The contents of this presentation have been approved for public release, distribution unlimited. (DARPA's Distribution Statement A)

MDA Tool Integration for Development of Real-Time Embedded Systems

Theckla Louchios

Christopher Andrews

Vision

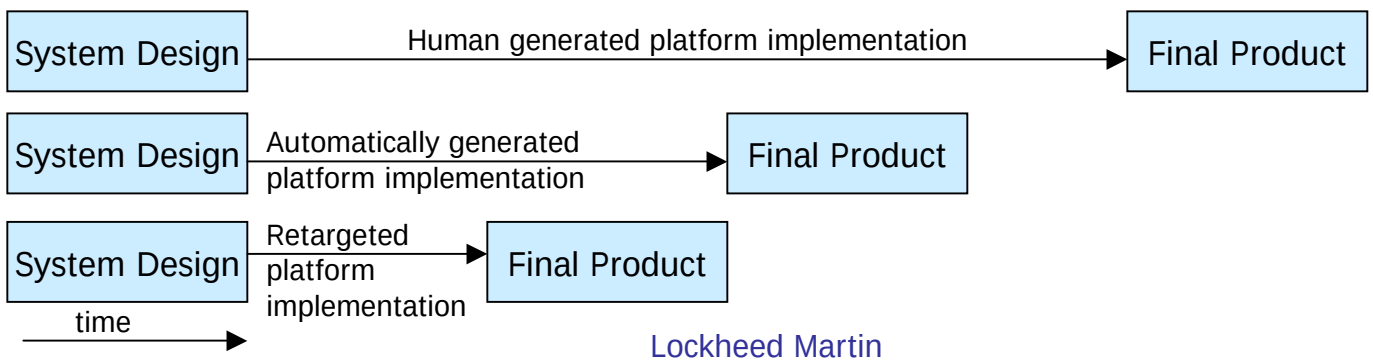
- To enable a product line architecture (PLA) leverages the strengths of current implementation technologies while interfacing with legacy technologies, and providing a level of abstraction that allows it to remain relevant with technological advancements over a 15+ year product lifetime.

CPLA Target Systems

- Component-based
  - Addressed by the way PICML and Cadena model the system
- Both asynchronous and synchronous communication
  - Tools define both of these types of interfaces between components
- Multiple versions of the entire target system each differing slightly for respective customers
  - Different platform specific models (PSM) can be targeted based on the same platform independent model (PIM)
- Different behaviors can be modeled based on the system's mode of operation
  - For example, modes could include: fully operation mode of operation, degraded mode of operation, complete system failure mode of operation
- Extended lifespan 15+ years
  - PIM is usable regardless of the targeted technology
- Legacy code
  - Ability to model connections to legacy code in PIM, and to specify actual communication mechanisms with legacy code in the targeted PSM
- Interchangeable components
  - Deployment view can to specify which components to use
  - Predefined component interfaces
- Modeling capability for both large and small scale systems
- Future capabilities
  - Requirements Tracing
    - Ability to connect system requirements modeled in a requirements tool to where they are satisfied in the system model

