algorithms

MDPI

Article

Reasoning about Confidence in Goal Satisfaction

Malak Baslyman [1], **Daniel Amyot** [2,*] and **John Mylopoulos** [2]

[1] Information and Computer Science Department, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia
[2] School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON K1N 6N5, Canada
* Correspondence: damyot@uottawa.ca; Tel.: +1-613-562-5800 (ext. 6947)

Abstract: Goal models are commonly used requirements engineering artefacts that capture stakeholder requirements and their inter-relationships in a way that supports reasoning about their satisfaction, trade-off analysis, and decision making. However, when there is uncertainty in the data used as evidence to evaluate goal models, it is crucial to understand the confidence or trust level in such evaluations, as uncertainty may increase the risk of making premature or incorrect decisions. Different approaches have been proposed to tackle goal model uncertainty issues and risks. However, none of them considers simple quality measures of collected data as a starting point. In this paper, we propose a Data Quality Tagging and Propagation Mechanism to compute the confidence level of a goal's satisfaction level based on the quality of input data sources. The paper uses the Goal-oriented Requirement Language (GRL), part of the User Requirements Notation (URN) standard, in examples, with an implementation of the proposed mechanism and a case study conducted in order to demonstrate and assess the approach. The availability of computed confidence levels as an additional piece of information enables decision makers to (i) modulate the satisfaction information returned by goal models and (ii) make better-informed decisions, including looking for higher-quality data when needed.

Keywords: confidence propagation; data validity; goal modeling; GRL; uncertainty

check for updates

Citation: Baslyman, M.; Amyot, D.; Mylopoulos, J. Reasoning about Confidence in Goal Satisfaction. *Algorithms* **2022**, *15*, 343. https://doi.org/10.3390/a15100343

Academic Editor: Edward Rolando Núñez-Valdez

Received: 2 June 2022
Accepted: 21 September 2022
Published: 23 September 2022

1. Introduction

Consider requirements analysis in a healthcare setting, where a hospital intends to use a real-time wait time estimation system that informs patients of expected waiting times. An analyst builds a goal model capturing design alternatives for the system-to-be. For the analysis, the analyst gathers data such as the manual effort required, and the frequency of tasks, in order to decide how to estimate wait times. However, reliable data, such as the average time it takes to carry out a task manually, are often unavailable, so they have to be estimated on the basis of data available from a similar context, or even guessed on the basis of experience or first principles. The quality of the data used in the decision process may hence influence heavily the confidence the analyst and stakeholders have in the chosen solution for the problem at hand. In this paper, we are interested in estimating that confidence during the goal analysis process, which then influences decision making.

Data availability for goal reasoning is a major challenge, especially in early phases of requirements and design analysis. New system design alternatives may not have been exposed to experimentation and performance testing. This results in uncertainties in the evaluation of these alternatives and, through propagation, in the satisfaction values of high-level goals. Moreover, ignoring the reliability level of sources from which data was collected may lead to uninformed decisions and inaccurate evaluations of high-level goals.

Letier et al. [1] introduced an elaborate, well-developed framework for decision making in the presence of uncertainty, which uses probability distribution techniques in the context of cost/benefit analysis. However, although mathematically sound, this framework is relatively complex and unaffordable in many situations where data is scarce or unavailable. Other proposed approaches, reviewed later in this paper, suffer from the same complexity and data availability concerns. This paper aims to propose a light-weight reasoning mechanism that can cope with incomplete data and is readily adoptable in requirements engineering and decision-making practices. The mechanism determines the *confidence level* of the satisfaction of a given goal can be computed from confidence level values of leaf goals through propagation rules. For example, data collected from a pilot project operationalizing a leaf goal is more robust and of higher quality than purely estimated data acquired by front-end analysts. Accordingly, a higher or lower confidence value of the goal satisfaction value shall be assigned to leaf nodes, and then propagated to higher-level goals.

The main objective of this work is to propose a *Data Quality Tagging and Propagation Mechanism* that enables the tagging of data with a certain quality level, and the propagation of the quality level to assign confidence values to the satisfaction of higher-level goals and actors in a goal model. The data quality level varies based on the reliability of the source from which the data was collected. The approach qualifies the goal satisfaction levels by reflecting the data quality levels of leaf goals and their parents, leading to a more realistic view of goal evaluation in the presence of uncertainty. The confidence levels propagated to top-level goals and actors should influence decision making by having analysts better trust higher-confidence solutions, or by having them look for additional evidence or more reliable data.

Although the ideas introduced herein can apply to many goal-oriented modeling languages, one specific language is used to present and formalize our proposal. We use the *Goal-oriented Requirement Language* (GRL) because GRL is part of an international standard (User Requirements Notation—URN), enables the modeling of stakeholders and their goals, supports indicators for quantitative reasoning, supports contribution relationships, as well as evaluation strategies and propagation algorithms. GRL is a domain-specific modeling language [2] that is also well supported by the jUCMNav tool [3] for evaluating the satisfaction of goals through different strategies.

The proposed approach was applied to a real-world case study in the healthcare domain and informally assessed for feasibility and usefulness, with positive results related to supporting decision making among alternatives and identifying areas where higher-quality data would be needed to sufficiently increase relevant levels of confidence.

The results reported here are important for researchers as they highlight important issues too often neglected in goal-based decision making, and a mechanism that can be adapted to different goal modeling languages. They are also important for practitioners as the proposed new mechanism addresses important concerns and can help decide whether to trust analysis results produced by a goal model.

The rest of the paper is organized as follows: Section 2 presents the main concepts of GRL. Section 3 provides an overview of related work on uncertainty reasoning in goal models. Section 4 explains our new data tagging and propagation mechanism. Section 5 presents the implementation of the mechanism. Section 6 illustrates the usefulness of the proposed approach in guiding decision making with the help of a real healthcare-related case study. Section 7 discusses important challenges in this area, while Section 8 concludes and identifies future work items.

2. Background

In this paper, we use URN's GRL to represent concerns and requirements as goals in a graphical form in order to facilitate understanding and communication with various stakeholders. Figure 1 presents the main elements of GRL including goals, softgoals, tasks, resources, indicators, and the links connecting these elements (AND/OR/XOR decompositions of, contributions to and dependencies among goals). The User Requirements Notation includes, in addition to GRL, a process notation called Use Case Maps (UCM), and various traceability links can be created and exploited between GRL and UCM views, e.g., goal-process alignment.

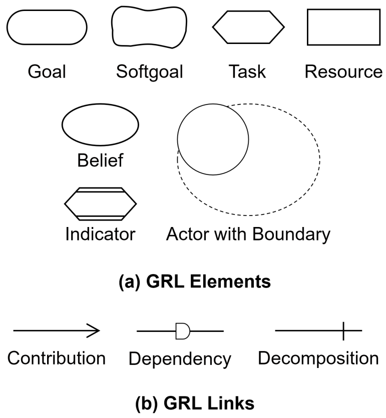

Figure 1. Syntax of GRL's modeling elements.

Actors represent systems and stakeholders, and carry intentions represented by the other GRL elements [4]. Goals are often used to capture high-level functional requirements that can be fully met, whereas softgoals are mainly capturing more quality-oriented aspects related to non-functional requirements (e.g., increased security). Tasks show activities or solutions considered. These model elements may have dependencies on various resources. Indicators (also known as Key Performance Indicators—KPIs) convert observable values in a given unit (e.g., Euro or km/h) to a GRL satisfaction value by comparing it to target, threshold, and worst-case parameters. These elements will be further illustrated in the case study (Section 6), but their nature does not really influence how confidence levels are computed in this paper.

GRL provides analysis techniques that help to resolve and highlight inconsistencies, incompleteness, or conflicts in requirements. The language also shows trade-offs, enabled by GRL strategies (i.e., sets of initial values), among stakeholder intentions. Propagation algorithms are used to propagate the satisfaction values of leaf goals through links to root goals. Satisfaction values are computed using initial satisfaction values of leaf goals and weights/types of links [4]. Satisfaction values can be qualitative (e.g., help, hurt, make, break) or quantitative, e.g., on a scale from 0 (dissatisfied, with the GRL element color-coded in red to 100 (satisfied, color-coded in green). For values in between, different shades of orange, yellow (for the neutral value 50), and chartreuse are used; the greener the better, and the redder the worse. Figure 2 explains the different labels that accompany a sample GRL model with two tasks initialized in a strategy, with the propagated satisfaction level. The C[x] and DQ[x] labels are not part of the standard language but are new types of labels proposed in this paper.

