2nd UCAAT
User Conference on Advanced Automated Testing
ETSI
World Class Standards
MUNICH, 16-18 September 2014
Organizers
ALL4TEC
German Testing Night

Cutting MBT Adoption Time with Domain Specific Modeling

Juha-Pekka Tolvanen, PhD, MetaCase

Stephan Schulz, PhD, Conformiq

Contents

- Introduction to DSM and MBT
- DSM + MBT = ?
- Case 1: Web application (IT)
- Case 2: Military radio (embedded)
- Results
- How to get started
- Summary, Q&A

Domain-Specific Modeling (DSM)

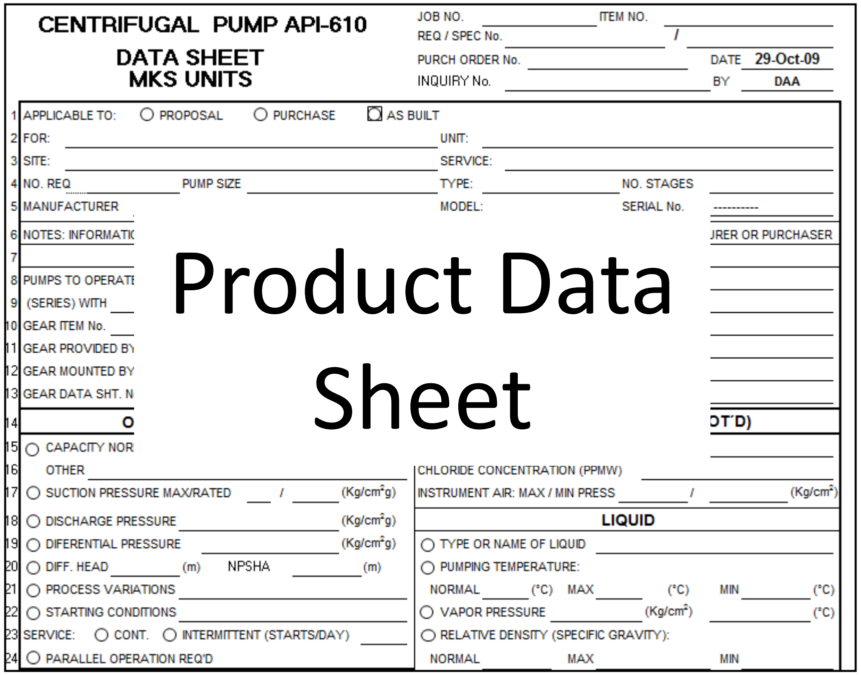
- Models expressed with domain concepts
  - No need to learn new languages
- Domain-Specific Modeling allows using:
  - existing terminology,
  - with known semantics, and
  - familiar notation
- DSM is applied in particular for automating repetitive development efforts*, but less in testing

* See references on EADS, NSN, Nokia, Panasonic, Polar Elektro, USAF

Example: Industrial Process Plant

Domain terminology and concepts

- Detailed information specifying functional & physical characteristics of a component of a system, plant or facility (e.g. pump)



