Timing Consistency and Prediction for IEC61131 Component-Based Designs (TCAP1131-CB)

Heinz Schmidt
Monash University

Capabilities
Benefits
Method
Problems
State and Plan
Scope

Joint work with Ian Thomas and Ralf Reussner

Monash University

www.monash.edu.au

- Australia's International University
  - Largest university in Australia, approx 50K students, 5K staff, 5K IT students, $700M annual budget, $1B assets
  - Established in 1958, now 10+ campuses:
    - **6 Melbourne**, Malaysia, South Africa
    - recently: London, Berlin, Prato
  - Famous science, medicine, engineering
  - Awarded national synchrotron
- Information Technology Faculty
  - 200 staff, >5K students (50% intl) 180 PhD + Master students
  - Recognized Research Strengths:
    distributed systems and software engineering
    uncertain reasoning
    constraint solving

- Monash Centre for Distributed Systems and Software Engineering (DSSE)
  - parallel and distributed systems
  - component software engineering
  - component mobility and mobile commerce
- DSTC National Collaborative Research Center
  - Brisbane, Melbourne, Sydney
  - 1999-2006 funding ~$50 million ~60staff
  - Aus govt, universities and intl. industry

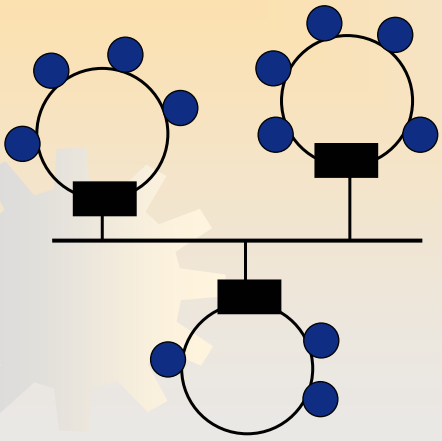
DISTRIBUTED SYSTEMS TECHNOLOGY CENTRE

SAP IBM CSIRO Microsoft Sun microsystems Mincom Telstra OzEMAIL

TCAP1131-CB Benefits

- Component behaviour modelling and documentation
  - IEC61131 behaviour abstractions packaged with components
  - Improved IIT Level-0 certification by property documentation
  - Local conformance checking – improved reuse consistency
- Compositional modelling
  - Flexible composition moderate to large-scale components
    - Not just functions
    - Ultimately multitasking and shared resources
  - Composable behaviour and property models
  - Enabling architecture-scale reuse – time savings in analysis and design
- Accurate safety and timing checks
  - Increased trust and risk reduction – failure cost reduction
  - Improved resource allocation – deployment hardware cost reduction

Distributed control systems ABB AC800M technology

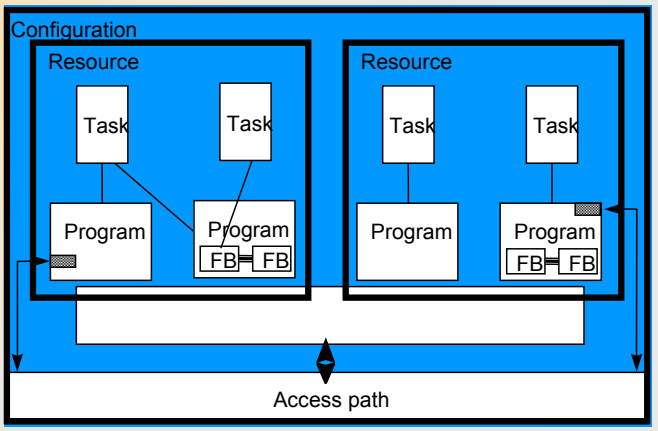


Field bus device clusters

- Few tens to hundreds meters
- Some buses with predictable delays, e.g. Profibus
- Small to moderate number of devices per controller

IEC61131-3 Standard Industrial Control Programming

- Strongly typed data;
- Software component libraries;
- Components in different programming notations;
- Standard notations in program text and diagrams:
  - Structured text (ST)
  - Instruction list (IL)
  - Function block (FB) diagrams
  - Sequential function charts (SFC)
  - Ladder diagram (LD)



TCAP1131-CB Approach

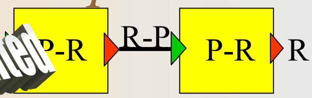
- Hierarchical behaviour models
  - Controller configuration and FB architecture
  - Behaviour abstractions packaged with components
- Dependency models
  - Finite state machine and Petri net abstractions
    - for interfaces
    - component composition
    - property dependencies
- Effective compositional property computations

Component Algebra for simple GK model based on known a... autom...

