to improve robustness and effectiveness, developers must decide between default in-memory session state provider and cache session. This scenario also requires model re-selection from existing models of variant features.

Another typical scenario is adding new features. For example, tenants may want to perform complex analysis on survey results. Currently, the application stores survey answers in blob storage. To provide the new feature, an SQL database (from different models under Storage Type) is the best solution for applying complex queries and join query. When adding a new feature, developers must identify whether the new feature is common or specific to certain clients. If the feature is common, the core UML model will be updated. If the feature is variable, the core UML model will remain the same and a model of variant feature for this variable feature will be generated. At this point, the MATA language detects relations and dependencies of the new feature to other features. The SQL Database also needs partitioning to support multi-tenancy. Thus, the developers must select one of the different partitioning models for SQL databases. Moreover, a new interface must be implemented to view and analyze survey data.

5. CONCLUSION

In this paper, we have proposed an integrated SPL and MDE modeling approach to address design decision variability and evolution concerns in multi-tenant SaaS cloud applications. We have applied feature modeling concepts to identify variability in implementation. The MATA language has been suggested to manage variability, and to support customization and evolution. Thus, the proposed approach allows features to be modeled independently. Furthermore, conflicts in the application structure and dependencies between models are detected. However, it requires improvements to enable cloud application development and multi-tenancy.

In our future work, we plan to enhance our approach by making the MATA language applicable for multi-tenant SaaS cloud applications and by developing a model to code transformation prototype to transform composed models to source code. A case study will be carried out to illustrate and evaluate the implemented tool. Moreover, we will compare our approach with other tools to identify benefits and drawbacks.

6. REFERENCES

[1] P. Mell, et al., "The NIST Definition of Cloud Computing", National Institute of Standards and Technology, Special Publication 800-145, Bethesda, Maryland, 2011

[2] D. Betts, et al., "Developing Multi-Tenant Applications for the Cloud on Windows Azure", Microsoft Patterns and Practices, 2013

[3] J. Guo, et al., "A framework for native multi-tenancy application development and management", Proceedings of the 9th IEEE Conference on E-Commerce Technology and the 4th IEEE Conference on Enterprise Computing, E-Commerce and E-Services, pp. 551–558, 2007.

[4] M. Abu-Matar, et al., "Towards Software Product Lines Based Cloud Architectures", Proceedings of the IEEE Conference on Cloud Engineering, 2014

[5] H. Gomaa, "Designing Software Product Lines with UML: From Use Cases to Pattern-Based Software Architectures", Addison-Wesley Professional, 2004

[6] I. Sommerville, "Software Engineering", Pearson, 2010

[7] K. Lee, et al., "Concepts and guidelines of feature modeling for product line software engineering", Proceedings of the 7th Conference on Software Reuse: Methods, Techniques, and Tools, pp. 62–77, 2002.

[8] R. Mietzner, et al., "Variability modeling to support customization and deployment of multi-tenant-aware Software as a Service applications", Proceedings of the 2009 ICSE Workshop on Principles of Engineering Service Oriented Systems, pp. 18-25, 2009

[9] S. Walraven, et al., "Efficient Customization of Multi-tenant Software-as-a-Service Applications with Service Lines", Journal of Systems and Software, Vol. 91, pp. 48-62, 2014.

[10] A. Shahin, A "Variability Modeling for Customizable SaaS Applications", International Journal of Computer Science and Information Technology, 6(5), pp. 39-49, 2014.

[11] I. Kumara, et al., "Sharing with a difference: Realizing service-based SaaS applications with run-time sharing and variation in dynamic software product lines", IEEE Conference on Services Computing, pp. 567–574, 2013

[12] A. Bergmayr, et al., "The Evolution of CloudML and its Manifestations", Proceeding of the 3rd Workshop on CloudMDE, 2015

[13] A. Bergmayr, et al., "UML-Based Cloud Application Modeling with Libraries, Profiles and Templates", Proceedings of the 2nd Workshop on CloudMDE, 2014.

[14] G. S. Silva, et al., "Cloud DSL: A Language for Supporting CloudPortability by Describing Cloud Entities", Proceedings of the 2nd Workshop on CloudMDE, 2014.

[15] E. Cavalcante, et al., "Exploiting Software Product Lines to Develop Cloud Computing Applications," the 16th Software Product Line Conference, 2012.

[16] P. Jayaraman, et al., "Model Composition in Product Lines and Feature Interaction Detection Using Critical Pair Analysis", Conference on Model Driven Engineering Languages and Systems, 2007

[17] F. Mohamed, et al., "SaaS Dynamic Evolution Based on Model-Driven Software Product Lines", Proceedings of the IEEE 6th Conference on Cloud Computing Technology and Science, 2014