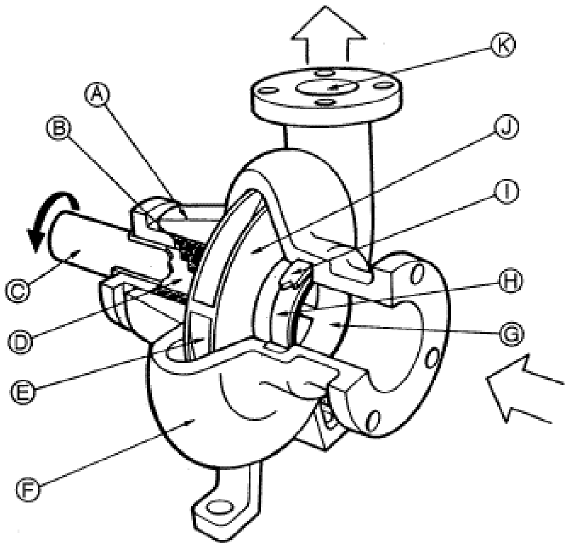
CENTRIFUGAL PUMP API-610

DATA SHEET
MKS UNITS

JOB NO. ____ ITEM NO. ____
REQ / SPEC No. ____ / ____
PURCH ORDER No. ____ DATE 29-Oct-09
INQUIRY No. ____ BY DAA

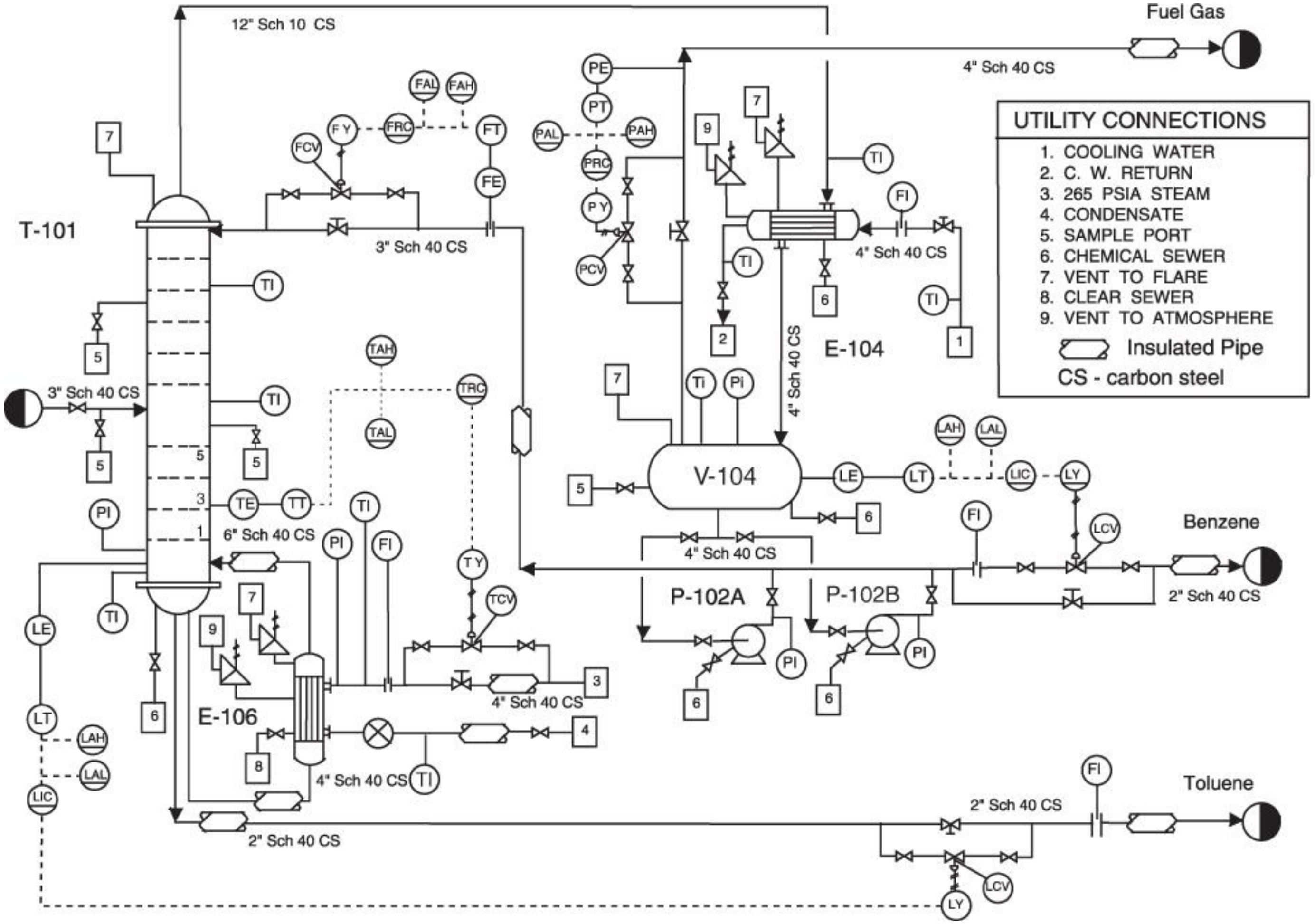
1 APPLICABLE TO: ○ PROPOSAL ○ PURCHASE ☐ AS BUILT
2 FOR: ____ UNIT: ____
3 SITE: ____ SERVICE: ____
4 NO. REQ ____ PUMP SIZE ____ TYPE: ____ NO. STAGES ____
5 MANUFACTURER MODEL: SERIAL No. ----------
6 NOTES: INFORMATIC JRER OR PURCHASER
7
8 PUMPS TO OPERATE
9 (SERIES) WITH ____
10 GEAR ITEM No. ____
11 GEAR PROVIDED BY
12 GEAR MOUNTED BY
13 GEAR DATA SHT. N
14 O OT´D)
15 ○ CAPACITY NOR
16 OTHER ____ CHLORIDE CONCENTRATION (PPMW) ____
17 ○ SUCTION PRESSURE MAX/RATED ____ / ____ (Kg/cm^{2}g) INSTRUMENT AIR: MAX / MIN PRESS ____ / ____ (Kg/cm^2)
18 ○ DISCHARGE PRESSURE ____ (Kg/cm^{2}g) **LIQUID**
19 ○ DIFERENTIAL PRESSURE ____ (Kg/cm^{2}g) ○ TYPE OR NAME OF LIQUID ____
20 ○ DIFF. HEAD ____ (m) NPSHA ____ (m) ○ PUMPING TEMPERATURE:
21 ○ PROCESS VARIATIONS ____ NORMAL ____ (°C) MAX ____ (°C) MIN ____ (°C)
22 ○ STARTING CONDITIONS ____ ○ VAPOR PRESSURE ____ (Kg/cm^2) ____ (°C)
23 SERVICE: ○ CONT. ○ INTERMITTENT (STARTS/DAY) ____ ○ RELATIVE DENSITY (SPECIFIC GRAVITY):
24 ○ PARALLEL OPERATION REQ'D NORMAL ____ MAX ____ MIN ____

Product Data Sheet



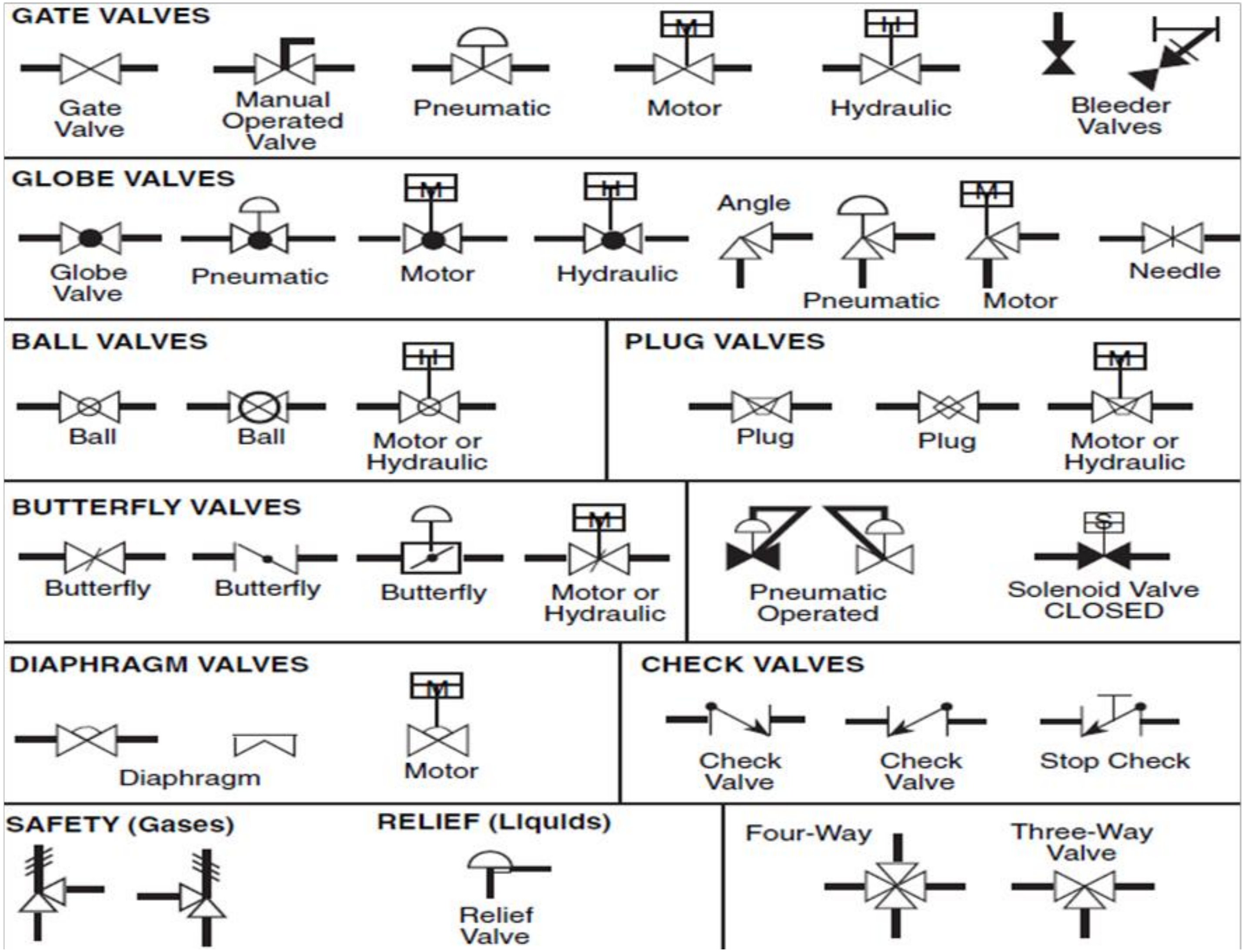
A Stuffing Box
B Packing
C Shaft
D Shaft Sleeve
E Vane
F Casing
G eye of Impeller
H Impeller
I Casing wear ring
J Impeller
K Discharge nozzle

