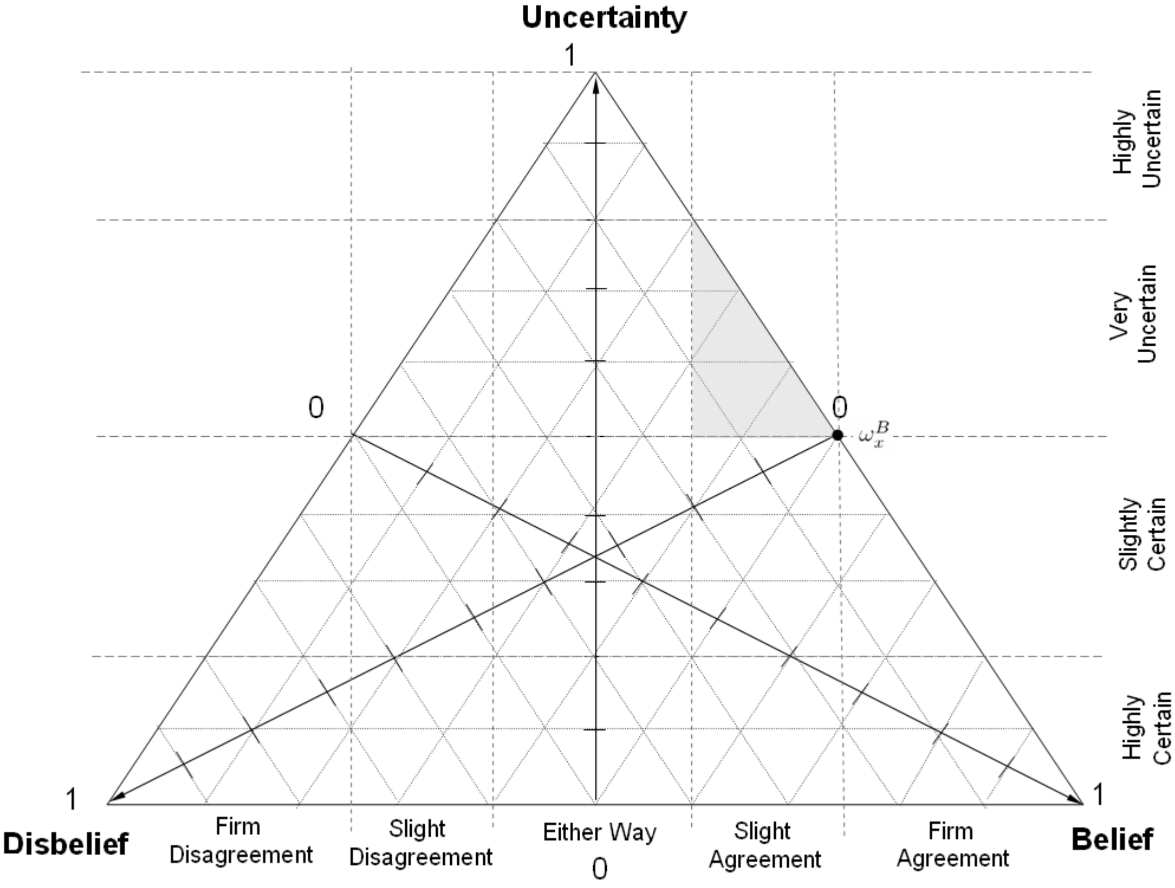


Figure 3. The Opinion Space Can be Mapped into the Interior of an Equal-sided Triangle.

As it can be seen in Figure 3, 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 highly uncertain, very uncertain, slightly certain, and highly 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 3 represents the kind of opinions by the experts where the expert is rather ignorant of the correctness of the proposition but thinks that the proposition is more likely to be correct; therefore, an opinion like $\omega_x^C = (0.3, 0.05, 0.65)$ is located in the gray area 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

