Conceptual design of discrete-event systems using templates

Lenko Grigorov
PhD thesis
Queen's University

supervisor:
Karen Rudie

Motivation

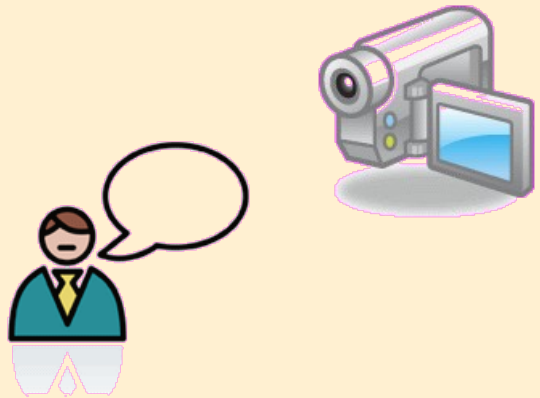
- Humans need tools to work with DESs
  - Very large state-space
  - Automatic generation of supervisors
- Need for better tools
  - Graphical environment not sufficient
    - Small mistakes may lead to completely incorrect solution
    - Not easy to modify or reuse models
    - Event synchronization is error-prone

Outline of work

- Observational study
  - How do people solve DES problems?
- Recommendations for DES software
- Template design
  - Theoretical framework
  - Tool implementation
- Evaluation of template design methodology
  - Are there positive effects in using the tool?

Observation study

- Five subjects
  - Graduate students with DES background
- Solve two problems
  - Familiar and unfamiliar (reformulated)
  - Using pen and paper and software
  - Think aloud
- Videotaped
- Protocol analysis



Some findings (1)

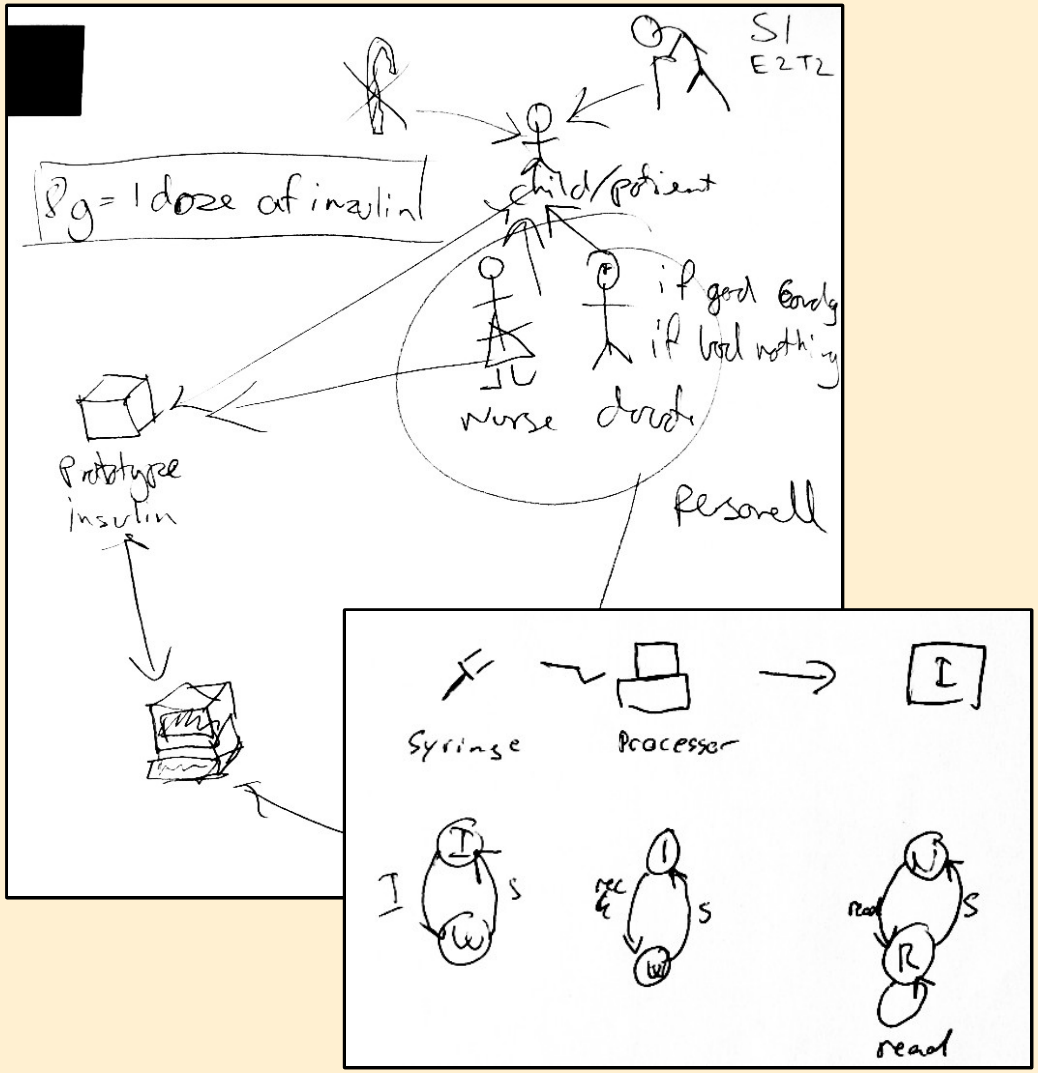
- Varied approaches
- Only one subject used recommended approach
- Manual construction of the supervisory solution
  - Sub-optimal
  - Passive control
  - No need for formal specifications
  - Simpler manual verification