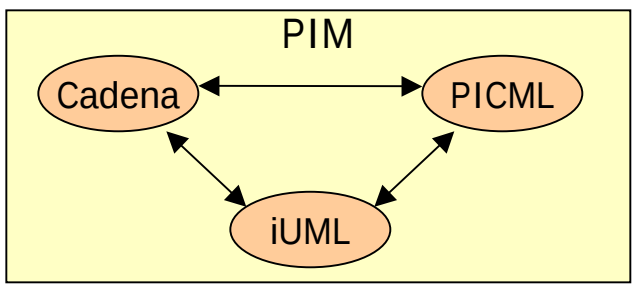
Modeling Objectives

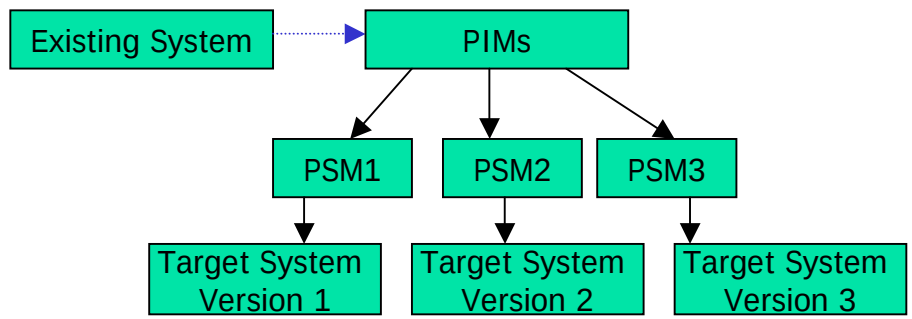
- Modeling and analysis needs to be done at the level of abstraction that is intuitive to the designer of the system.
  - Business logic model
  - QoS model
  - Communication model
  - System deployment model
- Develop a reusable platform independent model of the system that will be long-lived.
- Expedite development process of final system, ensuring all system requirements are met or exceeded
- Efficiently maintain existing systems
- To provide a baseline design in a platform independent manner for several product lines
- Diagnose shortcomings or problems early in the development process
- Automate generation of code deployment and configuration files
- Automate generation of implementation code
- Reduce design and development complexity that can be associated with sophisticated middleware



Approach

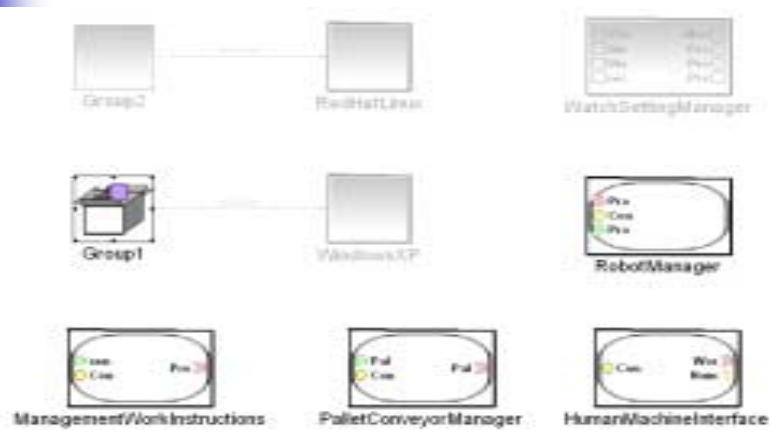
- Multiple modeling tools are needed to intuitively represent the different aspects of the high level model and to provide the ability to comprehensively analyze and test the model early on in the development process.
- Integration of existing platform independent tools to create a product line architecture
- Tool integration standards
  - Need to allow scalability for new tools to be added with ease to the development architecture
  - Effort needed to integrate tools should be minimal
  - Information should be preserved between tools

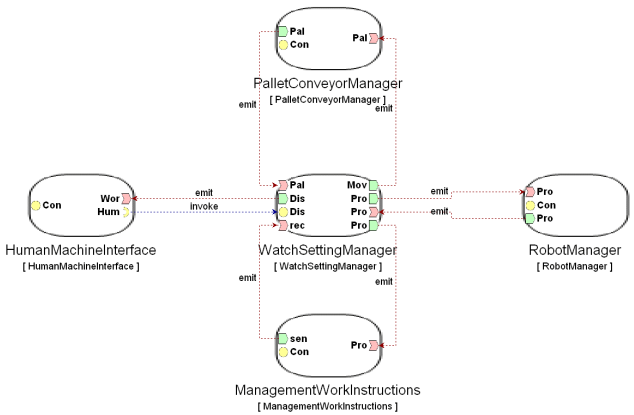




PICML

(Vanderbilt University)





- Tool suite for developing component models
- Automatic generation of deployment and configuration files based on PICML component model
- Options configuration language used to specify ORB-level configurations
- Provides means for event channel QoS specifications