Design with domain-concepts

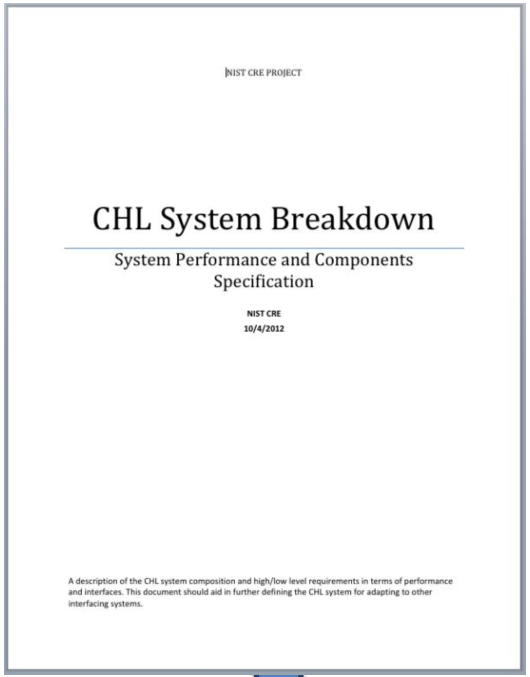


* Turton et al., Analysis, Synthesis and Design of Chemical Processes, Prentice Hall. 2012

Domain terminology: valves



Example Specification



NIST CRE PROJECT

CHL System Breakdown

System Performance and Components Specification

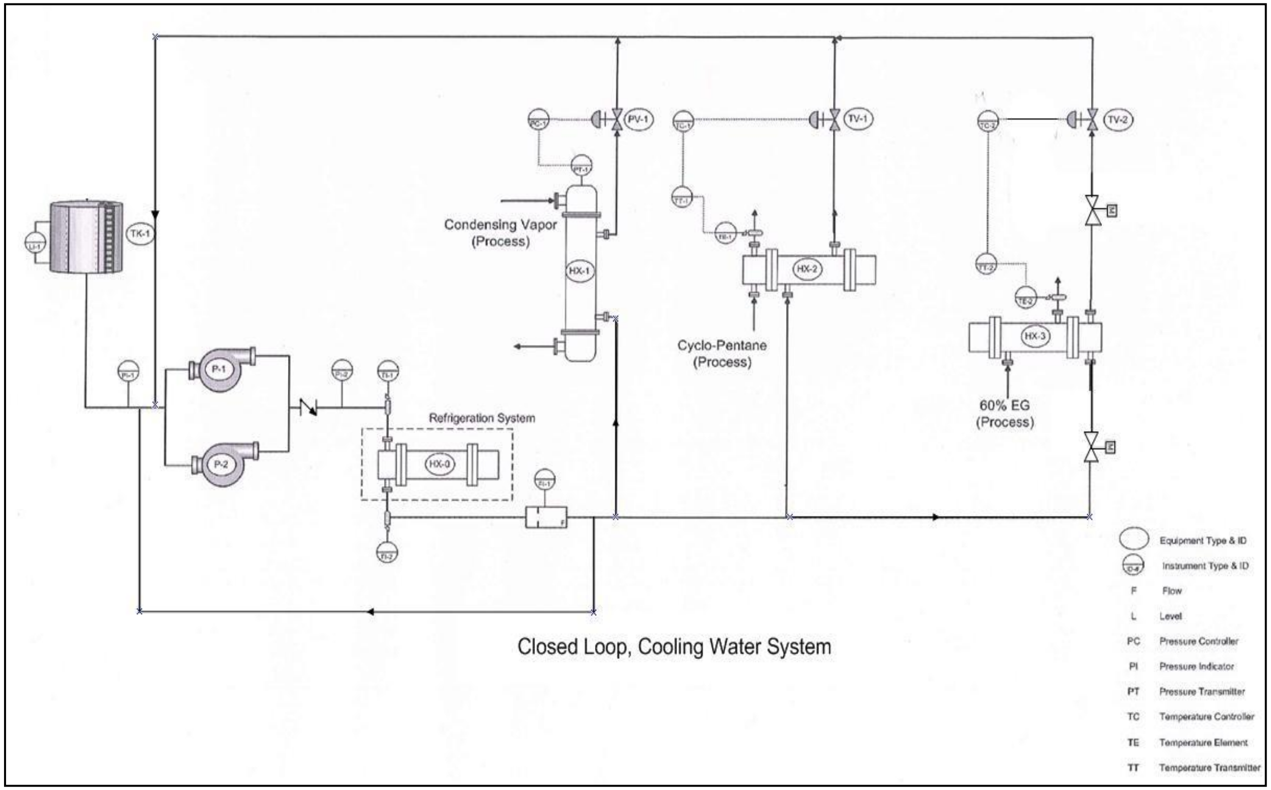
NIST CRE
10/4/2012

A description of the CHL system composition and high/low level requirements in terms of performance and interfaces. This document should aid in further defining the CHL system for adapting to other interfacing systems.

May include:

- System Requirements Tree
- System Requirements
- Component Requirements
- Interface Requirements