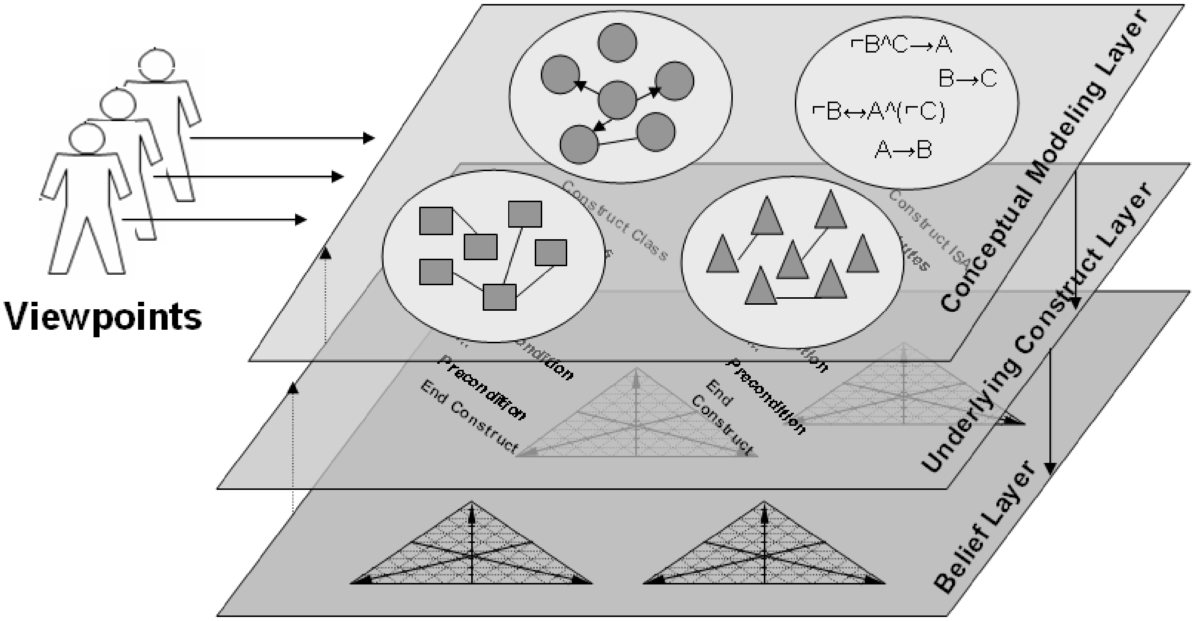


Figure 4. A Proposed Three Layer Model for Model Expression and Integration.

its mathematical 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, he expresses his disagreement with x as 'I firmly disagree with x and I am highly certain about this'. This expression of opinion about x can be converted to $\omega_x^D = (0.1, 0.8, 0.1)$ 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 mathematical opinion values to linguistic terms is also similarly performed.

4.2. The Underlying Modeling Construct

The integration of conceptual models is the issue of many different application domains such as requirement engineering, database schema development, and ontology creation to name a few. For this reason it would be enticing to create an integration model which 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 onto it and the reasoning process be performed based on the lower level construct. The results of the operation on the lower level models can then be mapped back to the higher level conceptual models. Figure 4 depicts how viewpoints can use a known conceptual model for expressing their opinion 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 onto it. 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 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. Preconditions 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 integration model, 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 model, OWL, etc trivially follow the same path and are very similar to what is shown in the following.

```
Construct Class (instanceName)
    Attributes
        name= instanceName
        Cardinality=1
        Contributors=*
    Opinion
        belief=1
        disbelief=0
        uncertainty=0
    Pre-condition
        belief(X, this.name)<disbelief(X, this.name)+ uncertainty(X, this.name)
    Post-condition
        if (belief(Class, this.name)>0.5) {
            disbelief(X-{Class}, this.name)=belief(Class, this.name)
            uncertainty(X-{Class}, this.name)=disbelief(Class, this.name)
                +uncertainty(Class, this.name)
            belief(X-{Class}, this.name)=0
        }
End Construct
```

```
Construct Aggregation (InstanceName, srcName, destName)
    Attributes
        name=instanceName
        Cardinality=1
        Contributors=*
    Opinion
        belief=1
        disbelief=0
        uncertainty=0
    Pre-condition
        belief(X,this.name)<disbelief(X, this.name)+uncertainty(X,this.name)
        belief(Class,srcName)>disbelief(X-{Class},srcName)+uncertainty(X-{Class},srcName)
        belief(Class,destName)>disbelief(X-{Class},destName)+uncertainty(X-{Class},destName)
    Post-condition
        if (belief(Aggregation,this.name)>0.5) {
            disbelief(X-{Aggregation},this.name)=belief(Aggregation,this.name)
            uncertainty(X-{Aggregation},this.name)=disbelief(Aggregation,this.name)
                +uncertainty(Aggregation,this.name)
            belief(X-{Aggregation}, this.name)=0
        }
End Construct
```

As it can be seen in the definition of both concepts, the name of each instance is to be assigned by the analyst, and only one instance of each concept with the same specification is permitted. Opinions are set to complete belief by default that will be over-ridden 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.

