Introduction

This study focuses on the use of domain modeling designed using UML class diagrams to check if requirements written in Natural Language following the “shall” style convention are deemed complete. To do this, the authors analyses the sensitivity (which is a measure that verifies if a domain model contains information that can help discover omission in the requirements) of domain models to omissions in requirements. This research is done in an industrial application and the requirements being used had already been finalised by the various industries hence deemed complete. Three case studies from 3 different industry domains were used which also involved domain experts from the different industries to work on the domain models with at least one of them being involved in the requirement creation. To evaluate the detection capability of domain models, requirements set where one requirement is missing and requirements where part of a requirement is missing were used to study the traceability links between requirements and domain models. To validate their findings, the authors clarified how sensitive domain models detect omission in requirements, if they can predict how clearly gaps will appear (sensitivity) based only on how the requirements document is written, without first creating a domain model or having a "perfect" reference.

Methodology

For the three case studies used in this research, 35 requirements are randomly selected from each case study and for each requirement atomic noun phrases are captured by the authors and given to the domain experts for review to select suitable domain concepts that they used to create the domain models with the help of existing domain-model sketches. The domain experts also add additional domain concepts they deemed necessary to properly explain the concepts they select from the noun phrases. To simulate realistic requirement omission (complete requirement omission and partial requirement omission), semantic interdependency between segments of the requirement are taken into account so the incomplete requirement still feels realistic after pruning. In all the case studies, for partial requirement omission, the segments of the requirement concerned were objects, conditions and constraints. To simulate the omissions in requirement, the algorithm randomly selects progressively larger sets of requirement elements (e.g., constraints or entire requirements) for omission. However, when constraints or objects are omitted, interdependencies between segments must be considered. Random selection may cause violations of these interdependencies. To resolve such violations, the algorithm applies minor adjustments. Given the simplicity of the interdependencies in the case studies, these adjustments can be performed straightforwardly without the need for complex solving techniques like search-based methods or backtracking algorithms.

Results

The results of the study show that domain models are generally sensitive to missing or incomplete requirements, but the level of sensitivity varies between different case studies. When entire requirements are omitted, a significant portion of the domain model elements lose support, with the relationship appearing almost linear in most cases. In contrast, the omission of individual segments such as conditions, constraints, or objects has a noticeably smaller impact. This suggests that complete requirements carry more essential information for supporting the domain model than partial segments. The study also finds that the frequency with which domain concepts appear in the requirements influences sensitivity, requirements where concepts appear more often tend to be less sensitive to omissions. To address real-world settings where the domain model is unavailable, the authors propose using unsupported keyphrases (noun and verb phrases) as a surrogate for predicting sensitivity. They observe a strong correlation between unsupported keyphrases and domain model elements, making this a practical approach for identifying incomplete requirements. Finally, by comparing subsets of requirements with full documents, the study confirms that the samples used are representative and that similar sensitivity patterns hold across the full requirement sets.

Critics and Synthesis

The study presents a rigorous and well-structured approach to evaluating requirement completeness using domain models, offering both theoretical insights and practical strategies. One of its strengths lies in the use of Monte Carlo simulations to model the impact of omissions, which provides a scalable way to explore sensitivity patterns. Additionally, the use of keyphrases as a surrogate for domain model support is both innovative and practical, especially in scenarios where full domain models are unavailable. However, a notable limitation is the reliance on manually coded omittable segments and interdependencies, which may introduce subjectivity despite the high interrater agreement reported. Furthermore, while the proposed method generalizes well across the presented case studies, its applicability to larger or more complex industrial-scale systems remains to be tested. Overall, the work contributes meaningfully to the broader conversation on requirement verification by offering a systematic, model-driven lens to detect incompleteness, and opens avenues for future research in automation and scalability.