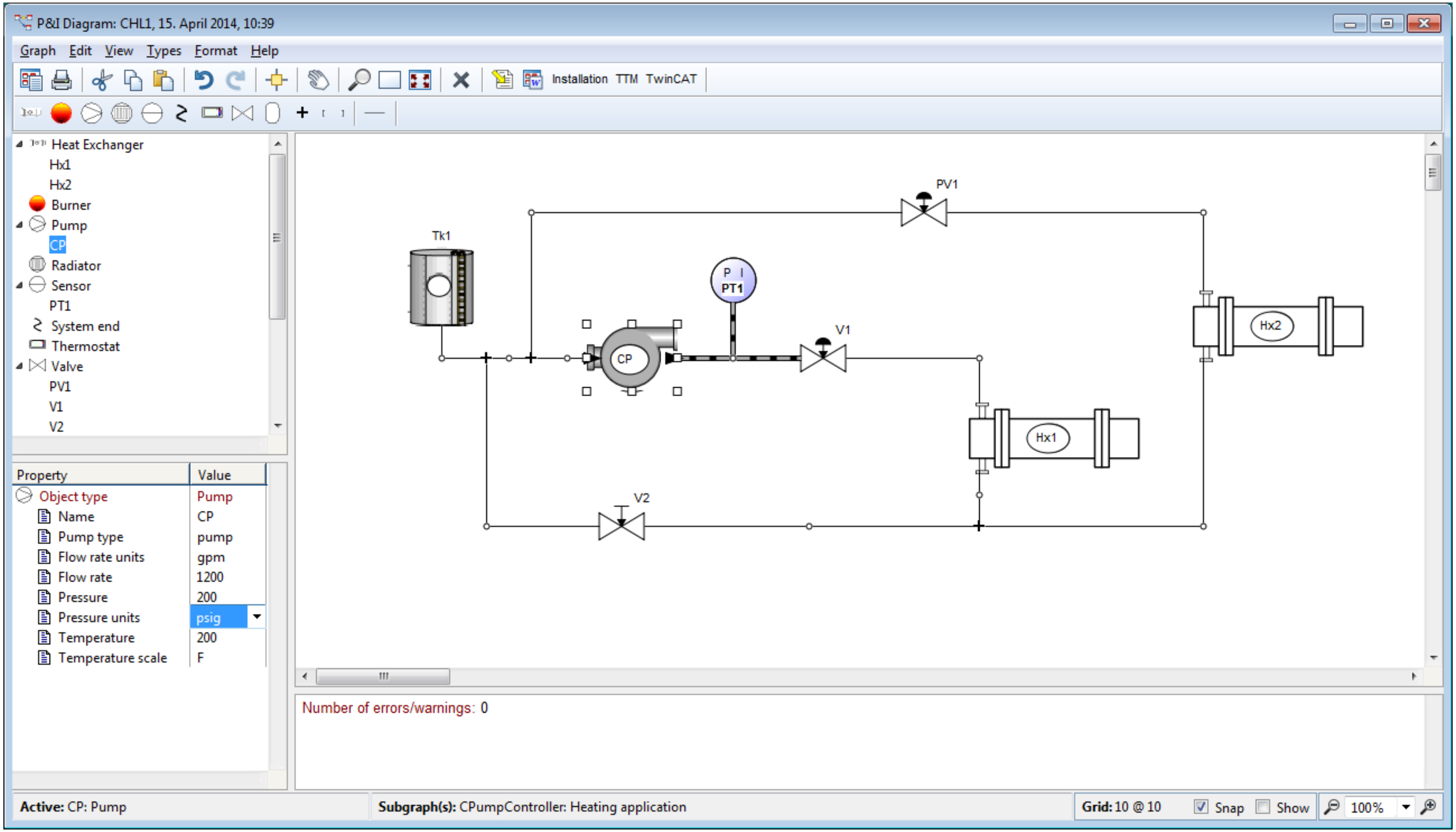
Closed loop, Heat transfer, Liquid circulating (CHL)



How to test a cooling system?

- Temperature
  - Produce too much heat?
- Pressure
  - Incorrect input/output pressure?
- Flow rates
  - Conflicting flow rates in the configuration?
- Control logic
- Instrument configuration

Example: Cooling in process plant*



Property	Value
Object type	Pump
Name	CP
Pump type	pump
Flow rate units	gpm
Flow rate	1200
Pressure	200
Pressure units	psig
Temperature	200
Temperature scale	F

Number of errors/warnings: 0

Active: CP: Pump

Subgraph(s): CPumpController: Heating application

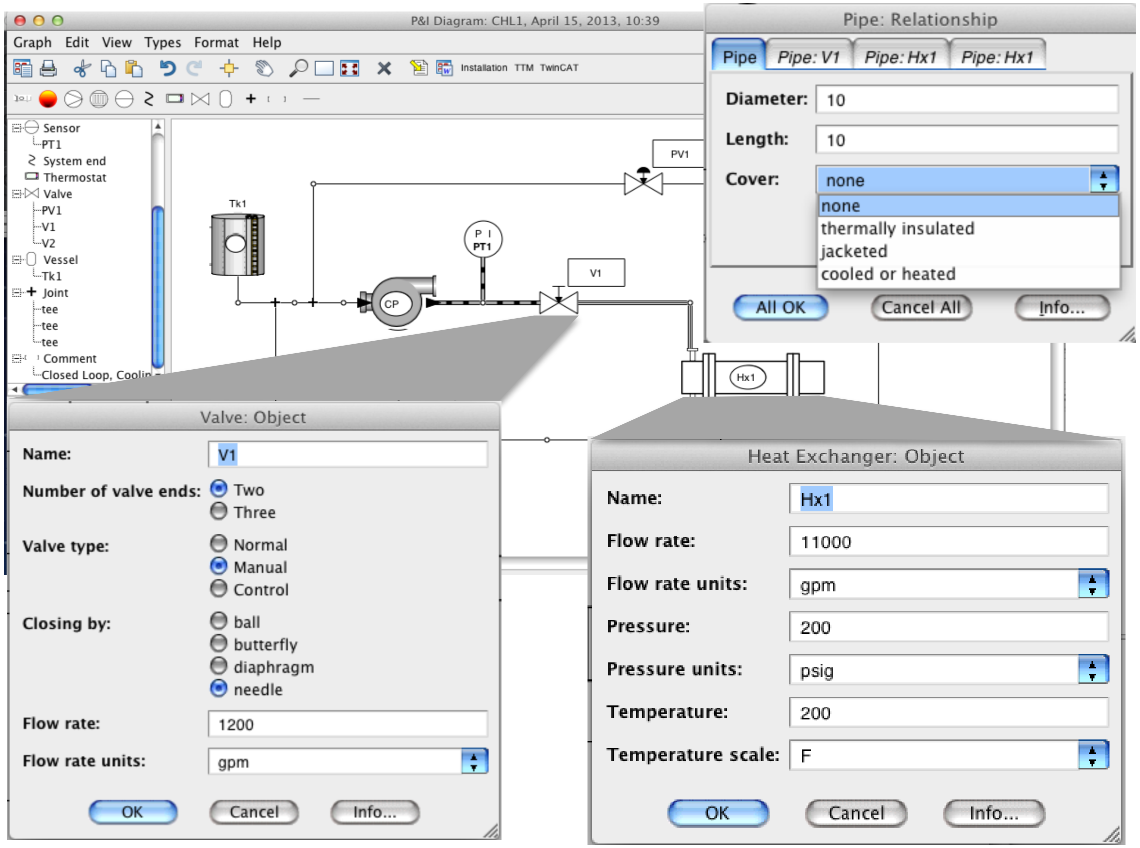
Grid: 10 @ 10 ☑ Snap □ Show 100%

* M. Blackburn, P. Denno, Virtual Design and Verification of Cyber-Physical Systems: Industrial Process Plant Design, Procedia Computer Science 28, Elsevier, 2014

Specifying properties of components

Generic

Specific



Pipe: Relationship

Pipe | Pipe: V1 | Pipe: Hx1 | Pipe: Hx1

Diameter: 10

Length: 10

Cover: none

- none
- thermally insulated
- jacketed
- cooled or heated

All OK | Cancel All | Info...

Valve: Object

Name: V1

Number of valve ends: ● Two ○ Three

Valve type: ○ Normal ● Manual ○ Control

Closing by: ○ ball ○ butterfly ○ diaphragm ● needle

Flow rate: 1200

Flow rate units: gpm

OK | Cancel | Info...