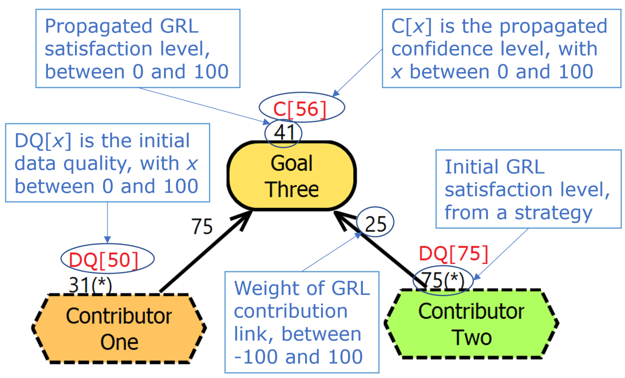

Figure 2. Explanation of standard and new GRL labels used in this paper.

Fan et al. [5] formalized the standard propagation algorithm using arithmetic equations that express how the propagated satisfaction values ($v(\ldots)$) are computed based on the type of GRL links connecting the GRL elements (this time independently of their type that is presented in Figure 1):

- *AND-Decomposition*: the minimum satisfaction value of sub-goals is propagated (see Figure 3), as expressed in Equation (1);

$$v(S) = Min(v(D_1), v(D_2), \ldots, v(D_n)), \quad (1)$$

where $v(S)$ is the satisfaction value of the decomposed source GRL element and $v(D_x), 1 \leq x \leq n$ are the satisfaction values of the decomposing GRL elements (i.e., the destinations of the decomposition link). In Figure 3, and others like it, note the color coding that reflects the satisfaction level of each element, as discussed earlier. In addition, in the jUCMNav tool used to produce these evaluated GRL models [3], the GRL elements that are initialized in a strategy are shown with (*) next to the quantitative satisfaction value, and with the element contour displayed with a dashed line. The other model elements have satisfaction values computed (using propagation rules) from initialized or already computed elements.

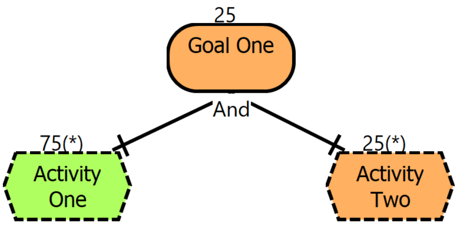

Figure 3. Propagation of element satisfactions in the case of an AND-decomposition.

- *OR-Decomposition*: using the same notation conventions, here the maximum satisfaction value among the decomposing elements is propagated, as shown in Equation (2) and Figure 4. GRL's XOR-decomposition is computed in the same way, with a warning if more than one decomposing element has a satisfaction value different from 0.

$$v(S) = Max(v(D_1), v(D_2), \ldots, v(D_n)), \quad (2)$$

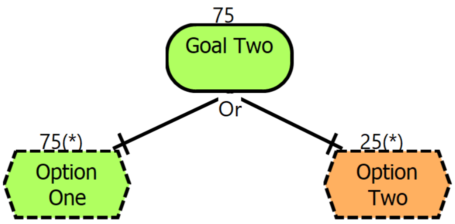

Figure 4. Propagation of element satisfactions in the case of an OR-decomposition.

- *Contribution*: in GRL, contributions can have a type/weight that indicates whether it is negative, neural (0), or positive, with a weight between 0 and 100. Hence, the contribution scale goes from -100 to $+100$. Here, the product of the weighted contribution values (C_x and the satisfaction values ($v(D_x)$) of the n contributing elements are propagated to the targeted GRL element, as illustrated in Figure 5 and Equation (3). Note that the final satisfaction is also truncated to a value between 0 and 100, inclusively:

$$v(S) = Max(0, Min(100, \frac{\sum_{x=1}^{n}(v(D_x) \times C_x)}{100})) \quad (3)$$

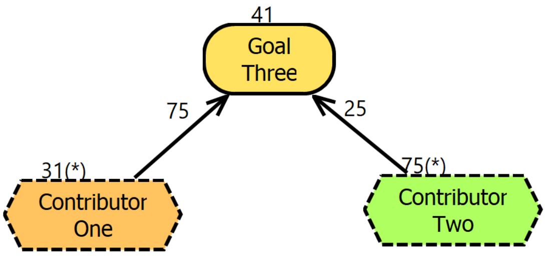

Figure 5. Propagation of element satisfactions in the case of contributions.

- *Dependency*: the satisfaction of the dependent element ($v(S)$) is truncated to the minimum satisfaction value of its depending elements (see Figure 6, where $v(S) \geq 25$ initially, and Equation (4)). This means that the GRL element that is the source of dependencies cannot be better satisfied than the other elements it depends on.

$$v(S) = Min(v(S), v(D_1), v(D_2), \ldots, v(D_n)) \tag{4}$$

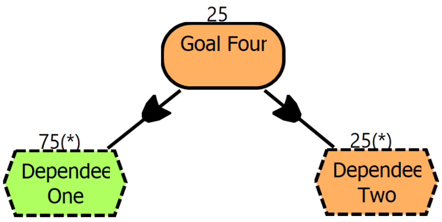

Figure 6. Propagation of element satisfactions in the case of dependencies.

3. Related Work

As indicated in Section 1, an important approach for managing uncertainty in goal models is proposed by Letier et al. [1]. Their proposal adopts a statistical decision-theoretic technique to support decision making under uncertainty. Probability distributions are used to represent uncertainties of alternatives for a decision. In addition, the study uses Monte Carlo simulations to assess the impact of uncertainties on a goal model, as well as Pareto-based multi-objective optimization techniques to guide alternative selection. This work was extended with the RADAR modeling language that optimizes and shortens the selection of architectural design in the presence of uncertainty. It was also implemented to automate developing the decisions model and its simulation [6,7]. Adopting a different approach, Cailliau and Lamsweerde extended the probabilistic goal/obstacle specification language to handle knowledge uncertainty [8]. Their approach provides probability-based metrics and a method to identify uncertainties about goal satisfaction and to highlight the most impacting obstacles on a goal model. Similar work was done in this area by Sabetzadeh et al. [9], especially in a technology qualification context. Chen et al. proposed another approach that used probabilistic models and analysis to handle uncertainty in goal-driven self-optimization processes [10]. More recently, Liaskos et al. explored extensions of the *i** goal modeling language (e.g., with probabilistic effects for tasks) to allow representing and reasoning about both uncertainty in the environment and preferential utility in goals [11].

Other approaches of a less probabilistic nature have been proposed to address uncertainty in goal models. Salay proposed a language-independent framework called MAVO to annotate and analyze formally uncertainty, targeting incompleteness in models [12]. Bowers et al. developed a search-based technique (Providentia) to address environmental and system uncertainty while analyzing and choosing the optimal non-functional requirements configuration for self-adaptive systems [13]. Sources and impact of uncertainty are specified in the executable specification to guide the selection of an optimal configuration using Genetic Algorithm (GA). Similarly, Fredericks et al. proposed AutoRELAX, which applies GA and executable specification to explore possible goal model configurations under environmental uncertainty constraints [14]. Zawawy et al. propose the use of Markov logic to support root cause analysis, using goal models [15]. Another interesting method to support strategic decisions about Information Technology initiatives under uncertainty is presented by Dąbrowski [16], while Pasquale et al. use quantified values between 0 and 1 associated with security requirements in a goal model to deal with uncertainty [17].

Bayesian Belief Networks (BBNs) have also been used in many methods to estimate the confidence of an argument. Hobbs and Lloyd reported on the power and flexibility of BBN to represent the structured argument of an assurance case, where a claim is supported with multiple evidence items characterized by different degrees of confidence [18]. In a similar context, Guiochet et al. proposed an approach to identify and estimate confidence in a safety case [19]. The safety case is modeled using the Goal Structuring Notation and the confidence of supporting arguments of a claim is estimated quantitatively and propagated using BBNs. There exist several other proposals in the area of uncertainty and confidence reasoning in goal models, as well as other approaches such as Hall's [20], where goal modeling and data mining techniques are combined.