Some findings (2)

- Mistakes
  - Usually of poor judgement
  - Small errors that render the whole solution incorrect
- Frequent reference to
  - Problem description
  - Previous versions of models
    - ...even if incorrect

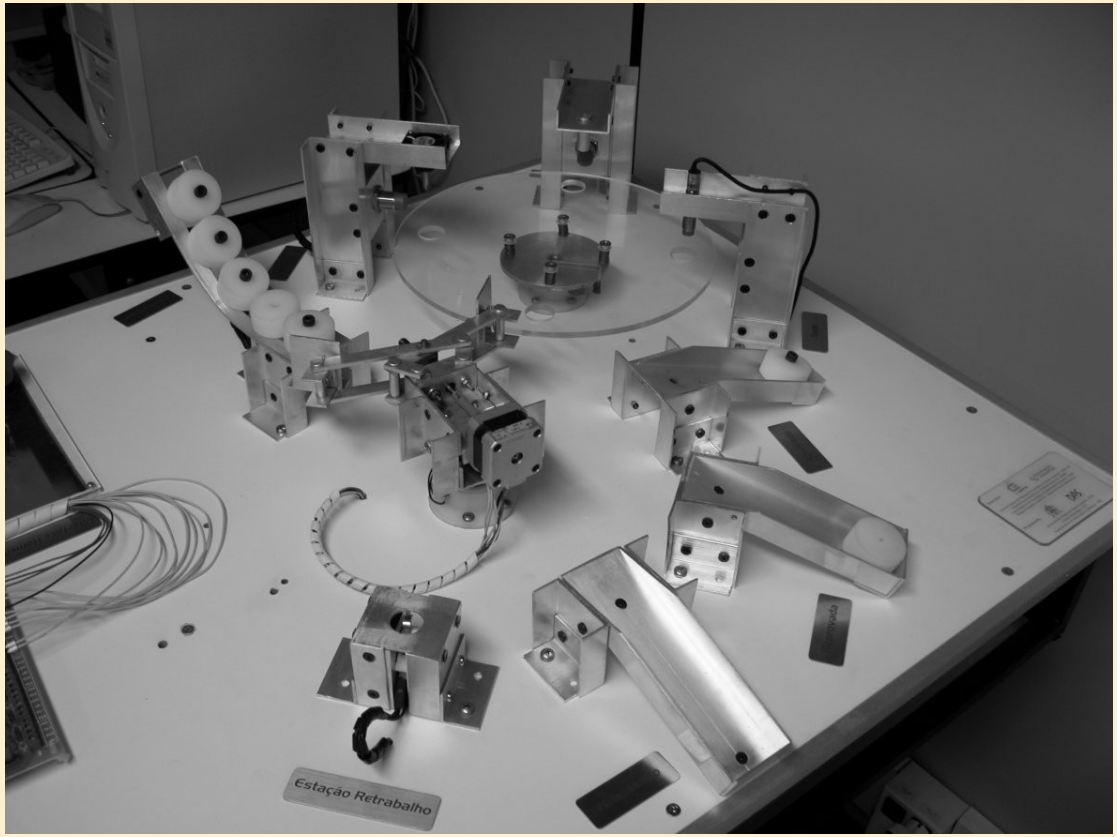
Findings (3)