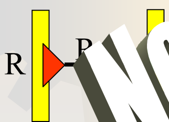
Theoretically limited

Practically promising

Novelty: property models and adapters

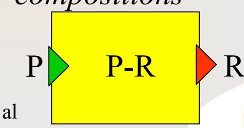


- ***Gate*** compositions
  - Interface at...
    - R = Requir...
    - P = Provided
  - Component ... adapters
  - *Regul... ...ions*



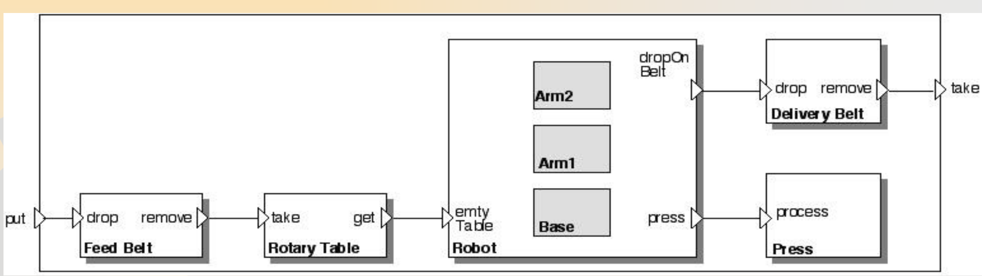
[De...ails: Schmidt et al, ...MOODS 03/02 and cited papers]

- ...*en* compositions
  - ...automata
    - Recursive transitions
    - Input, hidden, output model
    - Induced translation relation models property dependency
  - Recursive assembly
  - *Regular compositions*



[Details: Schmidt et al NATO Monterey Workshop 10/02]

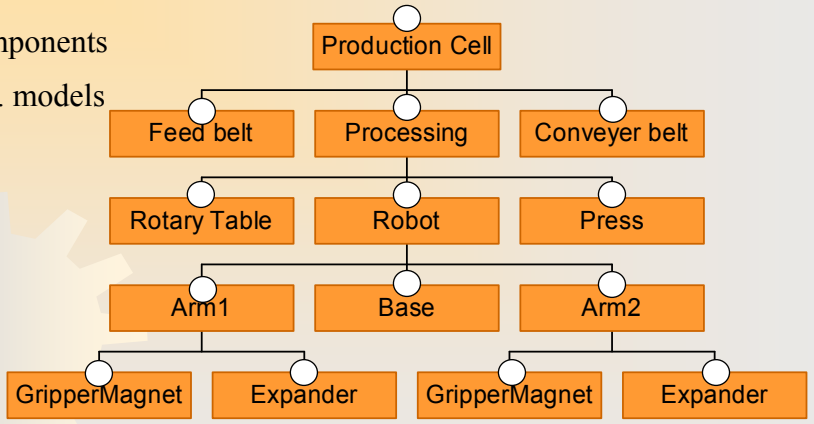
Production cell architecture and components



Hierarchical Behaviour Models

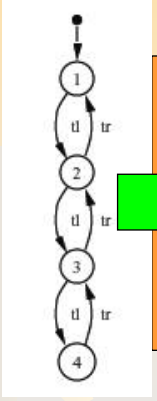
Components

Beh. models

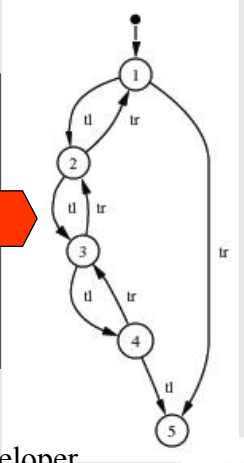


Dependency Models

RobotBaseController



parameterized dependencies



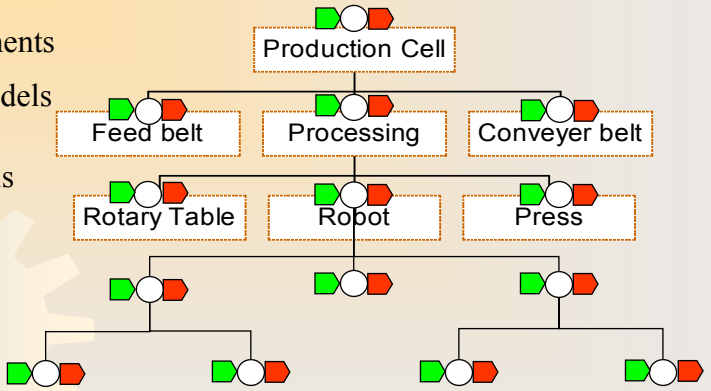
- for primitive components: provided by developer
- for composites: derived automatically by composition