Most of the above approaches, such as [1,6,9,18,20] suffer from a common weakness for informed decision making: they rely on an unrealistic amount of information (e.g., probabilities) that is hard to obtain in many practical contexts, and they require a good understanding of probabilistic reasoning. They also assume that historical data, regardless to the volume, is available for the analysis. However, when designing a new (innovative) system, data barely exists and decisions are made based on available evidence regardless of the source and context in which the evidence was observed. In our study, we provide a solution to overcome the paucity of data issue when reasoning about goal models in the presence of uncertainty. That happens by tagging leaf goals with numerical values that reflect the reliability of sources from which data was collected or estimated. Our proposal improves upon existing approaches by enabling the confidence analysis of evaluated goal model elements in the presence or absence of data, which makes it a practical solution in early stages of requirements engineering. It also brings a new dimension in computing the confidence of goal model elements during evaluations by considering the reliability of data sources and their impact on the goal model evaluations. The approach introduced in the next section relies on simple, coarse-grained assessment of confidence of inputs based mainly on data quality types that characterize how they were obtained.

4. Data Quality Tagging and Propagation Mechanism

The method presented in this section consists of two complementary sub-methods: *Data Quality Tagging* and *Propagation Mechanism*. In the former, the modeler tags leaf goals with a qualitative data quality level, which is then converted to initial quantitative confidence levels. In the latter, the confidence levels of leaf goals are propagated to other goals through their links (decompositions, contributions, and dependencies). Both methods are explained in the following subsections.

Uncertainty has been explored for modeling and analysis at design time [21] and also at run-time [22]. Our focus here is on supporting decision making based on goal models (mainly at design time, but also with data that reflects run-time snapshots) in the presence of uncertainty. In terms of the tridimensional conceptual basis introduced by Alwidian et al. [23] for uncertainty in goal modeling, our work focuses on: (1) *location*: uncertainty in inputs and outputs represented by the satisfaction levels of the model elements, (2) *level*: scenario uncertainty (e.g., without statistical evidence), and (3) *mature*: epistemic uncertainty (i.e., that could be reduced empirically, but likely with costly efforts).

4.1. Data Quality Tagging

In goal modeling, there are several systematic approaches for determining quantitative contribution and importance levels in goal models, such as the use of the Analytical Hierarchy Process (AHP) [24] and Delphi-based consensus [25], applied to GRL [26] and other goal modeling languages [27–31]. In our context, we want to provide quantitative values to data quality qualitative types. The data quality types are identified based on the procedure followed to collect or obtain the data. For each type, we have a qualitative value and a corresponding quantitative value. A qualitative value reflects the validity of the source from which data was collected. We identified sources and the associated qualitative attribute through several iterations of validation with industry practitioners, mainly from a healthcare domain. In addition, we propose a tagging mechanism that estimates the impact of the quality of collected data on goal satisfaction levels by giving it a quantitative *confidence value*, a concept that does not exist in standard GRL. These quantitative values were also identified through a consensus-building process among industry practitioners. Table 1 defines the proposed types of quality and corresponding confidence levels. Constructing the table is essential to propagate the confidence value regardless of the data quality types, which could change from one context to another.

Table 1. Data quality types and corresponding confidence values, in a decision-making context related to design alternatives.

Quality Type	Confidence Value	Definition
Valid	100	Data already measured and available for the design alternative, in the same context as the one under evaluation
Borrowed-Valid	100	Data already measured and available for the design alternative, in a context similar to the one under evaluation
Borrowed	75	Data already measured and available for the design alternative, but in a different context
Estimated-Context	50	No data available, but it was estimated according to a similar design alternatives in a different context
Estimated-Literature	25	No data available, but it was estimated based on the literature or previous studies
Unknown	0	No data-driven evidence will be used in the evaluation

Valid (100) and *Borrowed-Valid (100)* are the best data quality types because the data have been measured and validated, resulting in the highest confidence. An example of data tagged with *Borrowed (75)* in a healthcare system context is data collected for patient information documentation in an Emergency Room but that will be reused in documenting patient information in a Surgery Room. *Estimated-Context (50)* is meant to be used when data is not available but can be estimated based on similar tasks. For example, one could estimate data for 'write patient report after a surgery' based on data of a similar activity such as 'write patient report for consultation'. In case data cannot be estimated from similar activities, it could be estimated from the literature or studies of similar systems, and be tagged with *Estimated-Literature (25)*. *Unknown* is used to highlight missing data and its nullifying impact on the interpretation of goal/element satisfaction.

In each evaluation strategy corresponding to a certain design alternative, the confidence of goal-model element satisfaction values will be calculated and propagated. The following section presents how the confidence of a satisfaction value is propagated along different GRL links between model elements.

4.2. Propagation Method

GRL has four main types of links: AND-decomposition, XOR/OR-decomposition, contribution, and dependency (see Figure 1). For each type, the confidence level of the satisfaction of a parent goal in a model is calculated differently from the confidence of children goals. The algorithm for calculating confidence values is adapted from the standard GRL CalculateEvaluation algorithm [4]. Confidence is computed in an integrated manner from the decomposition, contribution, and dependency relationships, in that order. It is important to note that confidence can be computed *independently* from the satisfaction values of goals; for example, a goal could have a high satisfaction with a low confidence or a low satisfaction with a high confidence.

AND-decomposition: the confidence of the parent goal satisfaction is equal to the *average* of its sub-goals' confidence values. Unlike satisfaction propagation (where the minimum satisfaction is propagated to the parent goal), the average is used for confidence because:

- The confidence of the sub-goal with minimum satisfaction value might be different from the minimum confidence among all sub-goals of the AND-decomposition.
- The confidence of the sub-goal with minimum satisfaction value might be much higher or lower than the confidence of the other sub-goals. Yet, all sub-goals are taken into consideration during the propagation decision.

Since the satisfaction of the parent goal is computed based on the satisfaction levels of all of its sub-goals in the AND-decomposition, taking the confidence levels of all sub-goals into consideration in an AND-decomposition context (as formalized in Equation (5)) is hence a reasonable approach. As there are no explicit weights associated with AND-decomposition links, we assume here equal weights by using an average function. Note also that the average is preferable here to the maximum and minimum functions as the maximum of the confidence values is too optimistic in an AND-decomposition context whereas the minimum is too pessimistic, especially as the actual confidence level of the propagated satisfaction value might be ignored in both these alternative functions. Additionally, only propagating the confidence level of the selected (and lowest) satisfaction values is deemed insufficient here as this solution would ignore the confidence levels of the other considered nodes, which might be much lower or higher.

Figure 7 illustrates how the confidence of the satisfaction of the goal model element is computed for AND-decompositions. The average of 50 and 75 is 63, which is propagated to the top goal. Note that GRL models also include other types of values that could require some confidence assessment, for instance contribution values. However, as discussed in the previous section, several techniques already exist for reaching agreement on contribution weights [26–31]. New data quality (DQ) and confidence (C) values are highlighted in red in the figure. Note that for a model element initially tagged with DQ[x], the confidence is also that value (C[x] = DQ[x]).

$$cv(S) = \frac{\sum_{x=1}^{n} cv(D_x)}{n} \tag{5}$$

where $cv(S)$ is the confidence value of the decomposed GRL element and $cv(D_x), 1 \leq x \leq n$, are the confidence values of the n sub-elements.

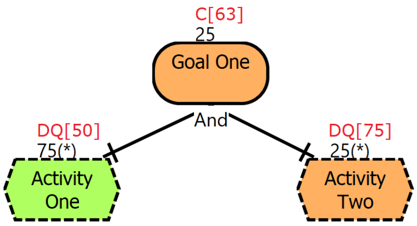

Figure 7. Confidence of element satisfactions in the case of an AND-decomposition.