- System diagrams
- Synchronization of modules
  - Demanding
  - Error-prone
- Operations and relations considered at the high level of modelling



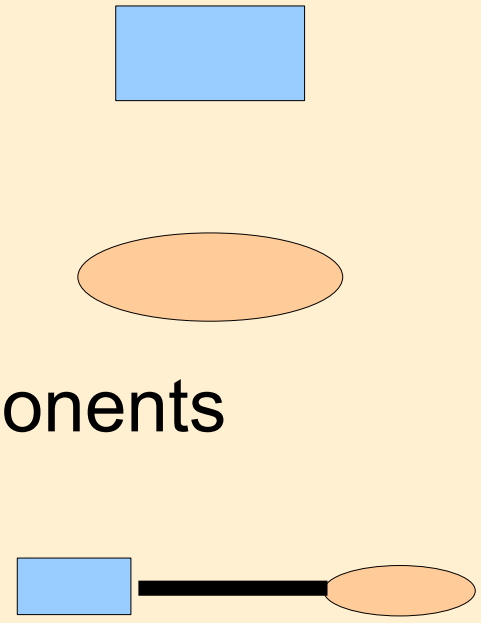
Robotic testbed control

- Frequent reconfiguration of model
- Changes in control specifications
- Need for PLC implementation of supervisor