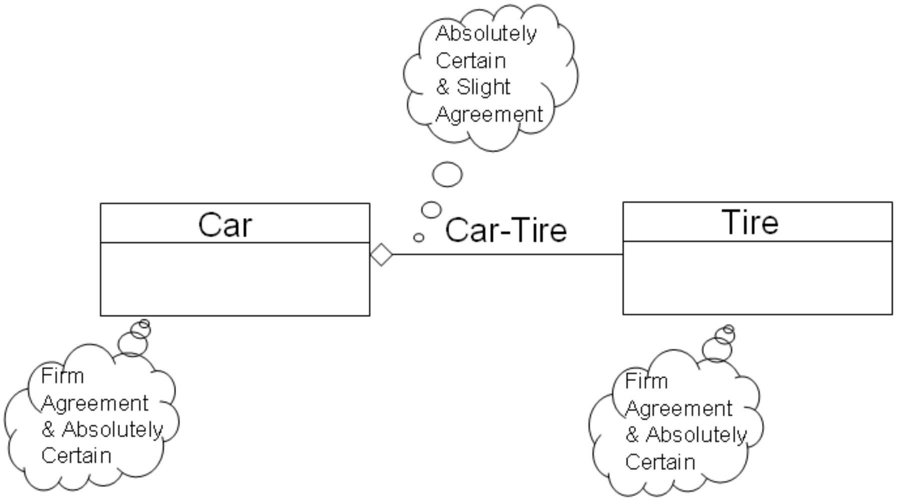

Absolutely Certain & Slight Agreement
Car
Car-Tire
Tire
Firm Agreement & Absolutely Certain
Firm Agreement & Absolutely Certain

Figure 5. A Simple Conceptual Model Expressed in a UML Class Diagram.

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 he/she 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 he/she agrees with the proposition that the concept should be modeled as a property. However on the contrary, if the analyst states that a given concept should be modeled as a class, it can be inferred that he does not believe in it being modeled as a property. 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.

Lets 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 mathematical values. Therefore, the following belief values will be developed:

$\omega_{car\_class} = (0.85, 0.05, 0.1)$,
$\omega_{tire\_class} = (0.85, 0.05, 0.1)$,
$\omega_{car-tire\_aggregation} = (0.65, 0.25, 0.1)$.

Based on these belief values we can see that the pre-conditions 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 need to be added to the set of propositions.

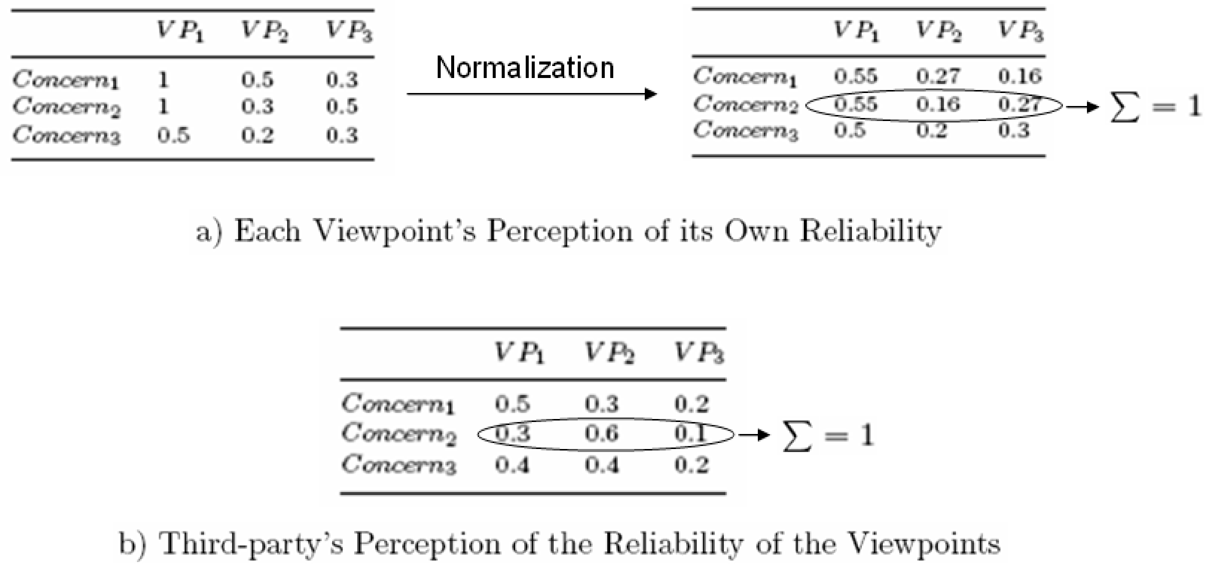
$\omega_{car\_property} = (0, 0.85, 0.15)$,
$\omega_{tire\_property} = (0, 0.85, 0.15)$,
$\omega_{car-tire\_composition} = (0, 0.65, 0.35)$.

