Asikainen, T., Mannisto, T., & Soininen, T. (2006). A unified conceptual foundation for feature modelling. Proc. 10th International Software Product Line Conference, 31-40.

Feature modeling is a powerful tool that has been around since 1990. However, it has not been adapted because of lack of conceptual foundation and lack of continuity in what concepts are included. This paper uses the domain ontology called Forfamel. This paper rigorously defines feature configuration as "a set of features (F), the subfeature relation (s), the attribute relation (a), the type function t, and the root feature (r) that is a member of the set of features. A feature is characterized by its type, subfeatures, and attributes." Type function is t:F->T, where T is the set of Types in the feature configuration. This paper also formally defines feature models. The validations of the feasibility of Forfamel was done by defining a language that "generates Java code from both lexical and syntax analysis based on grammar description" and then use a tool to translate a Forfamel model into this language. In future work, Forfamel could be applied to staged configuration in feature models also.

Chen; Wei Zhang; Haiyan Zhao; Hong Mei, "An approach to constructing feature models based on requirements clustering," Requirements Engineering, 2005. Proceedings. 13th IEEE International Conference on , vol., no., pp. 31-40, 29 Aug.-2 Sept. 2005

This article discusses a method for creating feature models through requirement clustering. I think we are looking for a way to use feature models tied together with rationale to create UML and then code. The reason I include this article is because it does include the idea of creating feature models based off of requirements. This reminded me of connecting rationale to code with Dr. Burge's work.

Czarnecki, K; Antkiewicz, M. (2005). Mapping features to models: A template approach based on superimposed variants. In proceedings of International Conference Generative Programming and component Engineering (GPCE '05), vol. 3676 of LNCS, 422-437, Springer

This article discusses the role presence conditions (PCs) and meta-expressions (MEs) play in the mapping of features to models. PCs indicate which features should be included in the model and which should not. They have built a tool, fmp2rsm, that is an Eclipse plug-in that works with the IBMs Rational Software Modeler and the Feature Modeling plug-in. It is able to work with PC constraints. A demonstration of the fmp2rsm is here and a demonstration of verifying model templates available online.

Czarnecki, K.; Pietroszek. (2006). Verifying feature-based model templates against well-formedness OCL constraints. In Proceedings of International Conferences on Generative Programming and Component Engineering (GPCE'06), ACM Press

Well-formedness constraints can be expressed in the Object-Constraint Language (OCL). "The semantics maps OCL constraints to propositional formulas, which are then fed into a SAT solver".

Czarnecki, K., Hwan, C., Kim, P., & Kalleberg, K. T. (2006). Feature models are views on ontologies. Proc. 10th International Software Product Line Conference, 41-51.

Ontology modeling is another domain modeling technique. The similarities, differences, and connections of feature modeling and ontology modeling are explored in this paper. The feature model is based on a hierarchical structure of features that can be selected. An ontology as defined by Gruber, is "an explicit specification of conceptualizations". I found this paper challenging. I was unable to follow the ideas throughout.

K. Czarnecki, S. Helsen, and U. Eisenecker. Staged Configuration Through Specialization and Multi-Level Configuration of Feature Models. Software Process Improvement and Practice, special issue on "Software Variability: Process and Management, 10(2), 2005, pp. 143 – 169

This paper adds cardinality and cardinality notation to feature models. It also addresses the ideas of staged configuration and multi-level configuration. This is the paper that I presented.

Czarnecki, K., & Wasowski, A. (2007). Feature diagrams and logics: There and back again. Proc. 11th International Software Product Line Conference SPLC 2007, 23-34.

This paper shows how to represent feature models as mathematical formulas. This paper focuses on how to create feature models from these mathematical formulas. One difficulty is that multiple feature models can be created from one formula and the multitude of these models can be too much information for the user. An algorithm is given in this paper that takes the traits of the feature model and forms a graph. Using the algorithm and GraphViz, a tool was created that can create accurate visualizations of the feature model. Future applications include "reverse engineering of feature models from formulas that were not obtained from other feature models, but, for example, reverse engineered from code."

Czarnecki, K. FeaturePlugin: Feature modeling plug-in for eclipse(2004).

This paper explains how the feature model plugin works. It also gives a short explanation of feature models. The examples in this paper are helpful in getting started with the tool, but it does not get into much theory about feature models or generative software development. I used some of the models for playing around with the plugin.

K. Czarnecki. Overview of Generative Software Development. In J.-P. Banâtre et al. (Eds.): Unconventional Programming Paradigms (UPP) 2004, Mont Saint-Michel, France, , LNCS 3566, pp. 313–328, 2005.

This shows where feature modeling fits into the framework of generative software development. It argues the importance of model driven development, and especially the need for different stages of model development. The paper shows that feature modeling fits into the mapping from the problem space to the solution space mostly by clearly defining the solution space and getting started with the mapping. Feature modeling does not continue to be useful all the way into the mapping of the solution space.

Gannod, G. C., & Timm, John T. E. (2004). An MDA-based approach for facilitating adoption of semantic web service technology. Paper presented at the Proceedings of the 8th IEEE EDOC Enterprise Computing Conference Workshop on Model-Driven Semantic Web, Monterey, California.

This article describes such concepts as the following: web services, semantics, ontologies. It also gives method for creating UML models with stereotypes that can be mapped to semantic web services.

Gannod, G. C., Timm, John T. E., & Brodie, R., J. (2006). Facilitating the specification of semantic web services using model-driven development. International Journal of Web Services Research, 3(3), 61-81.

This article helps to explain ontology, semantics, and a higher level modeling approach to web-services.

Sun, J., Zhang, H., Fang, Y., & Wang, L. H. (2005). Formal semantics and verification for feature modeling. Proc. 10th IEEE International Conference on Engineering of Complex, 303-312.

This paper explores how to represent feature models in rigorous formal mathematical statements. I then uses another program called Z/Eves which will act as a verifier for these mathematical feature model formulas. They give the Key Word in Context example and transform it into the mathematical feature model statement. Then they apply the theorem prover program Z/Eves and show that it works. They also analyzed the semantics with Alloy which is a first-order logic analyzer. I found this paper difficult.