Heat Exchanger: Object

Name: Hx1

Flow rate: 11000

Flow rate units: gpm

Pressure: 200

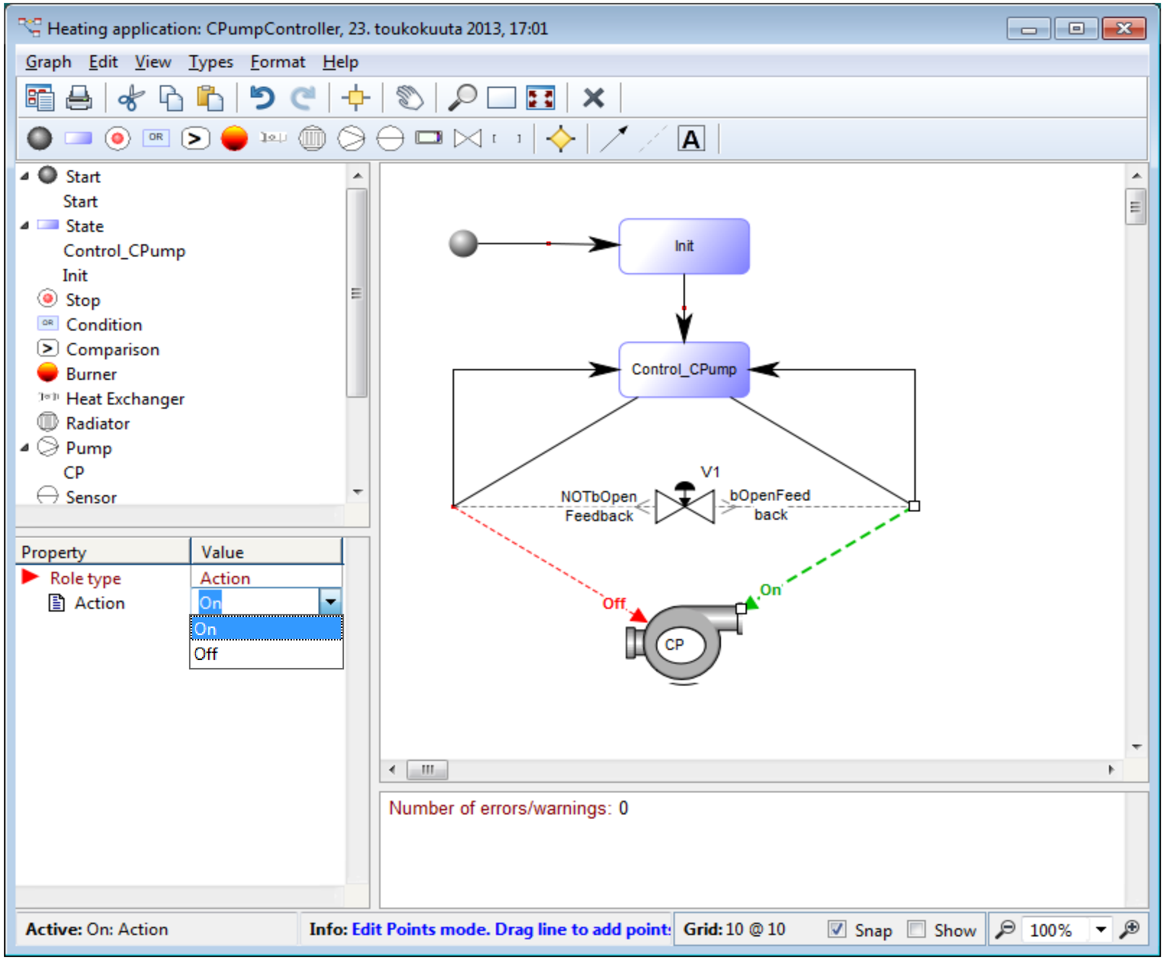
Pressure units: psig

Temperature: 200

Temperature scale: F

OK | Cancel | Info...

Example: Cooling in process plant*



Heating application: CPumpController, 23. toukokuuta 2013, 17:01

Property	Value
Role type	Action
Action	On

Number of errors/warnings: 0

Active: On: Action

Info: Edit Points mode. Drag line to add point:

Grid: 10 @ 10

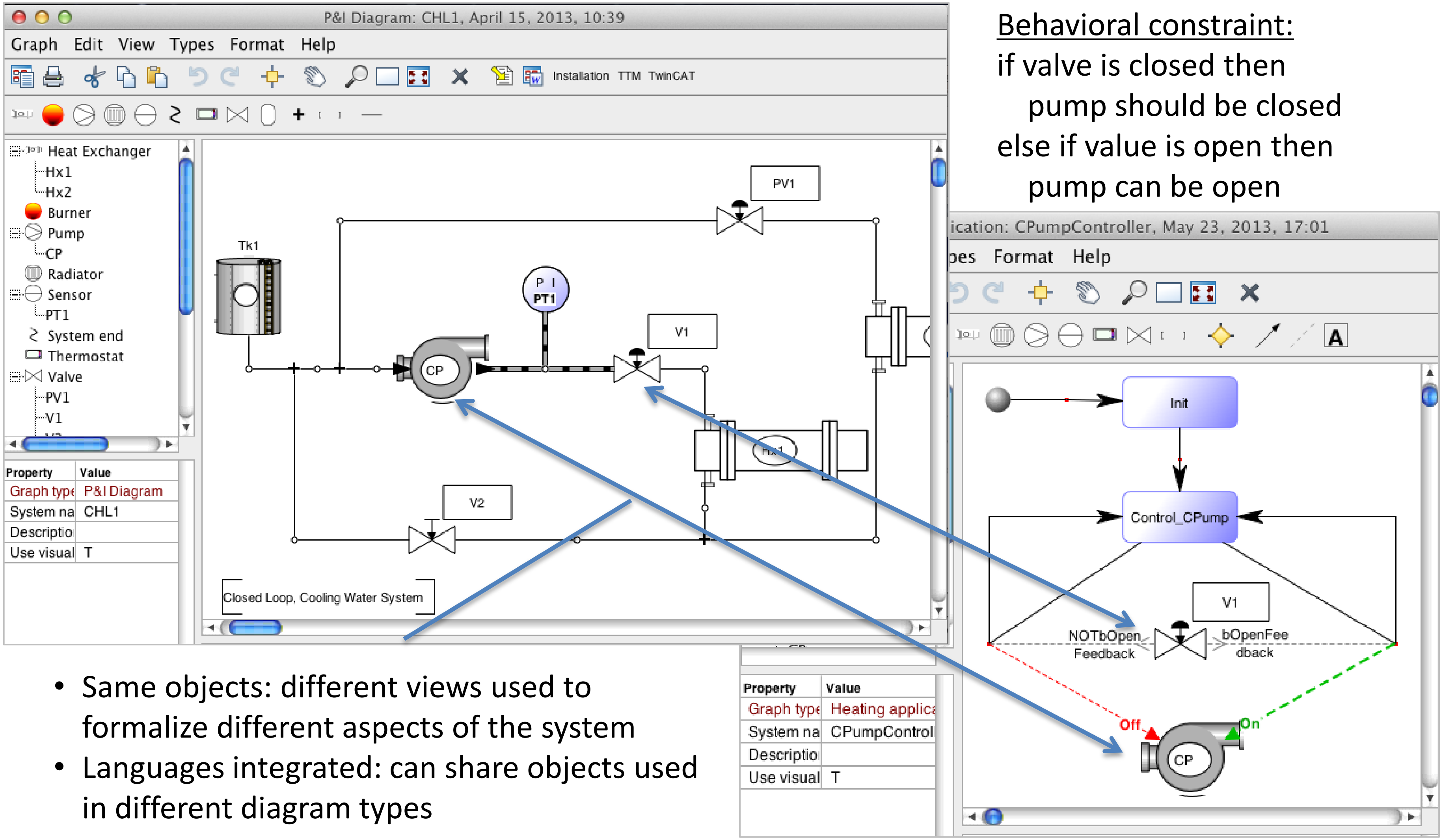
☑ Snap □ Show

* M. Blackburn, P. Denno, Virtual Design and Verification of Cyber-Physical Systems: Industrial Process Plant Design, Procedia Computer Science 28, Elsevier, 2014

Both structure and behavior

<u>Behavioral constraint:</u>

if valve is closed then
  pump should be closed
else if value is open then
  pump can be open



Property	Value
Graph type	P&I Diagram
System na	CHL1
Descriptio	
Use visual	T

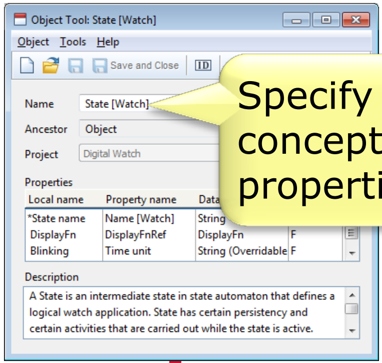
Property	Value
Graph type	Heating applica
System na	CPumpControl
Descriptio	
Use visual	T

- Same objects: different views used to formalize different aspects of the system
- Languages integrated: can share objects used in different diagram types

Domain-Specific means:

- Use of concepts from the problem domain
  - Already familiar => no need to learn new
  - Have known semantics
- Having a special focus
  - Use concepts that are relevant for the task: testing, verification, validation
- Use concrete syntax that enables communication and collaboration
  - Not a cryptic programming/scripting language
  - Apply style close to the domain’s natural representation

Steps for Defining Domain-Specifc Modeling Languages and Generators



Name: State [Watch]

Ancestor: Object

Project: Digital Watch

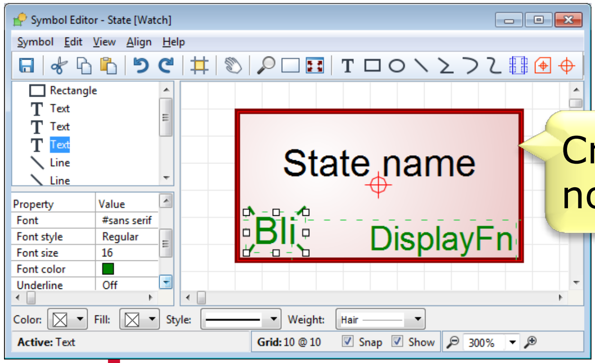
Properties

Local name	Property name	Data	
*State name	Name [Watch]	String	
DisplayFn	DisplayFnRef	DisplayFn	F
Blinking	Time unit	String (Overridable	F

Description

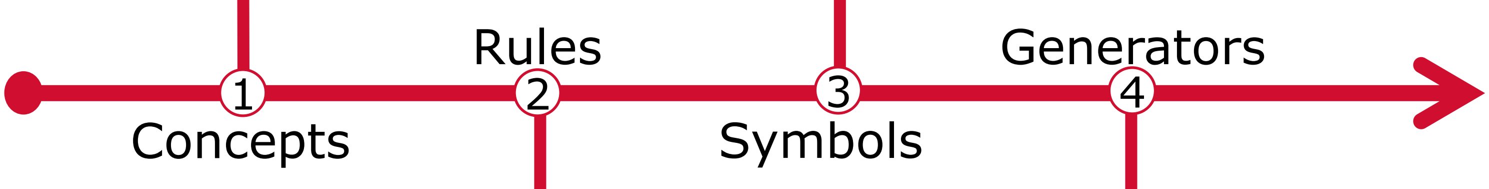
A State is an intermediate state in state automaton that defines a logical watch application. State has certain persistency and certain activities that are carried out while the state is active.

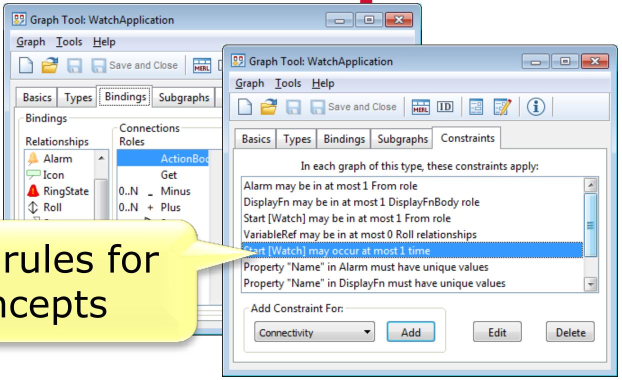
Specify language concepts & their properties



Property	Value
Font	#sans serif
Font style	Regular
Font size	16
Font color	🟩
Underline	Off

Create a notation

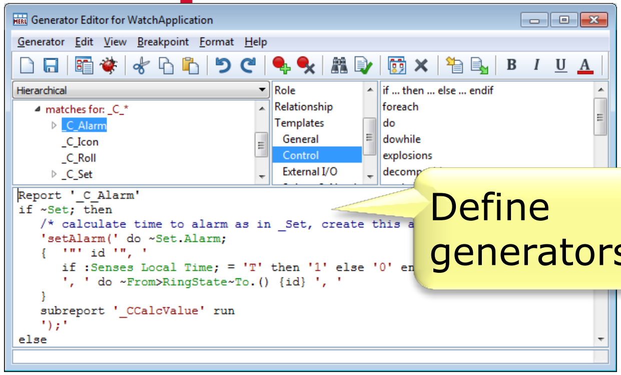




In each graph of this type, these constraints apply:

- Alarm may be in at most 1 From role
- DisplayFn may be in at most 1 DisplayFnBody role
- Start [Watch] may be in at most 1 From role
- VariableRef may be in at most 0 Roll relationships
- Start [Watch] may occur at most 1 time
- Property "Name" in Alarm must have unique values
- Property "Name" in DisplayFn must have unique values