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 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 he does not believe in them being a property of a class. The reason that the property 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 [Bowman et al., 1996] and [Clarke, 2002].

4.3. Viewpoint Reliability Structure

In our perception of viewpoint-based requirement specification, concerns are one of the most important aspects of the model. The definition of concern in our model is similar to what has been defined in [Sommerville and Sawyer, 1997]. Based on this definition, each viewpoint may be responsible for considering the information regarding several 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 requirement 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 have two faces. Firstly, the analysts involved in the creation of the specifications for each viewpoint are asked to express their degree of confidence in their understanding and knowledge of that area of concern. Secondly, 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 set equally for all viewpoints, if no extra information about the viewpoints are available. We intend to formalize how third-party reliability values are calculated through the negotiation process. For now, we assume that the third-party reliability value is available. Figure 6 shows how these two reliability measures can be assigned to each viewpoint.



	VP_1	VP_2	VP_3
$Concern_1$	1	0.5	0.3
$Concern_2$	1	0.3	0.5
$Concern_3$	0.5	0.2	0.3

Normalization

	VP_1	VP_2	VP_3
$Concern_1$	0.55	0.27	0.16
$Concern_2$	0.55	0.16	0.27
$Concern_3$	0.5	0.2	0.3

$\sum = 1$

a) Each Viewpoint's Perception of its Own Reliability

	VP_1	VP_2	VP_3
$Concern_1$	0.5	0.3	0.2
$Concern_2$	0.3	0.6	0.1
$Concern_3$	0.4	0.4	0.2

$\sum = 1$

b) Third-party's Perception of the Reliability of the Viewpoints

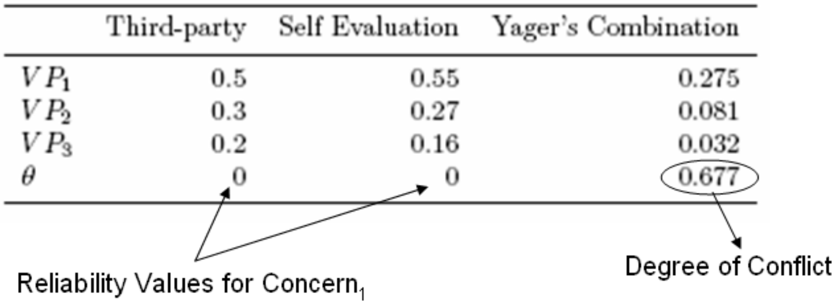
Figure 6. Viewpoints' Reliability Values are Assigned based on Different Concerns.

Viewpoint analysts are initially asked for the perception of their own reliability. They can state their opinion from within the range of $[0, 1]$. As it can also be seen in Figure 6a, the sum of the reliability values assigned to each concern 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 requirement 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 as-

[1] The hypothesis is that the related viewpoint is going to reveal the most useful information in that concern.



	Third-party	Self Evaluation	Yager's Combination
VP_1	0.5	0.55	0.275
VP_2	0.3	0.27	0.081
VP_3	0.2	0.16	0.032
θ	0	0	0.677

Reliability Values for Concern$_1$
Degree of Conflict

Figure 7. The Reliability Values of each Concern for the Viewpoints are Combined using Yager's Rule of Combination.

signed to each viewpoint by Yager's rule. Consider Figure 7 where the reliability of each viewpoint is being analyzed with regard to $concern_1$ following the information in Figure 6. Based on Equations 10 and 11, the degree of conflict is equal to 0.677. The initial degree of reliability assigned to each viewpoint is $0.275, 0.081$, and 0.032, respectively. We now need to proportionally re-distribute the conflicting mass onto the viewpoints. The ratio of the conflicting masses that each viewpoint is going to receive is $0.677 \times 0.71(0.275/0.388), 0.677 \times 0.21(0.081/0.388)$, and $0.677 \times 0.08(0.032/0.388)$, respectively; therefore, the three viewpoints will roughly receive $0.72, 0.2$, and 0.08 of belief assigned to them ($\sum = 1$). In order for the reliability values to be usable as a discounting measure for the expressed requirement 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. Based on this, the reliability belief for each of the viewpoints regarding $Concern_1$ would be $R^{C_1}_{VP_1} = (0.72, 0, 0.28)$, $R^{C_1}_{VP_2} = (0.2, 0, 0.8)$, and $R^{C_1}_{VP_3} = (0.08, 0, 0.92)$. These reliability beliefs can be used to discount the information that have been provided by each viewpoint in a concern.