XOR/OR-decomposition: the confidence of a parent goal satisfaction value is equal to the confidence value of its *maximally* satisfied sub-goals. In case there are more than one sub-goal sharing the maximum satisfaction, the parent's confidence level becomes the average confidence of the maximally satisfied sub-goals for the OR-decomposition (see Figure 8 and Equation (6)). For the XOR-decomposition, the maximum confidence among the maximally satisfied sub-goals is selected (as $i = 1$ in the equation). As the OR-decomposition is about selecting one or many alternatives and the XOR-decomposition only one alternative, only the confidence levels of the selected alternatives are considered by the confidence propagation mechanism. This is different from the AND-decomposition, where the average of all confidence values was considered, because the XOR/OR-decomposition is an optimistic decomposition operator by nature, and is concerned with selecting some options while ignoring others.

$$cv(S) = \frac{\sum_{x=1}^{n} cv(D_x) \quad \textbf{where } v(D_x) = Max((D_1)..(D_n))}{i \quad \textbf{where } i = Count(Max((D_1)..(D_n)))} \tag{6}$$

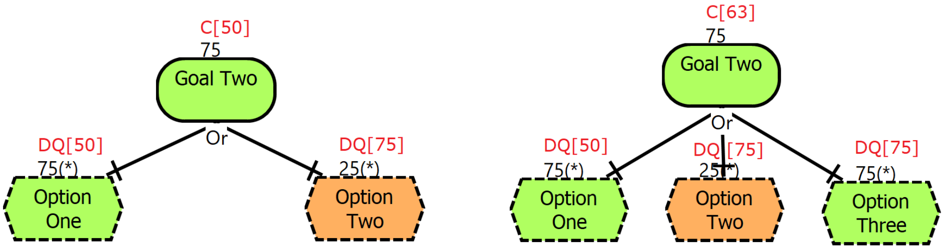

Figure 8. Confidence of element satisfactions in the case of an OR-decomposition. (**Left**): confidence of maximally satisfied sub-goal propagated to the parent goal. (**Right**): average confidence of maximally satisfied sub-goals propagated to the parent goal.

Contribution: the confidence of the satisfaction of a goal that is the target of contributions is the sum of the product of the contribution values of sub-goals and their confidence values, over 100 (and truncated to an integer value between 0 and 100), as formalized in Equation (7). C_x is a contribution level between -100 and 100. In the example of Figure 9, $(50 \times 75 + 75 \times 25)/100 = 56$. If the sum of the contribution weights is larger than 100 (overcontribution), in order to avoid 'confidence building', the computed confidence level is normalized. This is a mechanism similar to the propagation of satisfaction values in GRL, where all the contributions and their weights are considered and truncated whenever necessary.

$$cv(S) = Max(0, Min(100, \frac{\sum_{x=1}^{n}(cv(D_x) \times C_x)}{100})) \tag{7}$$

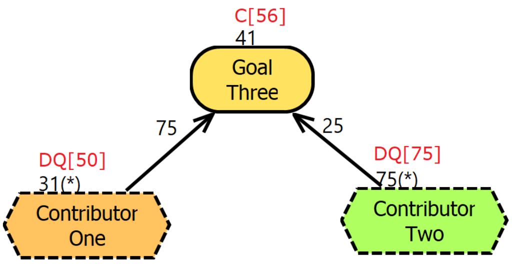

Figure 9. Confidence of element satisfactions in the case of a contribution.

Dependency: the confidence of a dependent element is its current confidence level (if any) when the depending elements all have higher (or equal) satisfaction levels than that dependent element's satisfaction (see Equation (8)). However, if some depending elements have lower satisfaction levels, then the confidence is computed as the minimum between the current confidence level (if any) and the confidence levels of the depending elements with the lowest satisfaction level (see Figure 10). This is a conservative propagation of confidence, and this choice is again dictated by the nature of the dependency link in GRL, which aims to identify locations for conservative evaluations in GRL models.

$$cv(S) = \begin{cases} cv(S) & \textbf{if } cv(S) \neq 0 \textbf{ and } v(S) < Min(v(D_1)..v(D_n)) \\ else & \frac{\sum_{x=1}^{n} cv(D_x) \quad \textbf{where } v(D_x) = Min((D_1)..(D_n))}{i \quad \textbf{where } i = Count(Min((D_1)..(D_n)))} \end{cases} \tag{8}$$

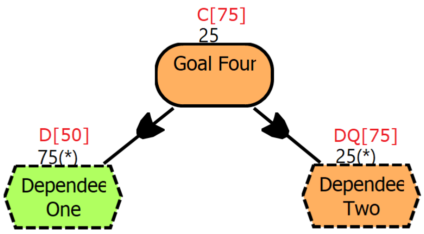

Figure 10. Confidence of goal satisfaction in the case of dependency.

More generally, computing the confidence for the satisfaction of a goal that is the destination of multiple dependency relationships *and* contribution relationships is done by first handling decomposition confidence values, then contribution values (the confidence previously computed from the decomposition is considered as another contribution), and finally dependency values. In the latter case, the confidence of the destination goal's satisfaction is the average between the confidence of the dependency relationships and the contribution's confidence values.

Alternative Selection: in a goal model, there could be several design alternatives that could be used in combination to evaluate the satisfaction of a goal. The confidence propagated to top-level goals and other elements reflects only the confidence of the chosen design alternative where the other alternatives' impact is considered to be *absent* if no confidence value is assigned to them. For the non-selected alternatives, an A letter (meaning: absent) is propagated by default to indicate that there could be some impact of other alternatives, which are linked to the top-level model elements, on the propagated confidence levels, but that impact is not considered in a particular evaluation strategy (see Figure 11).

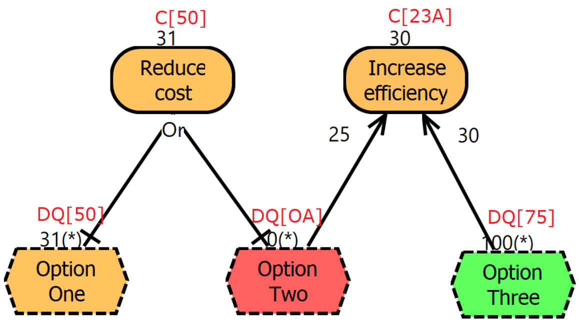

Figure 11. Propagation of "A" labels with confidence values, which are used to highlight the *absent* impact of non-selected alternatives (e.g., Option Two here).

5. Implementation

The proposed approach has been prototyped in jUCMNav [3], in support of a larger activity-based process integration approach that exploits goal and process models while taking uncertainty into account [32]. Data quality and the computation of the confidence levels of element satisfactions values are new to GRL. A confidence value is handled in a way similar to a satisfaction value, this time however with the propagation rules illustrated in Section 4. As confidence values do not exist in the standard URN metamodel [4], they are currently prototyped using URN *metadata*, which are essentially name-value pairs that can contain domain-specific labeled information.

The leaf intentional elements (including indicators) of a GRL model are annotated with initial confidence values computed from data quality information (DQ[x]). The algorithm for propagating confidence values (*CalculateEvaluationAndConfidence*) extends the *CalculateEvaluation* algorithm of standard GRL [4]. The algorithm, and the non-primitive data types are classes from the URN metamodel. The *CalculateEvaluationAndConfidence* algorithm generates a new confidence value (between 0 and 100) that can then be also stored as a metadata for the intentional element being evaluated. The algorithm invokes three sub-algorithms

(*CalculateDecompositions*, *CalculateContributions*, and *CalculateDependencies*), which are also modified. Once *CalculateEvaluationAndConfidence* has completed, the *ActorSatisfaction* algorithm (also modified) can be invoked to compute the satisfaction and confidence of an actor, and then the confidence can be stored again as metadata attached to that actor. The details of the algorithm can be found in Section 4.4 of Baslyman's thesis (https://ruor.uottawa.ca/bitstream/10393/38104/3/Baslyman_Malak_2018_thesis.pdf, accessed on 2 June 2022) and the corresponding Java module for jUCMNav is also available online (https://www.site.uottawa.ca/~damyot/pub/Baslyman/QuantitativeGRLStrategyAlgorithm.java, accessed on 2 June 2022).

Note that our extensions to GRL are simple and aim to capture data quality and confidence levels, mainly to support a type of analysis about confidence that is complementary to the existing propagation of satisfaction values coming from strategies. There exist processes to rigorously extend modeling languages, for instance the PRISE process [33] for *i**, which is a goal modeling language that shares many similarities with GRL (and from which GRL was itself derived). Such a process enables adapting *i** to a specific domain, e.g., human-centric users [34]. Given that the extension here is not for a specific domain but mainly to support a new type of analysis, it was not deemed necessary to use PRISE in our context.