Define rules for the concepts

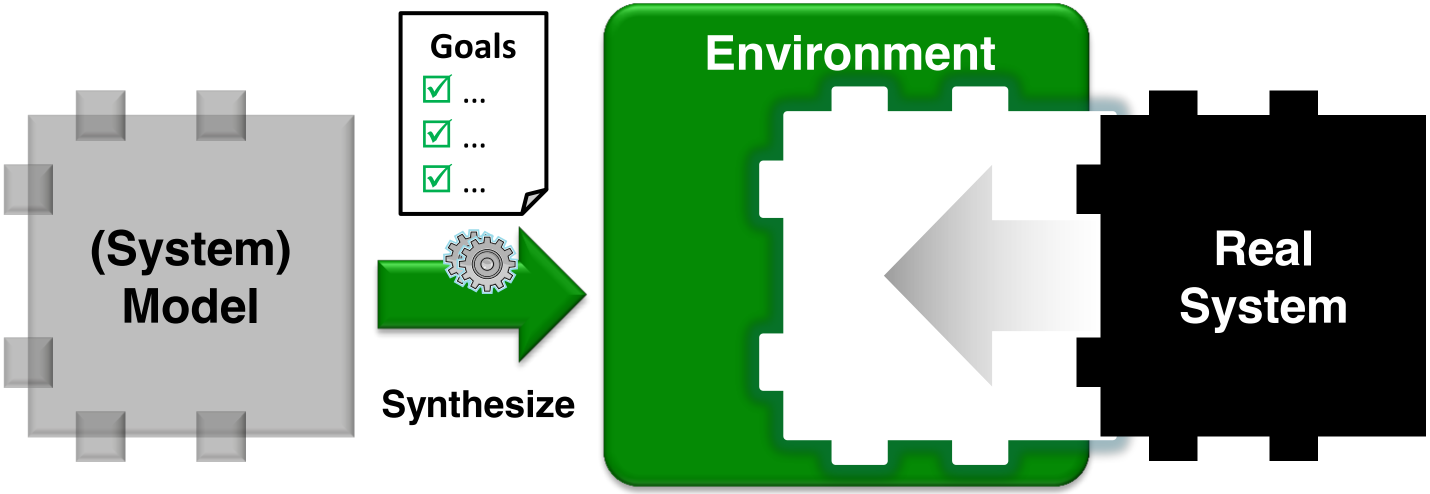


```
Report '_C_Alarm'
if ~Set; then
   /* calculate time to alarm as in _Set, create this a
   'setAlarm(' do ~Set.Alarm;
   {  '"' id '", '
      if :Senses Local Time; = 'T' then '1' else '0' en
      ', ' do ~From>RingState~To.() {id} ', '
   }
   subreport '_CCalcValue' run
   ');'
else
```

Define generators

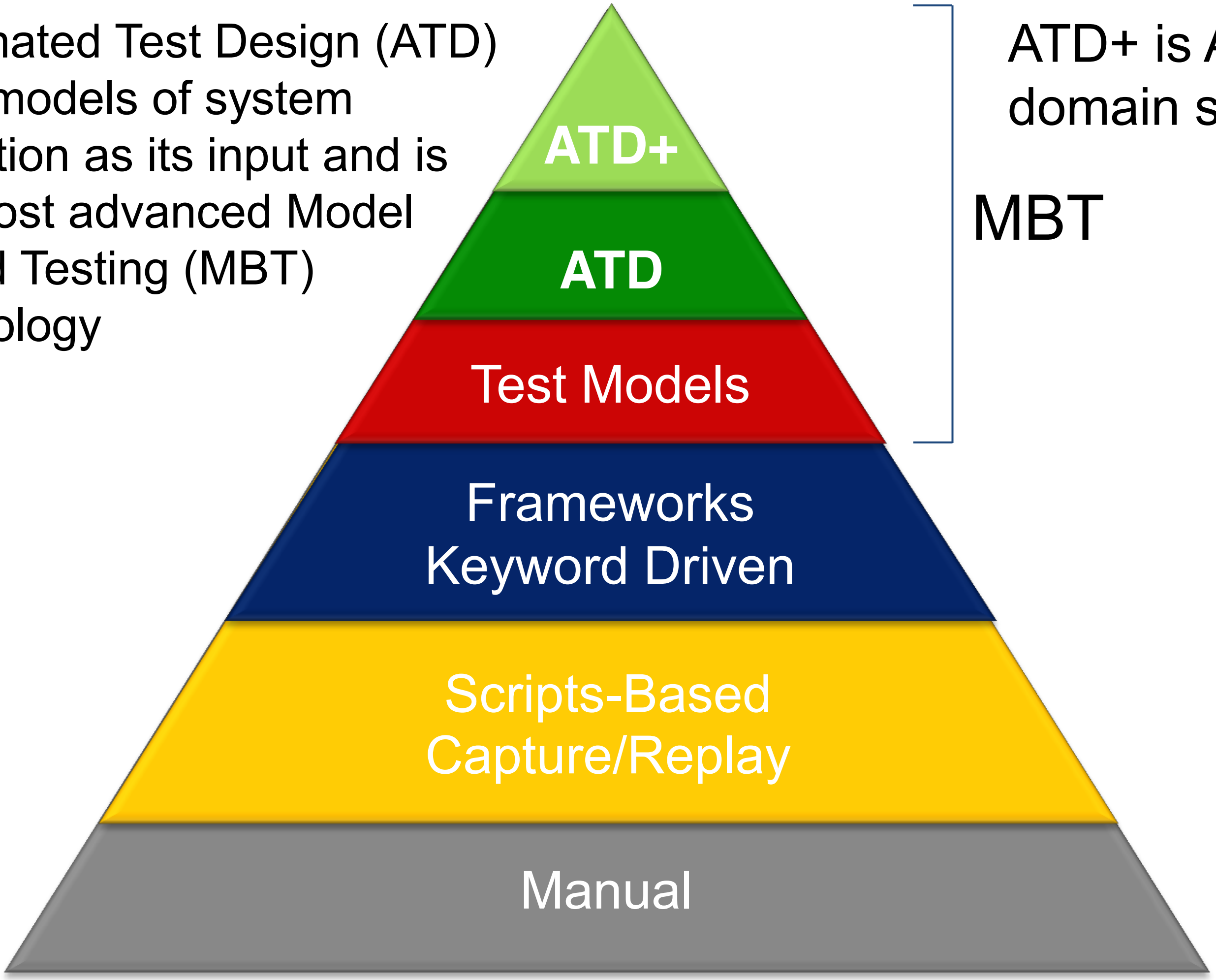
About Model-Based Testing (MBT)

- Umbrella term for using models in a testing context
- One approach is to use MBT for automating *test design*
  - Here model reflects operation of the system to be tested
  - MBT *complements* test execution
  - Recognized by worldwide industrial standards (ETSI)



Evolution of Software Testing

Automated Test Design (ATD) uses models of system operation as its input and is the most advanced Model Based Testing (MBT) technology



ATD+ is ATD driven by a domain specific language

Test Approach Comparison Heat Map

Test Approach	Test Coverage	Early Problem Discovery	Functional Complexity	Test Artifact Reuse	Required Skill Set	Test Process Optimization	Productivity Gain Initial	Productivity Gain Iteration
Manual Test	2	2	2	0	2	1	1	1
Test Scripts	5	5	6	6	7	4	4	3
Test Modeling	7	5	5	4	5	6	7	6
Automated Test Design	10	8	8	8	8	8	6	8
DSL Driven ATD	10	8	8	9	4	8	8	9

ATD+: DSL driven MBT

- Draws from all benefits of conventional ATD
  - Automated test design and traceability
  - Integration into test automation ecosystem
  - 5x improvements in productivity
- Enables testers to model system operation
  - No longer programming skills required
  - Less training and faster ramp up
- Allows other stakeholders to review models
  - "Shift (really) left" … engage your customer!