Figure 8 shows an overview of a sample conceptual model integration process 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 from a single shared perspective (single concern) and we further assume that the analysts are equally reliable which means that each of them is assigned a (0.33, 0, 0.66) reliability belief. The letters in the gray ovals represent viewpoint opinions in linguistic terms (See Figure 3). 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 into the underlying construct. While transforming the conceptual models into 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. Through the conceptual model integration process these discrepancies can be resolved and a final unique model can be reached.

5. EVALUATION MODEL

There are currently two major models that are used for evaluating requirement engineering methods. The first type of evaluation is concerned with the identification of the most important goals of the requirement engineering method, and based on them reasons on the suitability of the approach for that purpose. This model mainly performs a cross-check between the claims of the model and its ability (supporting tools) to fulfil those claims. In the second approach, the proposed model is tested in a real world environment under several case studies. The case studies are designed so that the strengths and the weaknesses of the requirement engineering model is identified. For example to evaluate our proposed model, the case studies should be complex enough so that multi-viewpoint requirement engineering is required for its analysis and also the model should not be so straight forward that the tools for uncertainty management be left intact.

Since our proposal addresses a subset of the requirement engineering process, we intend to undertake both of the evaluation models. For the first type of evaluation, we need to discuss how successful our proposed model is, regarding the issues that were raised in Section 3. The discussion should include the comparison of our proposed model with similar models (e.g. [Sabetzadeh and Easterbrook, 2006]) from the perspective of our intended contributions.

Furthermore, we intend to show the ability of the proposed model to handle the negotiation process among various modelers and analysts by performing several real-world case studies. The case studies will cover the design of a software system with the involvement of at least five graduate and undergraduate Computer Science students. Through the case studies, we will evaluate the suitability of the proposed model by measuring the degree of viewpoint satisfaction from the mediation process and also the resulting integrated model. It is also possible to monitor the affect of the merging effectiveness metrics on the consensus achievement process.