6. Case Study

Case studies are a common software engineering research method that studies the application of an artifact, in our case the proposed Data Quality Tagging and Propagation Mechanism, in its real world setting. A *descriptive case study* focuses on one instance or a small group of instances to describe a certain situation and demonstrate how these instances interact given certain conditions [35]. Each case study has a context and a unit of analysis. In our research, we conducted a descriptive case study to investigate to what extent Data Quality Tagging and Propagation Mechanism, the *unit of analysis*, supports the decision-making process in the *context* of a new monitoring system deployment in healthcare. This case study enables applying the mechanism to a real-life situation, in situ, and collect evidence of usefulness regarding support for organizational decision-making and the detection of low levels of confidence that could help identify needs for higher-quality data that feed indicators. The *data* used to prepare the input to the approach was collected through several meetings with stakeholders and through existing documents and processes. The input of the approach consists of the GRL goal model (including indicators), processes definitions, and the new monitoring system design alternatives. The case study was conducted over six months, in collaboration with the Quality and Safety team in a Saudi hospital.

The case study is particularly concerned with tracking the position of lab samples from the Emergency Room (ER) to the lab unit at Al-Rass Hospital in Saudi Arabia. In their ER process, when a physician requests a lab test, the level of urgency (Critical, Urgent, or Routine) shall be set. A nurse collects the sample, and then registers the patient and the sample information into the Medical Health Record (MHR). Then, a transporter carries the sample to the lab unit and drops it into a box. Nurses in the lab unit check the box every 15 min for new samples because they do not receive notifications upon the arrival of new samples. Therefore, in most cases, there is a delay in delivering the results of samples.

A customized *Real-Time Tracking Sample system* (RTTS) is being proposed to track samples in real time and report on the current position of samples while travelling from the ER to the lab. The samples are put into bags tagged with tracking chips that communicate wirelessly with the RTTS; then, the RTTS sends notification about the current position and time. There are two main possible alternatives for integrating the RTTS-related tasks into the current ER process:

1. having patient information entered into the RTTS by a nurse, or
2. having the RTTS pull patient information automatically from the MHR.

The aim of this case study is to illustrate and informally assess how our method can help and support the decision that should be made, by the hospital, on whether to deploy the RTTS system or not, given that the hospital greatly needs to improve and evaluate the ER performance for critical cases. The RTTS system was developed by a programmer in accordance with Al-Rass Hospital's needs. Figure 12 illustrates the goal model of the context. The contributions weights and the goals' importance values were gathered using AHP, according to Akhigbe's approach for GRL models [26].

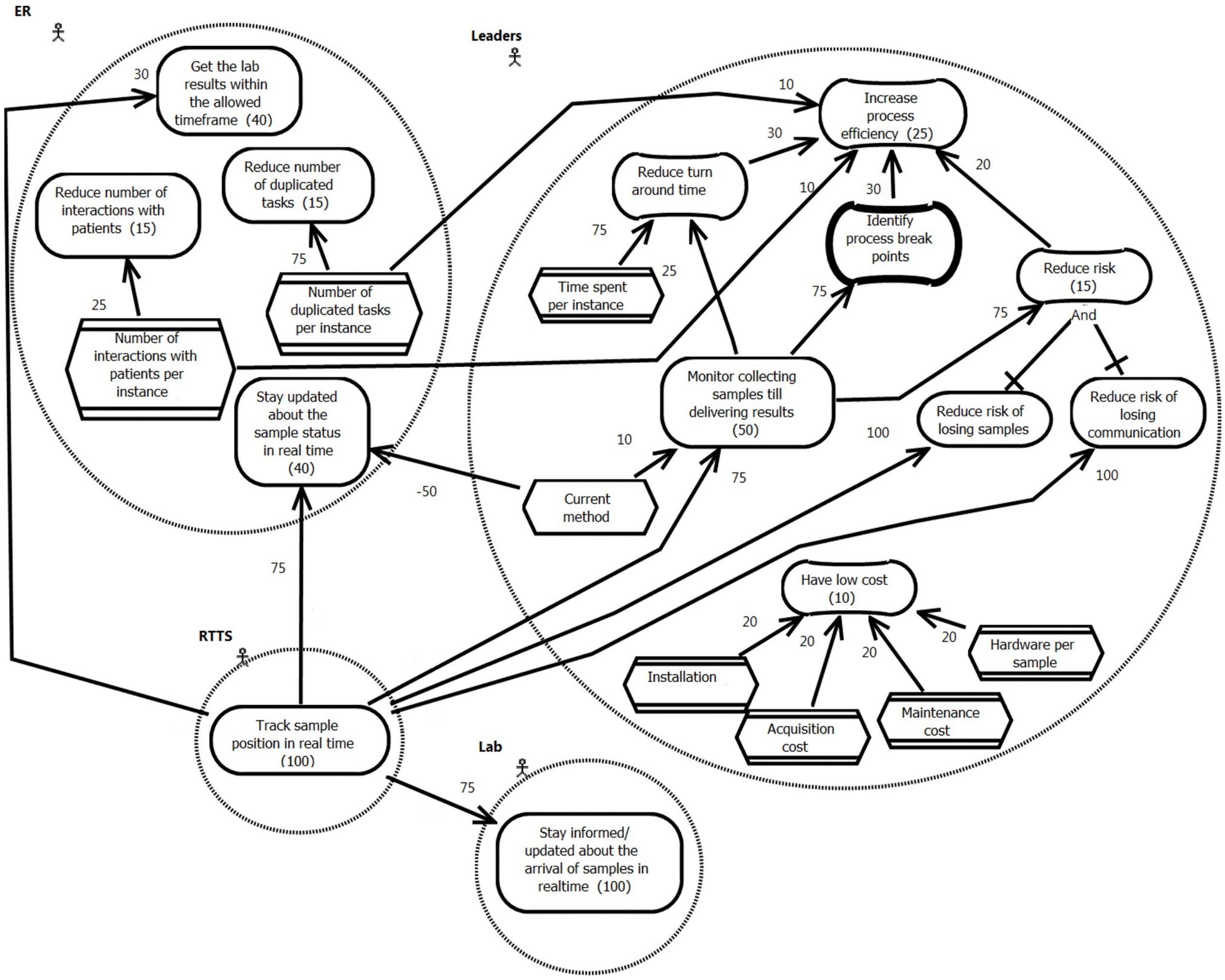

Figure 12. Goal model of the lab sample monitoring context.

We designed GRL evaluation strategies that correspond to each design alternative, to satisfy the two main stakeholders interests in this case study, namely: (1) evaluate the possibility of adopting and deploying the RTTS across the hospital (for other monitoring purposes such as monitoring the location of medical equipment), and (2) investigate the performance of the integration alternatives on the predefined criteria for each level of urgency. There are nine GRL evaluation strategies in total, which correspond to each combination of the three process integration alternatives (*current method—no RTTS*, *RTTS by nurse*, and *automated RTTS*) and each of the three levels of urgency (critical, urgent, and routine).

6.1. Application of Data Quality Tagging and Propagation Mechanism

Using our Data Quality Tagging and Propagation Mechanism, the indicators' evaluation values were tagged with a certain level of data quality (Table 1) according to the sources from which these values were collected (see Table 2). The indicators capture two aspects of the evaluation: (1) cost, and (2) process efficiency. The total yearly cost is composed of four sub-indicators: installation, acquisition, maintenance, and hardware costs. They are calculated in Saudi Riyals (SAR). The data quality values of the cost indicators are set to 25 (Estimated-Literature, see Table 1) because the hospital was not certain about the costs and they provided estimated costs based on an initial discussion with the RTTS programmer. The data quality values of the Number of interactions with patients per instance, Time spent per instance, and Number of duplicated tasks per instance in other indicators are set to 100 because the evaluation values were collected after conducting a pilot project, and hence they are reliable. The presence of these indicators is essential to the decision-making context to improve process efficiency and to better use the resources where they are needed, such as caregivers' time and effort.