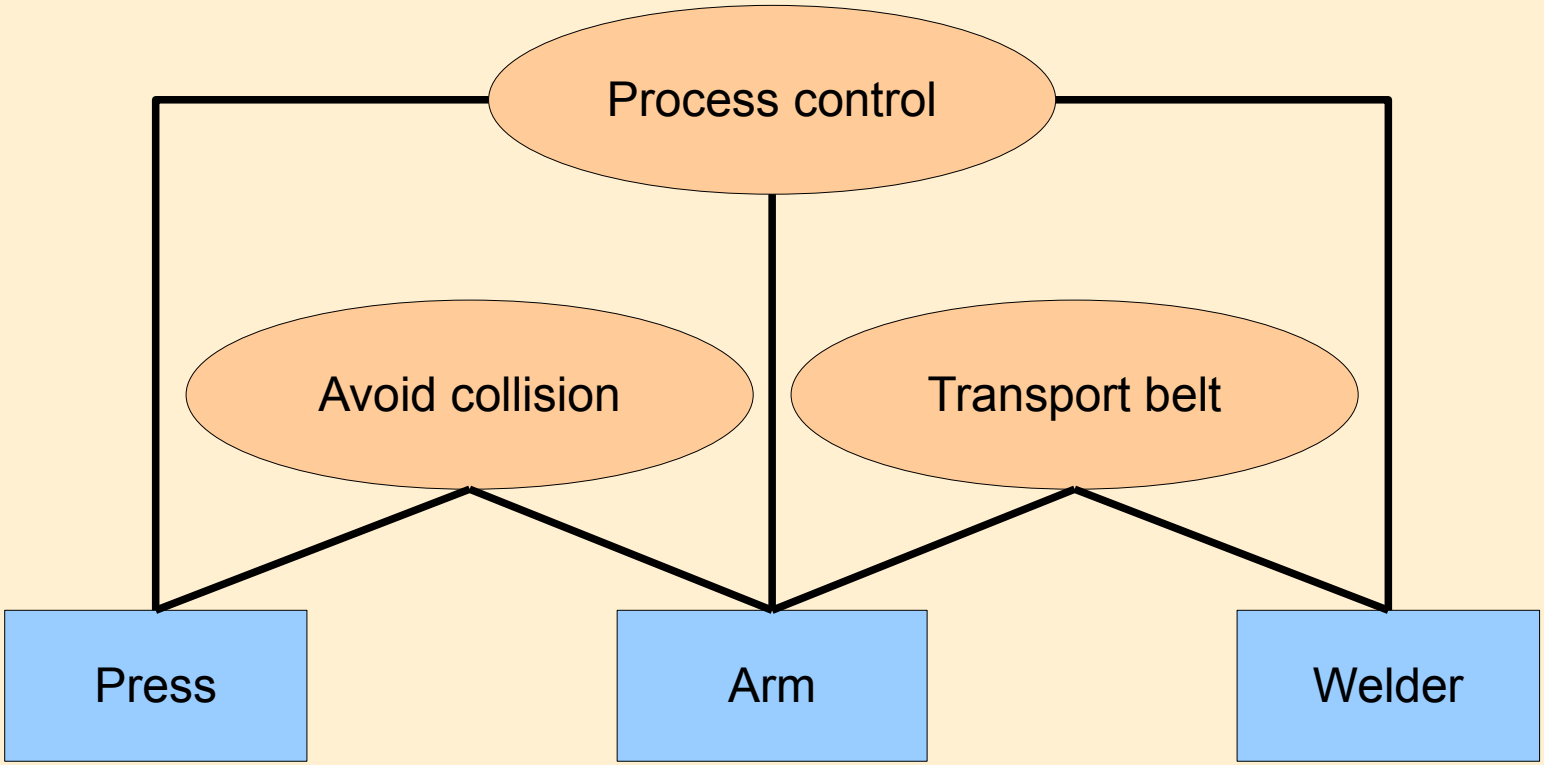
Conceptual design

- Active components
  - Event generators
- Passive components
  - Protocols between active components
- Connections between them
  - Synchronization
- Simultaneous structural & functional design

