

6. Related Work

UML is a diagram language which enables designers of information systems to illustrate high level system requirements, using use case diagrams, and to demonstrate low level system requirements, using activity diagrams [6]. Choi and Lee [7] proposed a model-driven approach that uses UML use case diagrams to elicit the requirement of context-aware systems. In particular, the approach helps analysts and stakeholders pay more attentions to context related issues such as system platform, target users, intelligence, possible context-aware services and agreement with other stakeholders, and understanding contexts with decision tables and trees.

ContUML [8] is a UML-based language for model-driven development of context-aware systems. However, ContUML essentially extends the UML class diagram with special classes for CIs and context-awareness mechanisms. Our context-aware use case diagrams are more abstract than class diagrams and so more suitable for requirement elicitation and analysis. It is understood that ContUML may be used for the realization of context-aware use case diagrams during system development. Almutairi et al. [9] extended the UML use case diagram and activity diagram to capture the security requirements of context-aware system. In particular, they introduces a “requires” relationship between a use case and CIs to indicate the CIs the behaviours described by that use case depend upon. In our approach, use context diagrams are used to specify CIs and their corresponding CSs; separately from the use cases that will utilize those CIs. This separation of concerns between functional requirements and context-awareness requirements is helpful, especially when dealing with large scale or complex context-aware systems.

7. Conclusion

This paper proposed an algorithm for translating a context-aware use case diagram into a CCA process in the aim of using the CCA tools to analyse the requirements of context-aware systems. It is demonstrated how the CCA interpreter can be used to execute and validate various scenarios of a use case diagram. The pragmatics of the approach is illustrated using a real-world example of a context-aware pedestrian collision avoidance system. In future work, it will be investigated how the model-checking tool ccaSPIN can be used to analyze the requirements of context-aware systems.

8. References

[1] A. Al-Alshuhai and F. Siewe, “An extension of the use case diagram to model context-aware systems,” in *SAI Intelligent Systems Conference*, 2015.

[2] F. Siewe, H. Zedan and A. Cau, “The Calculus of Context-aware Ambients,” *Journal of Computer and System Sciences*, vol. 77, no. 4, pp. 597-620, July 2011.

[3] J. G. Holzmann, “The model checker spin,” *IEEE Transactions on Software Engineering*, vol. 23, no. 5, pp. 1-17, May 1997.

[4] G. Booch, J. Rumbaugh, and I. Jacobson, *The Unified Modeling Language User Guide*. Addison Wesley, 1999.

[5] R. Milner, *Communication and Mobile Systems: The π -Calculus*. Cambridge University Press, 1999.

[6] A. Finkelstein and A. Savigni, “A framework for requirements engineering for context-awareness services,” in *First International Workshop from Software Requirements to Architectures*, 2001.

[7] J. Choi and Y. Lee, “Use-case driven requirements analysis for context-aware systems,” in *The Future Generation Information Technology Conference*. Springer, 2012.

[8] Q. Z. Sheng and B. Benatallah, “ContextUML: A UML-based modeling language for model-driven development of context-aware web services,” in *International Conference on Mobile Business (ICMB05)*, 2005.

[9] A. Almutairi, A. Abu-Samaha, G. Bella, and F. Chen, “An enhanced use case diagram to model context aware system,” in *SAI conference*, 2013.