Table 2. Data quality of indicators for each evaluation strategy.

Indicator	Definition	Unit	Data Quality
Total yearly cost	Summation of four other indicators: software installation cost, software acquisition cost, software maintenance cost, and the hardware cost per sample	SAR	Estimated-Literature 25
Installation cost	Software installation cost	SAR	Estimated-Literature 25
Acquisition cost	Software acquisition cost	SAR	Estimated-Literature 25
Maintenance and operation cost	Software maintenance cost	SAR	Estimated-Literature 25
Hardware cost per sample	Cost of the tracking chip	SAR	Estimated-Literature 25
Number of interactions with patients per instance	Number of interactions between a nurse and a patient inquiring about the lab sample result	Number	Valid 100
Number of duplicated tasks per instance	Number of duplicated tasks per instance in an alternative	Number	Valid 100
Time spent per instance	Time from collecting the lab sample to its delivery to the lab unit	Second	Valid 100

Then, the confidence levels of the satisfaction values of higher-level elements in the goal model were propagated using the rules presented in Section 4. Regardless of the indicator evaluation values in each of the GRL evaluation strategies, the propagated confidence values depend on the source and the alternative in which there were collected. Hence, the propagated confidence values change from one alternative to another (current method, by nurse, and using the automated RTTS) but they remain the same in the three urgency contexts (critical, urgent, and routine). Table 3 presents the propagated confidence values of each alternative in the critical context (see Figure 13).

Table 3. Propagated confidence levels of top-level goals in the critical context. **Sat** is the satisfaction value and **Conf** is the propagated confidence value.

	By Nurse		Automated RTTS		Current Method	
Goals	**Sat**	**Conf**	**Sat**	**Conf**	**Sat**	**Conf**
Increase process efficiency	70	66	76	66	23	30
Reduce turn around time	84	94	84	94	59	20
Identify process break points	56	56	56	56	7	8
Reduce risk	100	57	100	57	7	8
Monitor collecting samples till delivering results	75	75	75	75	10	10
Have low cost	52	20	46	20	100	20
Get the lab results within the allowed timeframe	30	30	30	30	0	0
Reduce number of interactions with patients	12	25	12	25	0	25
Reduce number of duplicated tasks	30	75	75	75	22	75
Stay updated about the sample status in real time	75	75	75	75	0	100
Stay informed/updated about the arrival of samples in real time	100	100	100	100	0	100
Track sample position in real time	100	100	100	100	0	0

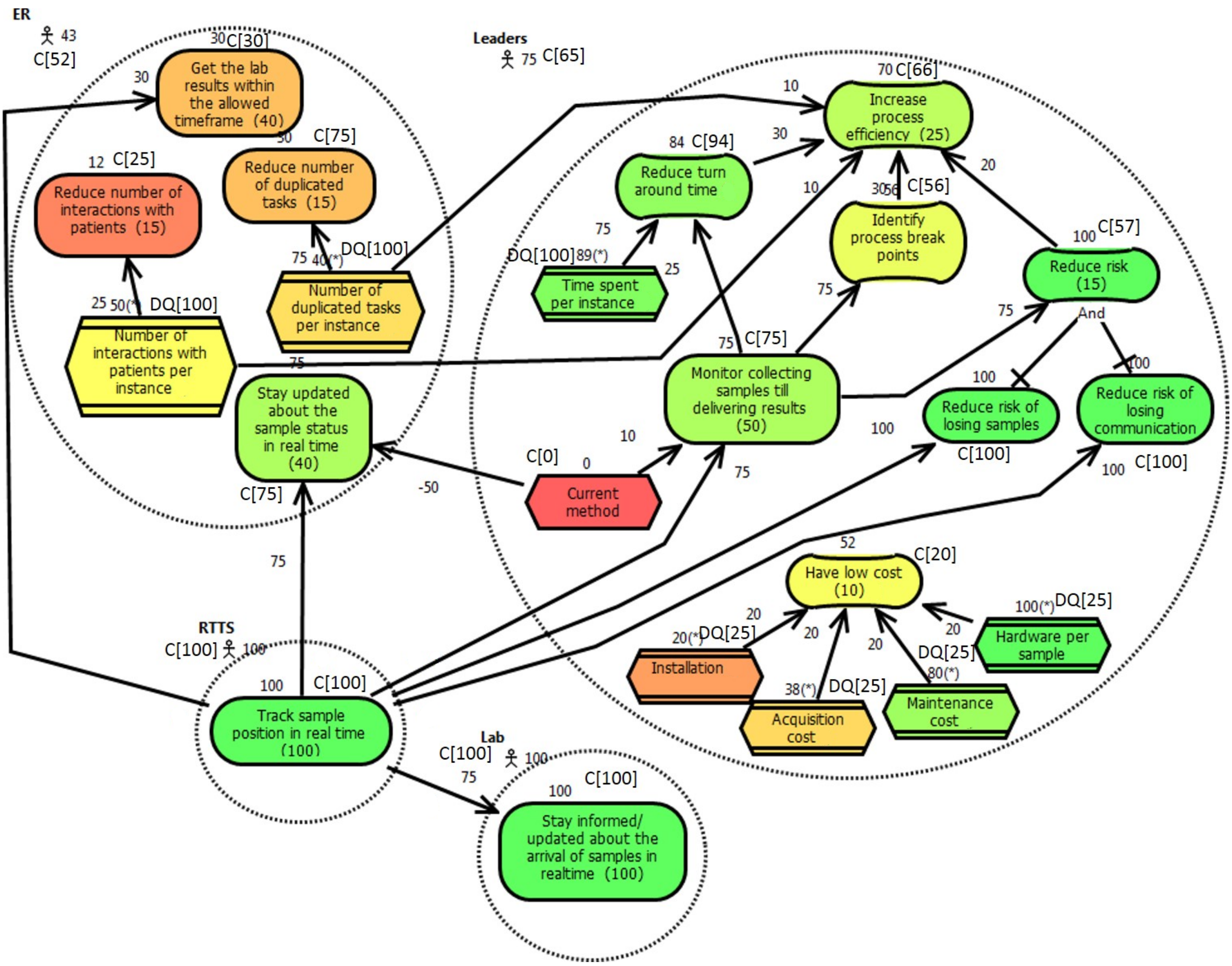

Figure 13. GRL model evaluation corresponding to the *RTTS by nurse* alternative in a critical situation.

6.2. Confidence Propagation Interpretation

The results of the nine GRL evaluation strategies were presented and discussed with the hospital representatives. Generally, the *current method* alternative is better than the RTTS alternative mainly because of the cost. The hospital receives around 98,550 urgent cases and 558,450 routine cases yearly, which increases the cost of using an RTTS system dramatically. For critical situations, where they receive 49,275 cases yearly, and the cost seems to be affordable. The confidence in "Have a low cost" is very low (25) meaning that there is no obvious clue whether the actual cost would be more or less than the estimated cost used in the evaluation. As a result, the hospital investigated the cost of similar existing systems that were deployed in other contexts of other hospitals where the cost was higher than the estimated cost of RTTS, which is not affordable giving their current budget. At this point, it is essential for the hospital to monitor and evaluate the performance in the critical cases; hence the RTTS is a good option for them, and this is supported by the analysis of the goal model for that context. However, it is still expensive to have the system for the other two cases (urgent and routine). The confidence value of the cost is estimates (20), which diminishes the soundness of the satisfaction value of the cost and highlights the lack of reliability of the collected data.

The evaluation of the goal model is not only about the cost; more importantly, it is about finding a solution to monitor the process of collecting samples in real time until results are delivered. The confidence in the satisfaction value of this goal, which is 75 using the RTTS, is quite high. In addition, there is high confidence in the satisfaction of other goals such as to increase process efficiency and reduce turn around time. In terms of the performance goals, the confidence levels fall between 75 and 100 when using an RTTS.

Another revealing aspect of the case study is that the hospital needs a real-time tracking system to track lab samples and other things; however, because the RTTS system was developed by a single programmer, not an official vendor, the hospital would not pay much for it at the moment. In addition, for the same reason, the *RTTS by Nurse* alternative was more welcomed than the *Automated RTTS* because the hospital would not risk typical security and privacy issues when exposing a wide range of patients medical records to the RTTS; nevertheless, the ER actor was less satisfied in the *RTTS by Nurse* alternative compared to the *Automated RTTS* alternative.

