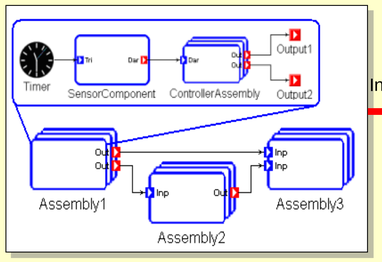
Composing Security Modeling Concepts into Embedded System Models

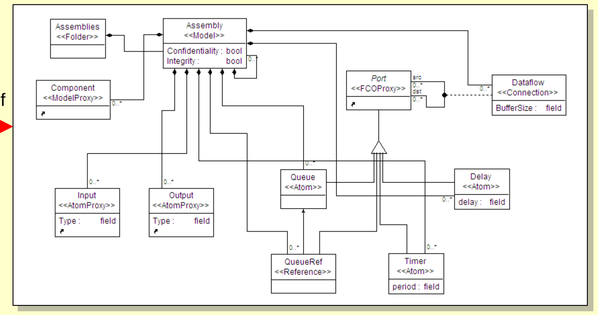
Jan Werner, Matt Eby, Janos Mathe, Gabor Karsai, Akos Ledeczi, Janos Sztipanovits
Institute for Software Integrated Systems (ISIS), Vanderbilt University

Metamodels – language definition

The metamodel defines the set of all syntactically correct models in a Domain Specific Modeling Language.



Instance of



Model of an embedded system designed using the SMOLES modeling language.

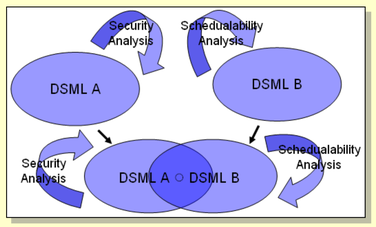
Metamodel: Part of the SMOLES modeling language metamodel (other parts include behavior and component models).

Metamodels are defined using UML and OCL constraints.

Language composition

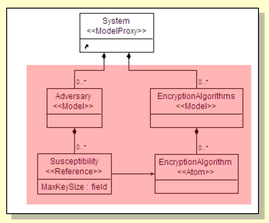
Modeling languages can be composed using composition operators of the meta-language. The resulting language retains the properties of the composed languages, and consequently models in the original languages are valid in the new resulting language.

Composition operators:
- Equivalence operator represents the equality of two classes.
- Inheritance operator represents class specialization.
- Containment operator represents class composition for building complex structures.

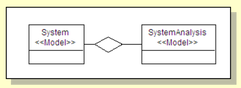


Language composition allows us to easily create modeling languages that capture multiple aspects.

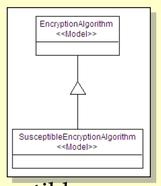
Examples of metamodel composition.



System model is extended with new concepts.

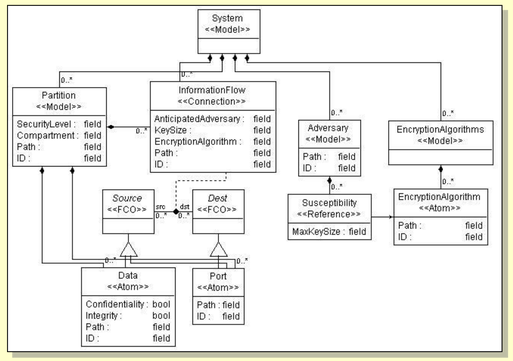


System and System Analysis represent the same class.

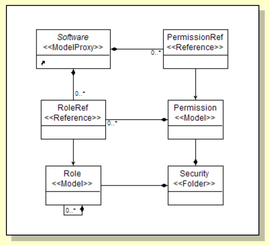


Susceptible encryption algorithm inherits properties of encryption algorithm.

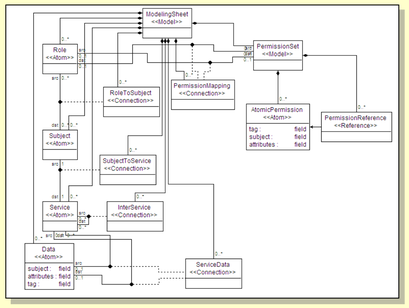
Examples of security modeling



The SAIF modeling language allows information flow analysis and vulnerability analysis, and can be composed with any language expressive enough to represent information flow between components.

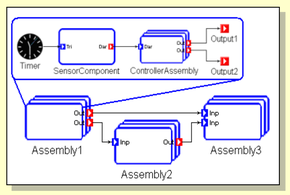


Hierarchical RBAC is used to describe fine-grained permissions in systems.

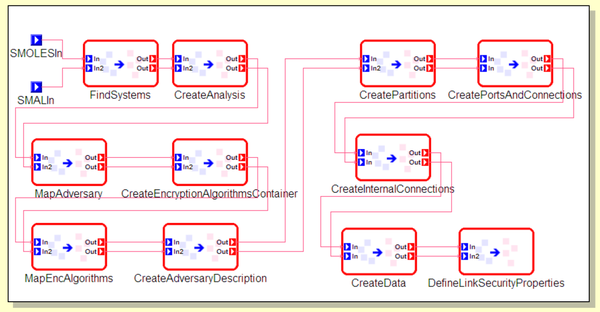


Permission mappings and actor interactions can be used to describe policies and interactions in Service Oriented Architecture.

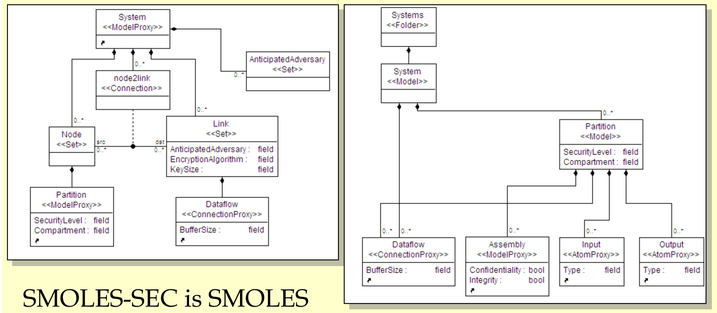
Tool integration



SMOLES is capable of capturing component structure, communication and component behavior. SMOLES does not allow the description of security properties of systems.

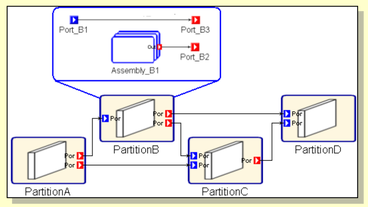


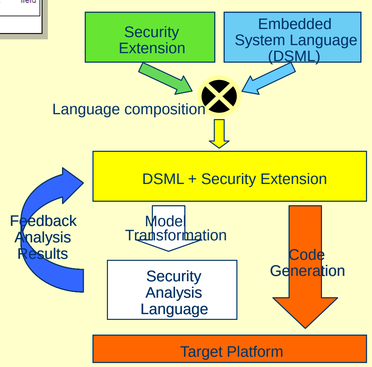
Model transformation defines the rules for modifying models or migrating between different modeling languages.
Transformation of models from SMOLES-SEC to SAIF allows system designers to verify static security properties of the system at design time.



SMOLES-SEC is SMOLES extended with concepts of partitioning, secure link properties and adversary models.

Example of embedded system model extended with the partition model and description of security properties of communication links.





Domain Specific Modeling Languages, security extensions, model transformations, code generators and model interpreters define the tool suite for designing secure embedded systems.