Compositional Property Computation

Components

Beh. models

Dep. models



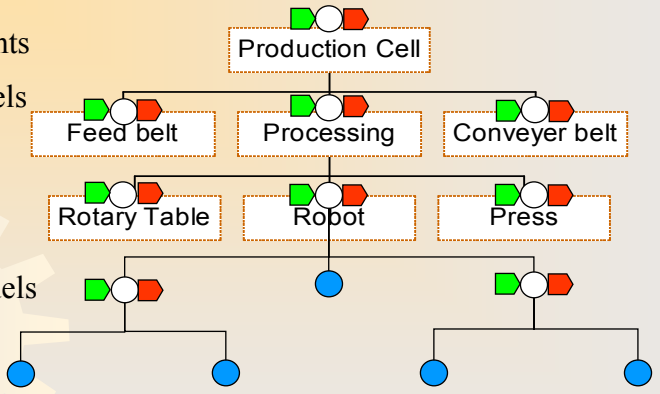
Compositional Property Computation

Components

Beh. models

Dep. models

Prop. models



property model = behaviour + dependency models

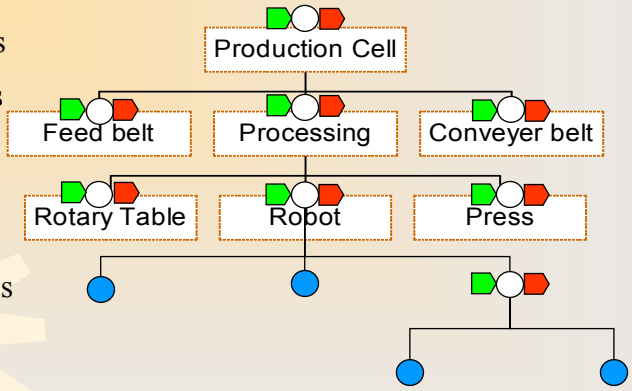
Compositional Behaviour

Components

Beh. models

Dep. models

Prop. models



property model composition follows hierarchy

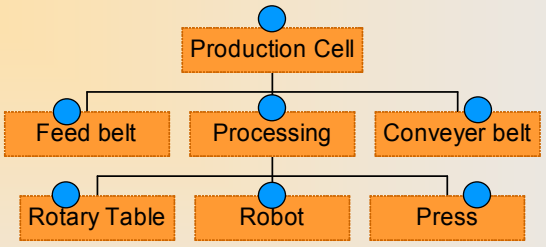
Compositional Behaviour

Components

Beh. models

Dep. models

Prop. models



Property enriched components =

Property models “canned” with packaged components but accessible for future composition

Implementation

- JAVA Initial Prototype
  - Hierarchical behaviour core classes using efficient hashing and iterators
  - O(S+T) time algorithm
  - Import/Export ASCII and XML
- Parameterization
  - Hardware specification files per property, e.g. wc-time
- Web front-end prototype (for demo and test)

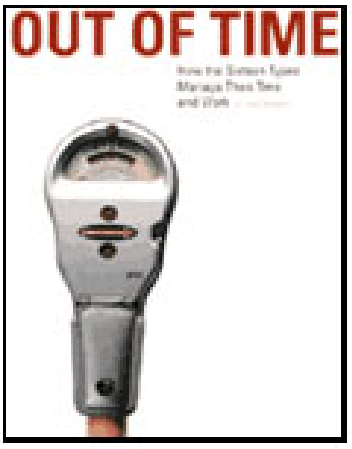
IEC61131 precision problems

- Different platforms require different performance models
  - Open: suitable hardware parameterization different chips
  - Open: OS specifics, interrupts, scheduling, comm delays
- Implementation freedom for compilers and interpreters of IEC61131 languages
  - Open: precise model of core 61131-3 language constructs

=> Joint work with platform vendor

Conclusion

- Key ideas and approach
  - Component-based time models for IEC61131
  - Hierarchical dependencies
  - Compositional packaging with components
- Initial experiments
  - Worst-case timing models
  - Proof-of-concept implementation
- Concepts promising for
  - Efficient integration in vendor tools
  - Large scale reuse (thousands) of industrial moderate scale (tasks and resources) components



?