The results of the evaluation of the alternatives for urgent and routine situations are similar to those for critical ones. The main difference rests in the cost involved, as discussed above. The hospital representative delivered the analysis results to administration leaders, supporting the suitability of using the RTTS as a temporary solution to fulfill identified needs for critical situations, while remaining affordable. Moreover, they will keep the current method for other situations (with higher loads) until they find a trusted vendor. The propagated confidence supported the idea of having a tracking system in order to satisfy almost all stakeholder goals in the short term and fulfill their immediate needs. However, for the long term, the confidence levels also show that some goals related to costs and to reducing the number of interactions with patients need to be fully addressed with more reliable data/evidence before adopting any solution.

7. Discussion

Although the decision-making process of the hospital from the case study is still in progress, the case study provided an instance where the data quality approach presented here adds a piece of information to the goal model that can estimate confidence in a simple and practical way, while identifying locations where data quality needs to be higher. Many studies have been proposed to support decision making in the presence of uncertainty. However, they are often not practical in industry. For example, in the healthcare sector, clinical data about patients, diagnoses, and treatments, in most cases, are available. In this context, advanced methods, such as those based on probability distributions, can be used. However, management data about performance or costs (beyond acquisition costs) related to a new technology or system design are rarely available [32]. The challenge increases when data is unavailable about the current technological solution being used in a hospital. In this case, the mechanism proposed in this paper would be beneficial to highlight data insufficiency and the need to collect more evidence. As we are interested in healthcare requirements engineering research [32], the approach has been discussed with healthcare IT workers with encouraging feedback. In addition to the potential benefits in practice (decision-making support, and coverage of goal contributions), this data quality approach could likely be applied to other goal modeling languages beyond GRL (e.g., *i**, KAOS, or the Goal Structuring Notation).

There are also several limitations to the method and threats to the validity of our work that deserve attention:

1. One major limitation is that data quality types (Section 4.1) could be improved based on other dimensions. The quality types presented in this paper are generic and function well, at least in the context of this paper and its case study. However, more precise types may be needed in other contexts. For example, the classification of data gathered through sensors in real time is not yet supported in the proposed approach.
2. It is also worth mentioning that assigning quality types to data is not trivial. Currently, this is done based on an assessment of analysts and stakeholders involved in the context where, probably, many disagreements and conflicts arise. Therefore, it is important to systematize and more formally describe the process of assigning quality types to data.

3. The functions used to propagate confidence levels (Section 4.2), although they did not generate complaints from the case study participants, are currently only justified through arguments. They could be further validated empirically, especially against alternative propagation functions.
4. The illustrative example may not reflect the complexity of other real-world cases and contexts, especially outside the healthcare domain. Additional cases studies in other domains would help raise our confidence in the suitability and generalizability of the mechanism proposed here. More formal experiments could also provide more reliable empirical evidence, for instance by comparing the outputs of the uncertainty reasoning proposal with those of domain experts, or by studying the scalability and usability of the reasoning as goal models get larger.
5. The approach was currently implemented for GRL models. Although we do not see major issues in porting it to other goal-oriented modeling languages, whether specific semantics of these languages will require major adaptations to some of the confidence propagation functions remains a research topic.
6. As the creators of the Data Quality Tagging and Propagation Mechanism also led the development and analysis of the case study, real or perceived biases represent another potential threat to the internal validity of our work. One potential mitigation would be to have people other than us lead experiments and case studies about the usefulness of the approach.

8. Conclusions

Data availability has always been a big challenge in industry. Yet, reasoning about system goals and alternatives in the presence of uncertainty related to data quality is important. In this paper, we propose a Data Quality Tagging and Confidence Propagation Mechanism that maps data quality to initial confidence levels and propagates this information to compute the confidence in the satisfaction values of other elements of a goal model. The approach, implemented in a modeling tool, improves on related work thanks to its simplicity and broader applicability, as illustrated in the example and informally reported by IT workers who perceived value in this approach. Not only does it provide confidence levels for all intentional elements of a (GRL) goal model, which can influence decision making, but it also helps identify locations where higher-quality data would help increase confidence in the goal-oriented analysis.

This work can also enable further research on the impact of weak/strong confidence of high/low satisfaction values on decision making. In particular, high-quality data may not be needed everywhere in a model to support proper decision-making; for example, the structure of a goal model could compensate weak confidence from one source with strong confidence from another source. In other words, there might be some built-in tolerance to some level of uncertainty in different goal model structures. On the other hand, the lack of sufficient confidence in a conclusion might require decision-makers to seek additional evidence or better sources of data to increase the resulting confidence above a certain threshold to actually make an informed decision.

For future work, different and more rigorous data quality dimensions, as well as the generalization of our mechanism to languages other than GRL, could both be explored. Different functions for aggregating confidence levels (e.g., average, maximum, etc.) based on GRL link types could also be more formally and empirically evaluated; their suitability might vary along different contexts, and we could even enable analysts to select them explicitly for a given model. Further industrial validation is also required, especially outside the healthcare sector, and ideally done by people other than the developers of the approach. Furthermore, we see an opportunity to combine our approach with other existing methods presented in related work to provide more concrete evidence to support decisions by considering the source, volume, and quality of data being used.

Author Contributions: Conceptualization, M.B. and D.A.; methodology, M.B., D.A. and J.M.; software, M.B.; validation, M.B.; formal analysis, M.B. and D.A.; investigation, M.B., D.A. and J.M.; resources, M.B. and D.A.; data curation, M.B.; writing—original draft preparation, M.B.; writing—review and editing, D.A. and J.M.; visualization, M.B.; supervision, D.A.; project administration, D.A.; funding acquisition, M.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Ministry of Education, Kingdom of Saudi Arabia (Ph.D. scholarship).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors thank Yasser Al-Shallahi and managers at Al-Rass Hospital for their collaboration in the case study. We also thank the anonymous reviewers for their useful suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AHP	Analytical Hierarchy Process
BBN	Bayesian Belief Networks
ER	Emergency Room
GA	Genetic Algorithm
GRL	Goal-oriented Requirement Language
KPI	Key Performance Indicator
MHR	Medical Health Record
RTTS	Real-Time Tracking Sample system
SAR	Saudi Riyals
URN	User Requirements Notation

References

1. Letier, E.; Stefan, D.; Barr, E.T. Uncertainty, Risk, and Information Value in Software Requirements and Architecture. In Proceedings of the 36th International Conference on Software Engineering (ICSE), Hyderabad, India, 31 May–7 June 2014; ACM: New York, NY, USA, 2014; pp. 883–894. [CrossRef]
2. Frank, U. Domain-Specific Modeling Languages: Requirements Analysis and Design Guidelines. In *Domain Engineering: Product Lines, Languages, and Conceptual Models*; Reinhartz-Berger, I., Sturm, A., Clark, T., Cohen, S., Bettin, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2013; pp. 133–157. [CrossRef]
3. Mussbacher, G.; Amyot, D. Goal and scenario modeling, analysis, and transformation with jUCMNav. In Proceedings of the 31st International Conference on Software Engineering—Companion Volume, Vancouver, BC, Canada, 16–24 May 2009; pp. 431–432. [CrossRef]
4. ITU-T. Recommendation Z.151 (10/18) User Requirements Notation (URN)—Language definition. 2018. Available online: https://www.itu.int/rec/T-REC-Z.151/en (accessed on 2 June 2022).
5. Fan, Y.; Anda, A.A.; Amyot, D. An Arithmetic Semantics for GRL Goal Models with Function Generation. In Proceedings of the International Conference on System Analysis and Modeling, Copenhagen, Denmark, 15–16 October 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 144–162. [CrossRef]
6. Busari, S.A. Modelling and Analysing Software Requirements and Architecture Decisions under Uncertainty. Ph.D. Thesis, UCL (University College London), London, UK, 2019. Available online: https://discovery.ucl.ac.uk/id/eprint/10067421 (accessed on 2 June 2022).
7. Busari, S.A.; Letier, E. Radar: A lightweight tool for requirements and architecture decision analysis. In Proceedings of the 2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE), Buenos Aires, Argentina, 20–28 May 2017; pp. 552–562. [CrossRef]
8. Cailliau, A.; van Lamsweerde, A. Handling knowledge uncertainty in risk-based requirements engineering. In Proceedings of the IEEE 23rd International Requirements Engineering Conference (RE), Ottawa, ON, Canada, 24–28 August 2015; pp. 106–115. [CrossRef]