Cadena

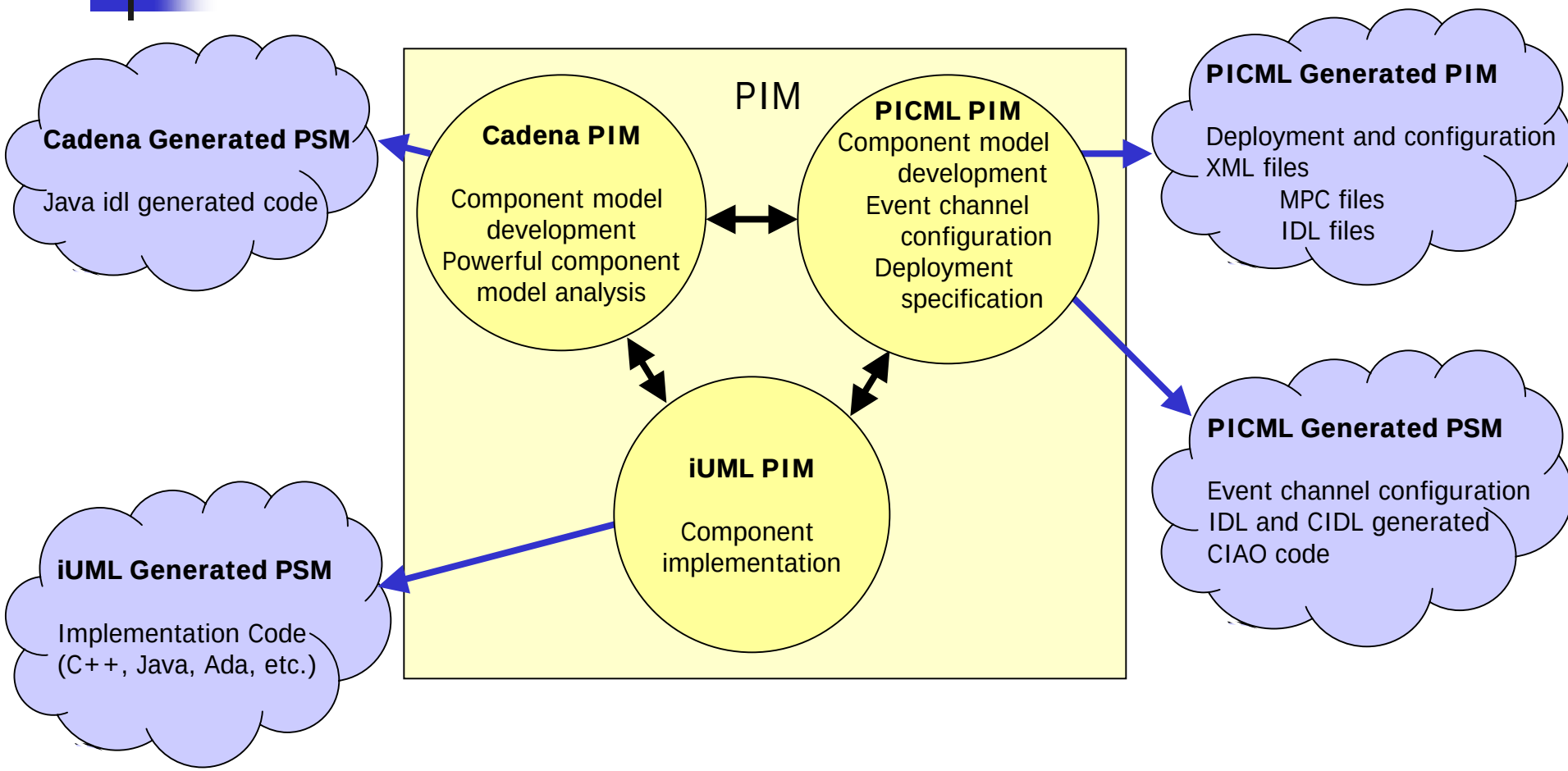
(Kansas State University)

- Component analysis capabilities
  - Cycle checks
  - Forward/Backward slicing
  - Chopping model between endpoints
  - Modal analysis
  - Timing analysis
    - Rate assignment
    - Rate conflict detection and resolution
  - Model consistency checks