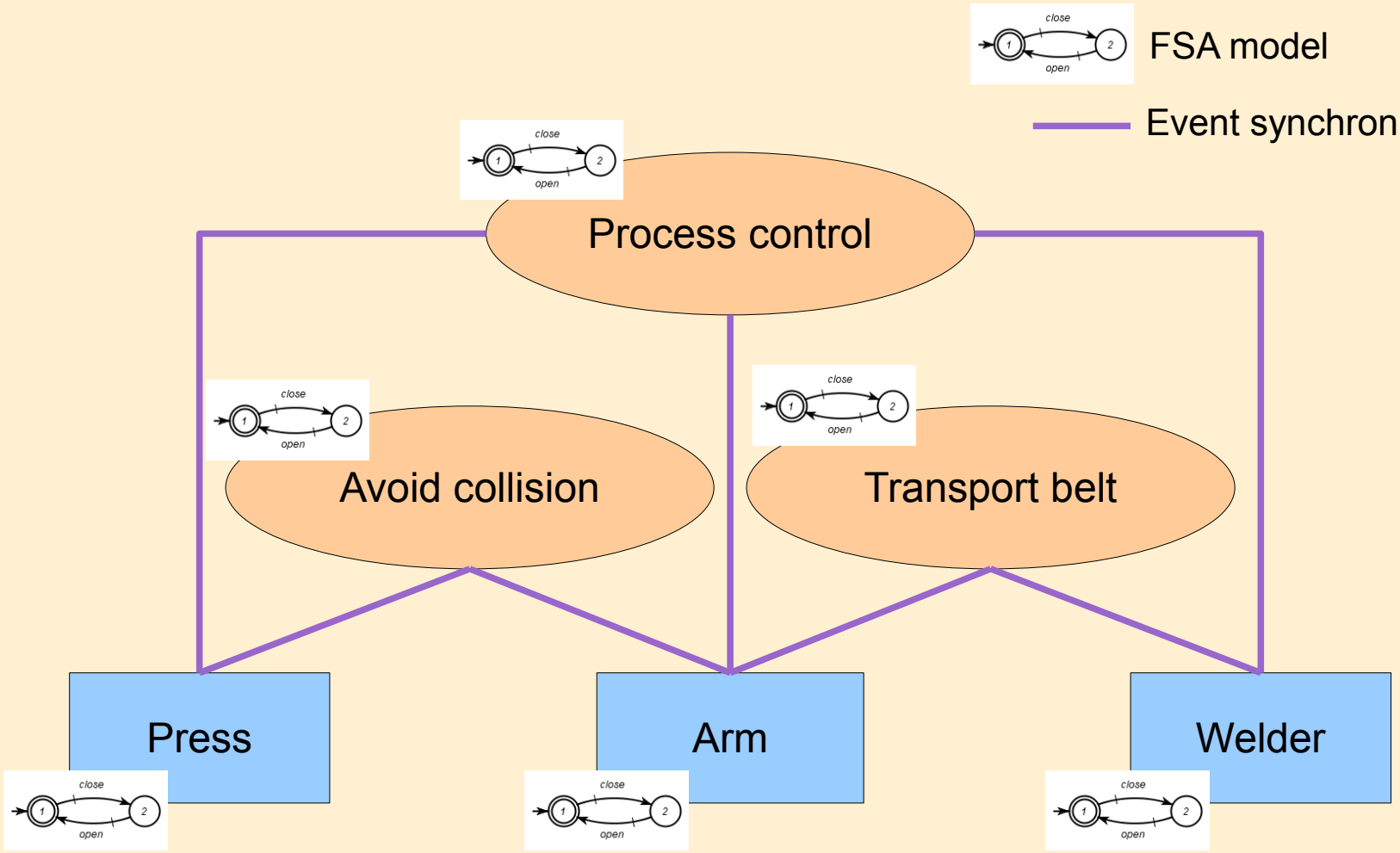


(Santos *et al.*, A computational model for supporting conceptual design of automatic systems, 2001)