9. Sabetzadeh, M.; Falessi, D.; Briand, L.; Alesio, S.D.; McGeorge, D.; Åhjem, V.; Borg, J. Combining Goal Models, Expert Elicitation, and Probabilistic Simulation for Qualification of New Technology. In Proceedings of the IEEE 13th International Symposium on High-Assurance Systems Engineering, Boca Raton, FL, USA, 10–12 November 2011; pp. 63–72. [CrossRef]
10. Chen, B.; Peng, X.; Yu, Y.; Zhao, W. Uncertainty handling in goal-driven self-optimization–limiting the negative effect on adaptation. *J. Syst. Softw.* **2014**, *90*, 114–127. [CrossRef]
11. Liaskos, S.; Khan, S.M.; Mylopoulos, J. Modeling and reasoning about uncertainty in goal models: A decision-theoretic approach. *Softw. Syst. Model.* **2022**, 1–24. [CrossRef]
12. Salay, R.; Famelis, M.; Chechik, M. Language independent refinement using partial modeling. In Proceedings of the 15th International Conference on Fundamental Approaches to Software Engineering, Tallinn, Estonia, 24 March–1 April 2012; Springer: Berlin/Heidelberg, Germany, 2012; pp. 224–239. [CrossRef]
13. Bowers, K.M.; Fredericks, E.M.; Cheng, B.H. Automated Optimization of Weighted Non-functional Objectives in Self-adaptive Systems. In Proceedings of the 10th International Symposium on Search Based Software Engineering, Montpellier, France, 8–9 September 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 182–197. [CrossRef]
14. Fredericks, E.M.; DeVries, B.; Cheng, B.H. AutoRELAX: Automatically RELAXing a goal model to address uncertainty. *Empir. Softw. Eng.* **2014**, *19*, 1466–1501. [CrossRef]
15. Zawawy, H.; Mankovskii, S.; Kontogiannis, K.; Mylopoulos, J. Mining software logs for goal-driven root cause analysis. In *The Art and Science of Analyzing Software Data*; Elsevier: Amsterdam, The Netherlands, 2015; pp. 519–554. [CrossRef]
16. Dąbrowski, J. Towards an adaptive framework for goal-oriented strategic decision-making. In Proceedings of the 2017 IEEE 25th International Requirements Engineering Conference (RE), Lisbon, Portugal, 4–8 September 2017; pp. 538–543. [CrossRef]
17. Pasquale, L.; Spoletini, P.; Salehie, M.; Cavallaro, L.; Nuseibeh, B. Automating trade-off analysis of security requirements. *Requir. Eng.* **2016**, *21*, 481–504. [CrossRef]
18. Hobbs, C.; Lloyd, M. The Application of Bayesian Belief Networks to Assurance Case Preparation. In *Achieving Systems Safety, Proceedings of the Twentieth Safety-Critical Systems Symposium, Bristol, UK, 7–9 February 2012*; Dale, C., Anderson, T., Eds.; Springer: London, UK, 2012; pp. 159–176. [CrossRef]
19. Guiochet, J.; Do Hoang, Q.A.; Kaaniche, M. A Model for Safety Case Confidence Assessment. In Proceedings of the 34th International Conference, Computer Safety, Reliability, and Security, Delft, The Netherlands, 23–25 September 2015; Springer: Cham, Switzerland, 2015; pp. 313–327. [CrossRef]
20. Hall, R.J. Forensic System Verification. In Proceedings of the 17th IEEE International Requirements Engineering Conference, Atlanta, GA, USA, 31 August–4 September 2009; pp. 111–120. [CrossRef]
21. Dhaouadi, M.; Spencer, K.; Varnum, M.H.; Grubb, A.M.; Famelis, M. Towards a generic method for articulating design uncertainty. *J. Object Technol.* **2021**, *12*, 3. [CrossRef]
22. Bencomo, N.; Götz, S.; Song, H. Models@run.time: A guided tour of the state of the art and research challenges. *Softw. Syst. Model.* **2019**, *18*, 3049–3082. [CrossRef]
23. Alwidian, S.; Dhaouadi, M.; Famelis, M. A Vision towards a Conceptual Basis for the Systematic Treatment of Uncertainty in Goal Modelling. In Proceedings of the 12th System Analysis and Modelling Conference, Montreal, QC, Canada, 19–20 October 2020; Association for Computing Machinery: New York, NY, USA, 2020; pp. 139–142. [CrossRef]
24. Velasquez, M.; Hester, P.T. An analysis of multi-criteria decision making methods. *Int. J. Oper. Res.* **2013**, *10*, 56–66. Available online: http://www.orstw.org.tw/ijor/vol10no2/ijor_vol10_no2_p56_p66.pdf (accessed on 2 June 2022).
25. Lilja, K.K.; Laakso, K.; Palomäki, J. Using the Delphi method. In Proceedings of the 2011 Proceedings of PICMET'11: Technology Management in the Energy Smart World (PICMET), Portland, OR, USA, 31 July–4 August 2011; pp. 1–10. Available online: https://ieeexplore.ieee.org/document/6017716 (accessed on 2 June 2022).
26. Akhigbe, O.; Alhaj, M.; Amyot, D.; Badreddin, O.; Braun, E.; Cartwright, N.; Richards, G.; Mussbacher, G. Creating quantitative goal models: Governmental experience. In Proceedings of the International Conference on Conceptual Modeling, Atlanta, GA, USA, 27–29 October 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 466–473. [CrossRef]
27. Liaskos, S.; Jalman, R.; Aranda, J. On eliciting contribution measures in goal models. In Proceedings of the 2012 20th IEEE International Requirements Engineering Conference (RE), Chicago, IL, USA, 24–28 September 2012; pp. 221–230. [CrossRef]
28. Vinay, S.; Aithal, S.; Sudhakara, G. Effect of Contribution Links on Choosing Hard Goals in GORE Using AHP and TOPSIS. In *Emerging Research in Electronics, Computer Science and Technology: Proceedings of International Conference, ICERECT 2012*; Springer: New Delhi, India, 2014; pp. 737–745. [CrossRef]
29. Kassab, M. An integrated approach of AHP and NFRs framework. In Proceedings of the IEEE 7th International Conference on Research Challenges in Information Science (RCIS), Paris, France, 29–31 May 2013; pp. 1–8. [CrossRef]
30. Zhao, T.; Zhao, H.; Zhang, W.; Jin, Z. Goal model driven alternative selection: A quantitative approach. In Proceedings of the 2015 IEEE International Model-Driven Requirements Engineering Workshop (MoDRE), Ottawa, ON, Canada, 24 August 2015; pp. 1–10. [CrossRef]
31. Sumesh, S.; Krishna, A. Hybrid analytic hierarchy process-based quantitative satisfaction propagation in goal-oriented requirements engineering through sensitivity analysis. *Multiagent Grid Syst.* **2020**, *16*, 433–462. [CrossRef]
32. Baslyman, M.; Almoaber, B.; Amyot, D.; Bouattane, E.M. Activity-based Process Integration in Healthcare with the User Requirements Notation. In Proceedings of the 7th International Conference on E-Technologies, Ottawa, ON, Canada, 17–19 May 2017; Springer: Berlin/Heidelberg, Germany, 2017; pp. 151–169. [CrossRef]

33. Gonçalves, E.; Araujo, J.; Castro, J. PRISE: A process to support iStar extensions. *J. Syst. Softw.* **2020**, *168*, 110649. [CrossRef]
34. Singh, H.; Khalajzadeh, H.; Paktinat, S.; Graetsch, U.M.; Grundy, J. Modelling human-centric aspects of end-users with iStar. *J. Comput. Lang.* **2022**, *68*, 101091. [CrossRef]
35. Runeson, P.; Höst, M. Guidelines for conducting and reporting case study research in software engineering. *Empir. Softw. Eng.* **2009**, *14*, 131–164. [CrossRef]