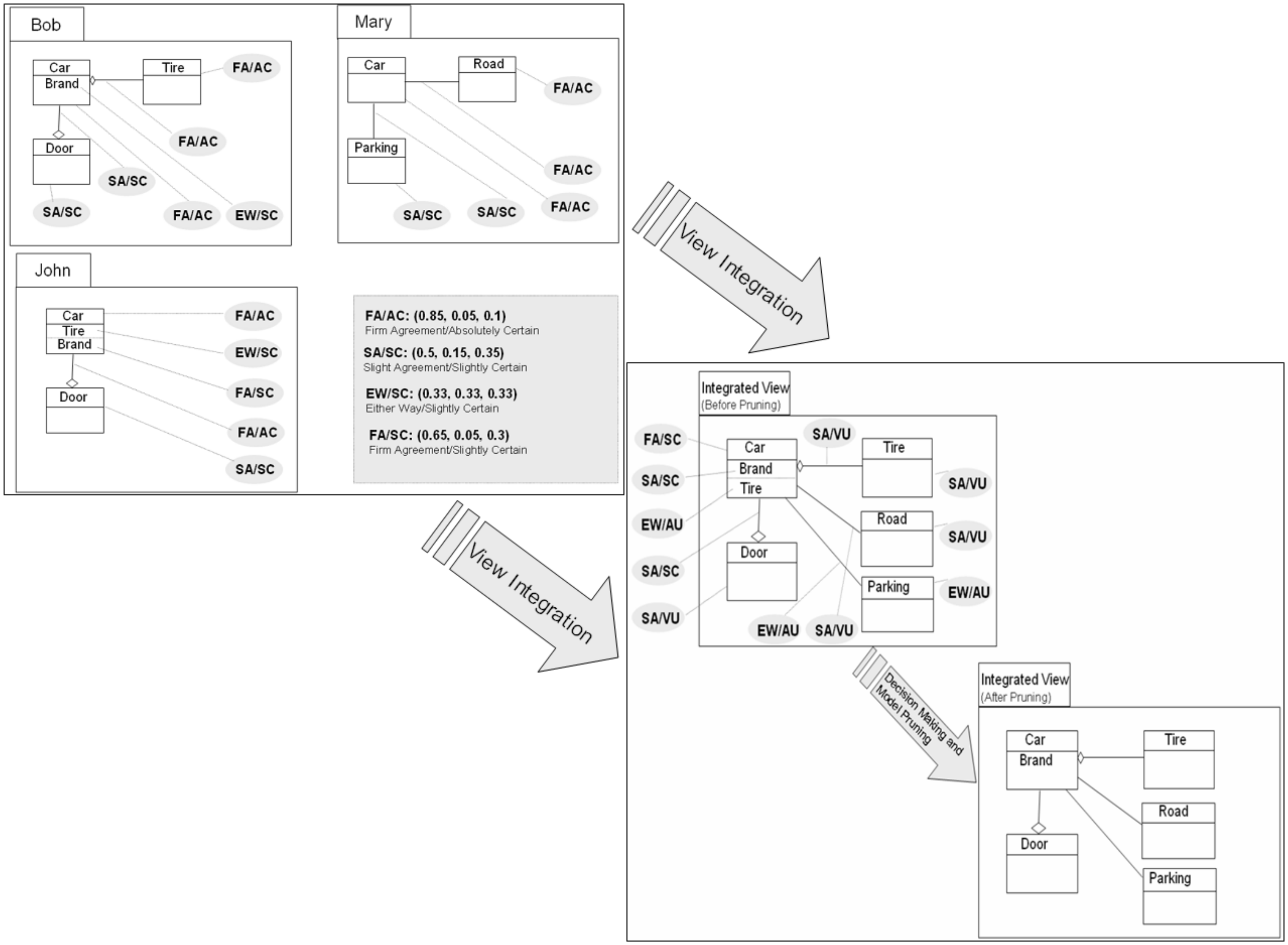


Figure 8. A Sample Integration of Conceptual Models from Multiple Viewpoints.

6. WORK PLAN

The tentative timetable is shown in Table 2. The cited tasks are described below:

Task 1 Design of the Underlying Modeling Construct.

Task 2 Design of the Consensus and Negotiation Model.

Task 3 Design of the Reliability Assessment Model.

Task 4 Creating Proper Consensus Effectiveness Metrics.

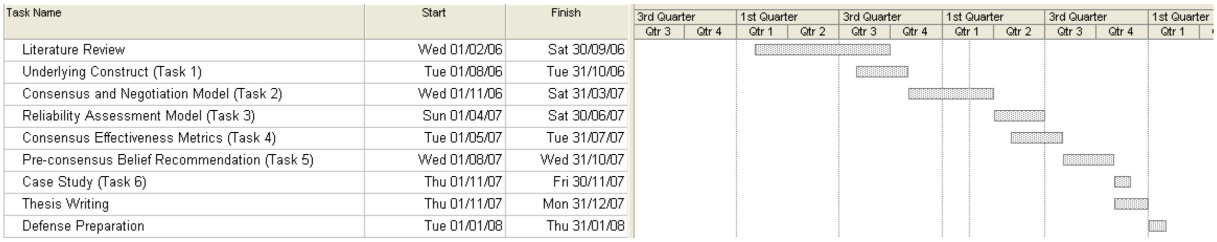
Task 5 Developing the Pre-consensus Belief Recommendation Model.

Task 6 Performing Case Studies and Evaluation.

Task	Duration (months)	Starts	Ends
Literature Review	8	Feb 2006	Sept 2006
Underlying Construct (Task 1)	3	Aug 2006	Oct 2006
Consensus and Negotiation Model (Task 2)	5	Nov 2006	Mar 2007
Reliability Assessment Model (Task 3)	3	Apr 2007	Jun 2007
Consensus Effectiveness Metrics (Task 4)	3	May 2007	July 2007
Pre-consensus Belief Recommendation (Task 5)	3	Aug 2007	Oct 2007
Case Study and Evaluation (Task 6)	1	Nov 2007	Nov 2007
Thesis Writing	2	Nov 2007	Dec 2007
Defense Preparation	1	Jan 2008	Jan 2008

Table 2. Thesis Time Table

The gantt chart of the time table is also shown in Figure 9.



Task Name	Start	Finish
Literature Review	Wed 01/02/06	Sat 30/09/06
Underlying Construct (Task 1)	Tue 01/08/06	Tue 31/10/06
Consensus and Negotiation Model (Task 2)	Wed 01/11/06	Sat 31/03/07
Reliability Assessment Model (Task 3)	Sun 01/04/07	Sat 30/06/07
Consensus Effectiveness Metrics (Task 4)	Tue 01/05/07	Tue 31/07/07
Pre-consensus Belief Recommendation (Task 5)	Wed 01/08/07	Wed 31/10/07
Case Study (Task 6)	Thu 01/11/07	Fri 30/11/07
Thesis Writing	Thu 01/11/07	Mon 31/12/07
Defense Preparation	Tue 01/01/08	Thu 31/01/08

Figure 9. The Gantt Chart Outlining the Proposed Time-line for the Completion of the Ph.D. Program.