Example diagram

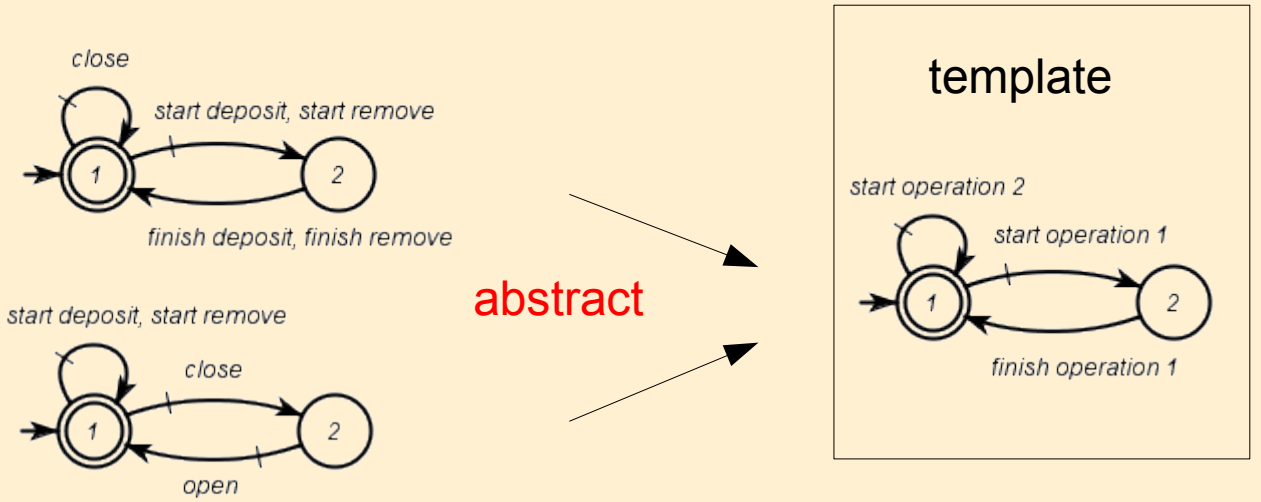


Example diagram



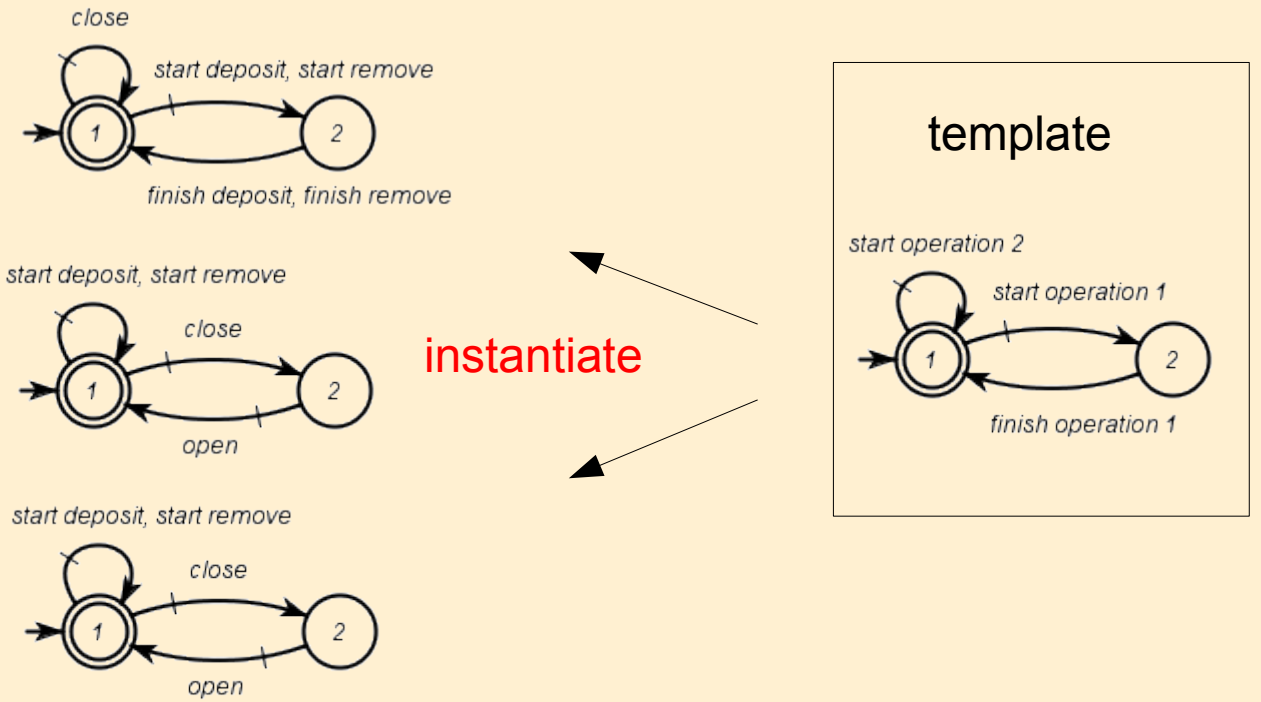
Templates (abstract)

- Abstract common behavior



Templates (instantiate)