MDA

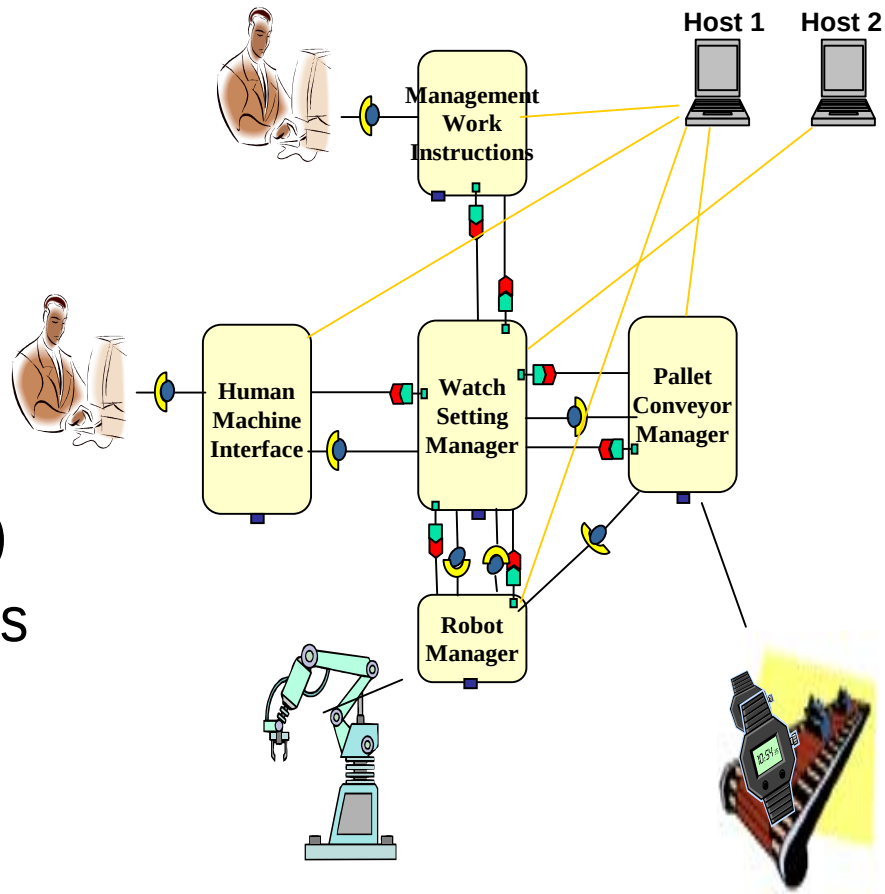
- Provides a platform independent approach to specifying the component implementation
  - Utilizes UML diagrams and a behavioral specification language (ASL) to result in a precisely defined platform independent model that can be translated into generated executable code for targeted platforms
  - Provides an integrated and central location for system requirements, analysis and design work to be done in a technology neutral manner

Tool Transformations



Tool Integration Test

- Integration with legacy code
- Asynchronous communication
- Periodic input
- Interchangeable components (dynamically)
- Distribution of components among processors
- Deadlock analysis
- Cyclic analysis



Modeling Benefits

- Separation of concerns between design and development of business logic from communication mechanism and QoS requirements.
  - Permits parallel development between fine tuning communication and developing business logic
- Ease of system maintenance
  - Modification of the PIM simplifies maintenance changes
- Prevents design artifact obsolescence
- Formal analysis can be done at the model level
  - Detects early design flaws
- System functionality and requirements are maintained throughout system life cycle in a platform independent way

Current Limitations

- Model version control for parallel development efforts
- Expansion of analysis and debug capabilities at the model level
  - Data flow analysis
  - Timing analysis
- Behavior specification language limitations
  - Lacks string manipulation
  - Lacks complex mathematical capabilities
  - Lacks common data structures such as queues, stacks, linked lists
- Ease of developing model compiler/ model interpreters
  - Model compiler/interpreter needs to guarantee correctness of generated code
  - Performance baseline of generated code
  - Humans need to be able to understand the generated code

Conclusion

- Separation of technology concerns from system functionality
  - Model can be targeted towards different technologies
- Support for model-level analysis
- Hierarchical modeling abstractions for varying levels of system detail
- Integration of autonomous legacy systems

References

- http://www.dre.vanderbilt.edu/cosmic
- http://www.cs.wustl.edu/~schmidt/CIAO.html
- http://cadena.projects.cis.ksu.edu
- http://www.isis.vanderbilt.edu/Projects/mobies
- http://www.kc.com

Extra Slide

Component Based Product Line Architecture Workflow