REFERENCES

[Andrade et al., 2004] Andrade, J., Ares, J., Garcia, R., Pazos, J., Rodriguez, S., and Silva, A. (2004). A methodological framework for viewpoint-oriented conceptual modeling. *Software Engineering, IEEE Transactions on*, 30(5):282–294.

[Bagheri and Ghorbani, 2007] Bagheri, E. and Ghorbani, A. A. (2007). *A Multi-Perspective Goal-Oriented Risk Analysis Methodology for Critical Infrastructure Protection.* in Critical Infrastructure Protection: Issues and Solutions. Springer, Boston.

[Barr and Wells, 1990] Barr, M. and Wells, C. (1990). *Category theory for computing science*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

[Belnap, 1977] Belnap, N. D. (1977). A useful four-valued logic. In Dunn, J. and Epstein, G., editors, *Modern Uses of Multiple-Valued Logics*, pages 8–37. Reidel, Dordrecht.

[Boehm and In, 1996] Boehm, B. and In, H. (1996). Identifying quality-requirement conflicts. *IEEE Softw.*, 13(2):25–35.

[Bowman et al., 1996] Bowman, H., Derrick, J., Linington, P., and Steen, M. (1996). Cross-viewpoint consistency in open distributed processing. *Software Engineering Journal*, 11(1):44–57.

[Clarke, 2002] Clarke, S. (2002). Extending standard uml with model composition semantics. *Science of Computer Programming*, 44(1):71–100.

[Dardenne et al., 1993] Dardenne, A., van Lamsweerde, A., and Fickas, S. (1993). Goal-directed requirements acquisition. *Sci. Comput. Program.*, 20(1-2):3–50.

[Darke and Shanks, 1996] Darke, P. and Shanks, G. (1996). Stakeholder viewpoints in requirements definition: A framework for understanding viewpoint development approaches. *Requirements Engineering*, V1(2):88–105.

[Easterbrook, 1991] Easterbrook, S. (1991). Handling conflict between domain descriptions with computer-supported negotiation. *Knowl. Acquis.*, 3(3):255–289.

[Easterbrook and Chechik, 2001] Easterbrook, S. and Chechik, M. (2001). A framework for multi-valued reasoning over inconsistent viewpoints. In *ICSE '01: Proceedings of the 23rd International Conference on Software Engineering*, pages 411–420, Washington, DC, USA. IEEE Computer Society.

[Fiadeiro and Maibaum, 1995] Fiadeiro, J. L. and Maibaum, T. (1995). Interconnecting formalisms: supporting modularity, reuse and incrementality. In *SIGSOFT '95: Proceedings of the 3rd ACM SIGSOFT symposium on Foundations of software engineering*, pages 72–80, New York, NY, USA. ACM Press.

[Finkelstein et al., 1994] Finkelstein, A. C. W., Gabbay, D., Hunter, A., Kramer, J., and Nuseibeh, B. (1994). Inconsistency handling in multiperspective specifications. *IEEE Trans. Softw. Eng.*, 20(8):569–578.

[Greenspan and Feblowitz, 1993] Greenspan, S. and Feblowitz, M. (1993). Requirements engineering using the sos paradigm. In *Proceedings of IEEE International Symposium on Requirements Engineering*, pages 260–263.

[Josang, 2001] Josang, A. (2001). A logic for uncertain probabilities. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 9(3):279–311.

[Josang, 2002] Josang, A. (2002). The consensus operator for combining beliefs. *Artif. Intell.*, 141(1):157–170.

[Kulasekere et al., 2004] Kulasekere, E. C., Premaratne, K., Dewasurendra, D. A., Shyu, and Bauer, P. H. (2004). Conditioning and updating evidence. *International Journal of Approximate Reasoning*, 36(1):75–108.

[Leite and Freeman, 1991] Leite, J. C. S. P. and Freeman, P. A. (1991). Requirements validation through viewpoint resolution. *Software Engineering, IEEE Transactions on*, 17(12):1253–1269.

[Liu, 2006] Liu, W. (2006). Analyzing the degree of conflict among belief functions. *Artif. Intell.*, 170(11):909–924.

[Magrez and Smets, 1989] Magrez, P. and Smets, P. (1989). Epistemic necessity, possibility, and truth. tools for dealing with imprecision and uncertainty in fuzzy knowledge-based systems. *International Journal of Approximate Reasoning*, 3(1):35–57.