- Instantiate existing templates



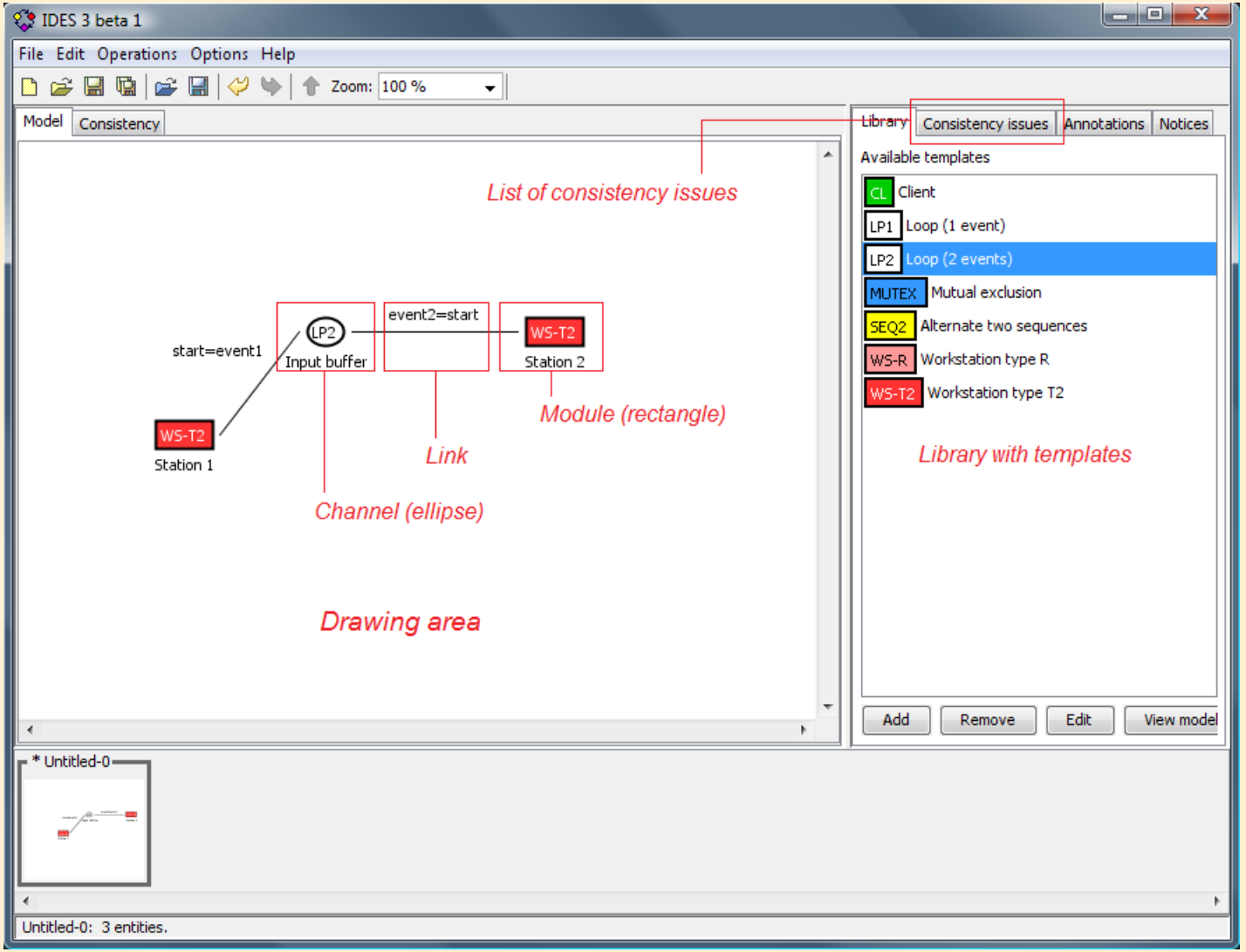
Templates

- Faster
  - No need to remodel similar behavior
  - Can reconfigure model as needed
- More robust
  - Can be thoroughly tested
  - Synchronization is independent of events
- Easier
  - Graphical conceptual designs
  - Sharing and reuse of existing models

Template design implementation

- Prototype implementation
  - Observed the user experience
  - PLC code generation
- Final implementation
  - Plugin for the IDES software package
  - Reuse of FSA capabilities and existing algorithms
  - Take advantage of results in HCI

Interface


IDES 3 beta 1
File Edit Operations Options Help
Zoom: 100 %
Model
Consistency
Library
Consistency issues
Annotations
Notices
List of consistency issues
Available templates
CL Client
LP1 Loop (1 event)
LP2 Loop (2 events)
MUTEX Mutual exclusion
SEQ2 Alternate two sequences
WS-R Workstation type R
WS-T2 Workstation type T2
start=event1
LP2
Input buffer
event2=start
WS-T2
Station 2
WS-T2
Station 1
Module (rectangle)
Link
Channel (ellipse)
Library with templates
Drawing area
Add
Remove
Edit
View model
* Untitled-0
Untitled-0: 3 entities.

Evaluation

- Twelve subjects
  - Graduate students with DES background
  - From different research groups
- Solve two equivalent problems
  - Modify existing solution
  - Using IDES
  - With and without the template design environment

Metrics

- Rate of task completion
- Time of task completion
  - Time to supervisor computation
  - Total time
- Error rate
- Experiential confidence
- Experiential learnability
- System Usability Scale

Results

- Significant evidence that it is faster to use template designs
- No evidence for variety in
  - Error rate
  - Experiential confidence
  - Experiential learnability

- The SUS for template design is higher
- Subjects preferred the template design environment

Contributions according to subjects

- High-level structure
- (Handling of self-loops in the specifications)
- Templates
- Automation of modelling
- Modelling is less error-prone
- Speed of modelling
- Convenient user interface

Conclusions

- Observations of DES problem solving
  - Described new aspects of problem solving
  - Concretized suspected issues
- Template design environment
  - Reinterprets existing theories
  - Results in faster modelling
  - Supports the reuse of models
  - Provides a more enjoyable experience

Future work

- Longitudinal evaluation of template design
- Parametrization of templates
- Extension with other modelling frameworks
  - Petri nets, temporal logic, inequalities ...
- Better tools
  - Solution verification
  - Implementation of control solutions
  - Runtime computation of supervisory control