CoCoME Jury Evaluation and Conclusion

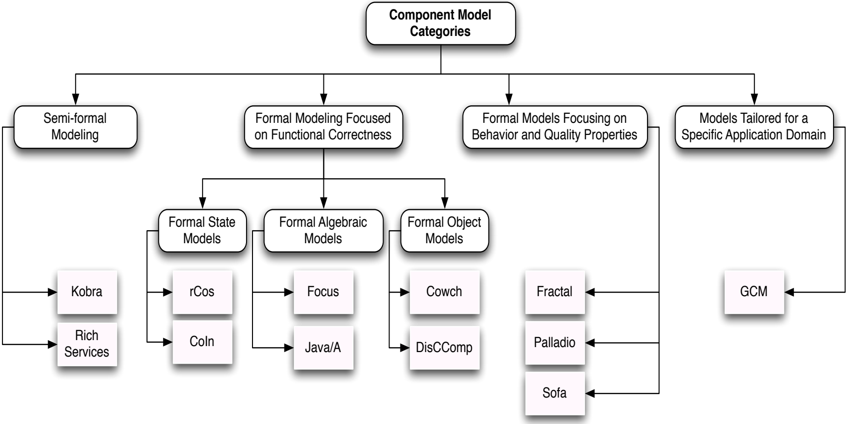
Abdel Hakim Hannousse

14th Meeting

Outline

1. **Component Models Classification**
   - Classification
   - Semi-Formal Modeling
   - Formal Modeling Focused on Functional Correctness
     - Formal State Models
     - Formal Algebraic Models
     - Formal Object
   - Formal Models Focusing on Behavior and Quality Properties
   - Models Tailored for a Specific Application Domain
2. **Conclusion**

Component Models Classification



Kobra

1. Strengths
   - easy-to-read description
   - the ability to discuss the design with expert-domains before committing to concrete implementation
2. Weaknesses
   - Lack of dedicated tool that ensure that UML is used in a Kobra compliant way
   - Non-functional properties are not considered
   - Confusion between Kobra an UML

Rich Services

1. Strengths
   - Architectural support
   - Tool verification support
   - Separate logical models from their implementation -> introduce performance optimization at deployment time
2. Weaknesses
   - hard to compare with other models, since most of them changed the structural view of original CoCoME

rCOS

1. Strengths
   - Clear and straight-forward object-oriented analysis model
   - Formal support for step-wise refinement
   - Code generation support through refinement
2. Weaknesses
   - Unclear relationship to component model
   - Unclear specification of provable properties by the model

CoIn

1. Strengths
   - Expressive power and flexibility due to the parametrized composition operator
   - Detailed model
   - Completeness
   - Model checking support
2. Weaknesses
   - None support of extra-functional properties (*)

Focus

1. Strengths
   - Ability to model reactive systems
2. Weaknesses
   - Unable to model system instantiation, it only supports fixed structures

Java/A

1. Strengths
   - Ability to derive properties of composite components throught their inner components
   - Avoid architectural erosion
2. Weaknesses
   - The relation of observational equivalence is so strong property
   - Quantitative aspects can be integrated into Java/A (*)

Cowch

1. Strengths
   - Clear semantics fundation on top of object-oriented programming languages
   - Model the complete functional and structural properties of CoCoME
   - Fill the gap between high-level architecture and desciption techniques
2. Weaknesses
   - Behavioral part is not modeled, it is only refered to Java Code
   - Strongly related to code, it is not at the higher level of abstraction
   - Does not support verification and analysis

DisCComp

1. Strengths
   - Provides a sound semantic model for concurently executed components
2. Weaknesses
   - Unclear approach for specifying CoCoME
   - None support for non-functional properties

Palladio

1. Strengths
    - Provides different architectural design alternatives supporting their design decisions with quantitative performance prediction
    - Able to model complex system interactions such as Use case 8 of CoCoME

Fractal

1. Strengths
   - Support different concepts of CBSE from architectural design to deployment -> Better way to model applications like CoCoME
2. Weaknesses
   - Does not support asynchronous communication
   - Does not support extra-functional properties

Sofa

1. Strengths
    - As Fractal with support of asynchrounous communication
    - Compared to Fractal the specification code is less than the specification of Fractal

GCM

1. Strengths
   - Seamless distrubution due to the support of virtual nodes (thanks to virtual nodes)

Conclusion

Conclusion

Each of the modeling approaches, when applied to CoCoME, exhibited an in- teresting set of strengths and weaknesses. While, pointwise, there is much to be learned from each of the approaches, the jury concludes with a mixed message. CoCoME's top-level challenge was and is about all-of-system modeling and none of the approaches enabled comprehensive modeling at that level. To be fair, the approaches at hand are the results of ongoing research and it isn't the purpose of research efforts to deliver fully rounded "complete" solutions. However, the gap between practical applicability and demonstrated modeling capability remains significant, leaving much room for further work.