[Moreira et al., 2005] Moreira, A., Rashid, A., and Araujo, J. (2005). Multi-dimensional separation of concerns in requirements engineering. In *RE '05: Proceedings of the 13th IEEE International Conference on Requirements Engineering (RE'05)*, pages 285–296, Washington, DC, USA. IEEE Computer Society.

[Mylopoulos et al., 1990] Mylopoulos, J., Borgida, A., Jarke, M., and Koubarakis, M. (1990). Telos: representing knowledge about information systems. *ACM Trans. Inf. Syst.*, 8(4):325–362.

[Nejati et al., 2007] Nejati, S., Sabetzadeh, M., Chechik, M., Easterbrook, S., and Zave, P. (2007). Matching and merging of statecharts specifications. In *29th International Conference on Software Engineering.*

[Noy and Musen, 2000] Noy, N. F. and Musen, M. A. (2000). Prompt: Algorithm and tool for automated ontology merging and alignment. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 450–455. AAAI Press / The MIT Press.

[Nuseibeh et al., 1994] Nuseibeh, B., Kramer, J., and Finkelstein, A. (1994). A framework for expressing the relationships between multiple views in requirements specification. *IEEE Transactions on Software Engineering*, 20(10):760–773.

[Pope and Josang, 2005] Pope, S. and Josang, A. (2005). Analysis of competing hypotheses using subjective logic. In *10th CCRTS: The Future of Command and Control.*

[Pottinger and Bernstein, 2003] Pottinger, R. and Bernstein, P. A. (2003). Merging models based on given correspondences. In *VLDB*, pages 826–873.

[Richards and Menzies, 1997] Richards, D. and Menzies, T. (1997). Extending knowledge engineering to requirements engineering from multiple perspectives. In *Proceedings of 3 rdAustralian Knowledge Acquisition Workshop.*

[Robinson and Pawlowski, 1999] Robinson, W. N. and Pawlowski, S. D. (1999). Managing requirements inconsistency with development goal monitors. *IEEE Trans. Softw. Eng.*, 25(6):816–835.

[Rogova and Nimier, 2004] Rogova, G. L. and Nimier, V. (2004). Reliability in information fusion: Literature survey. In *International conference on Information Fusion.*

[Sabetzadeh and Easterbrook, 2006] Sabetzadeh, M. and Easterbrook, S. (2006). View merging in the presence of incompleteness and inconsistency. *Requir. Eng.*, 11(3):174–193.

[Sandri et al., 1995] Sandri, S. A., Dubois, D., and Kalfsbeek, H. W. (1995). Elicitation, assessment, and pooling of expert judgments using possibility theory. *Fuzzy Systems, IEEE Transactions on*, 3(3):313–335.

[Sentz and Ferson, 2002] Sentz, K. and Ferson, S. (2002). Combination of evidence in dempster-shafer theory. Technical report, Sandia National Laboratories, SAND 2002-0835.

[Smets, 2000] Smets, P. (2000). Data fusion in the transferable belief model. In *Information Fusion, 2000. FUSION 2000. Proceedings of the Third International Conference on*, pages 21–33. IEEE Press.

[Sommerville and Sawyer, 1997] Sommerville, I. and Sawyer, P. (1997). Viewpoints: principles, problems and a practical approach to requirements engineering. *Annals of Software Engineering*, V3(0):101–130.

[Spanoudakis and Constantopoulos, 1995] Spanoudakis, G. and Constantopoulos, P. (1995). Integrating specifications: A similarity reasoning approach. *Automated Software Engineering*, V2(4):311–342.

[Spanoudakis and Zisman, 2001] Spanoudakis, G. and Zisman, A. (2001). *Inconsistency Management in Software Engineering: Survey and Open Research Issues*, volume 1.

[Stephanou and Lu, 1988] Stephanou, H. E. and Lu, S.-Y. (1988). Measuring consensus effectiveness by a generalized entropy criterion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 10(4):544–554.

[Yager, 1987] Yager, R. R. (1987). On the dempster-shafer framework and new combination rules. *Inf. Sci.*, 41(2):93–137.

[Zadeh, 1984] Zadeh, L. (1984). Review of shafers mathematical theory of evidence. *Artificial Intelligence Magazine*, 5:81–83.

[Zadeh, 1996] Zadeh, L. A. (1996). *Fuzzy sets*, pages 19–34. in Fuzzy sets, fuzzy logic, and fuzzy systems: selected papers by Lotfi A. Zadeh. World Scientific Publishing Co., Inc., River Edge, NJ, USA.

[Zave and Jackson, 1997] Zave, P. and Jackson, M. (1997). Four dark corners of requirements engineering. *ACM Trans. Softw. Eng. Methodol.*, 6(1):1–30.