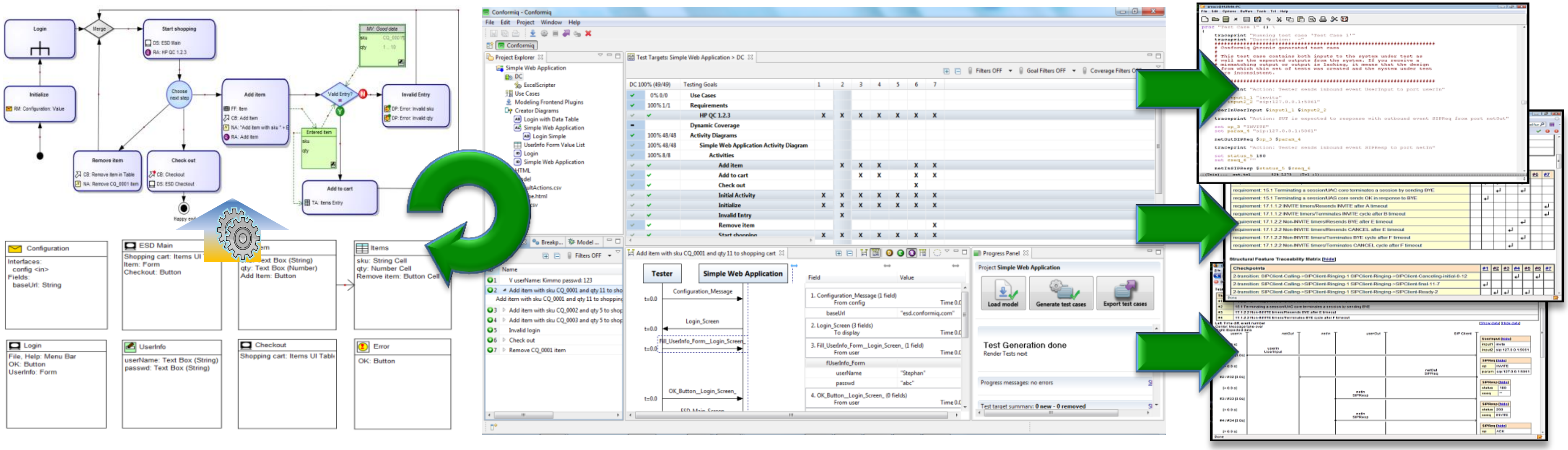
~5x (DSL) combined with ~5x (ATD) = **???**

Automated Test Design Workflow

Model
System Operation

Direct & Review
Test Design

Generate Test Scripts
& Documentation



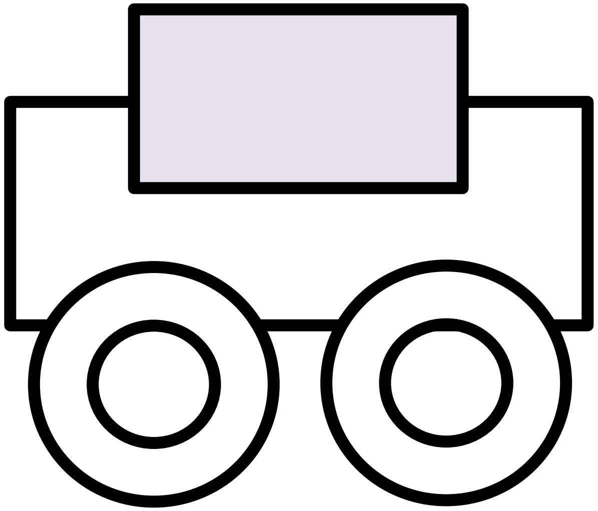
Domain Specific
Modeling Tool

Model Based
Test Design Tool

Test Execution
Tool(s)

Why are DSLs so Important in Testing?

rectangle(3,1, grey)
rectangle(5,2)
circle (2), circle(2)
circle(1), circle(1)

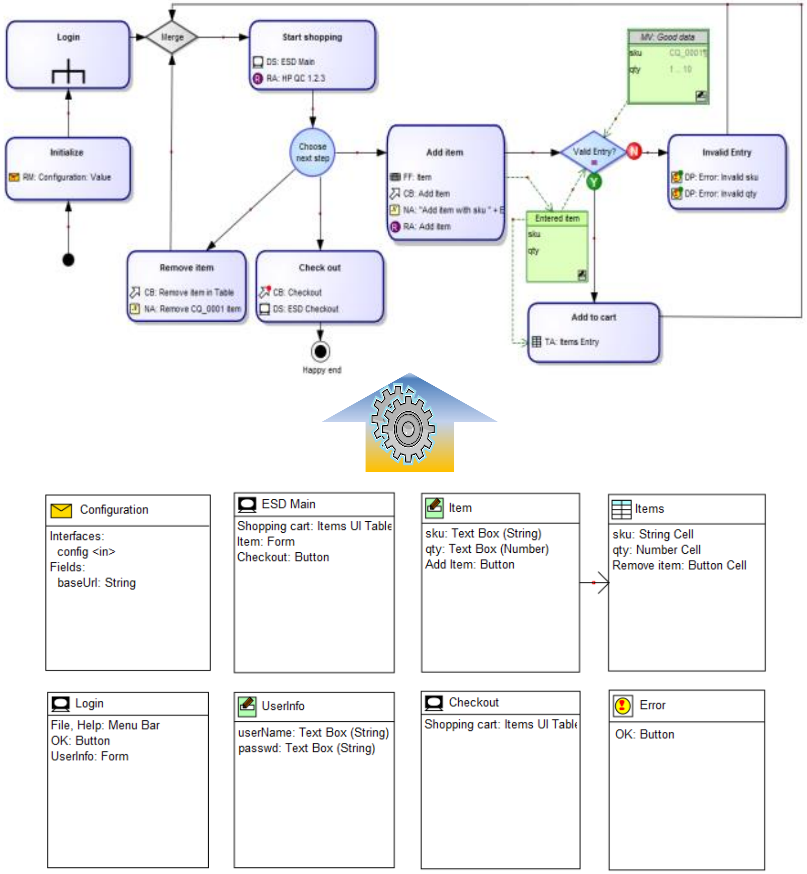


Testing is about achieving a common understanding

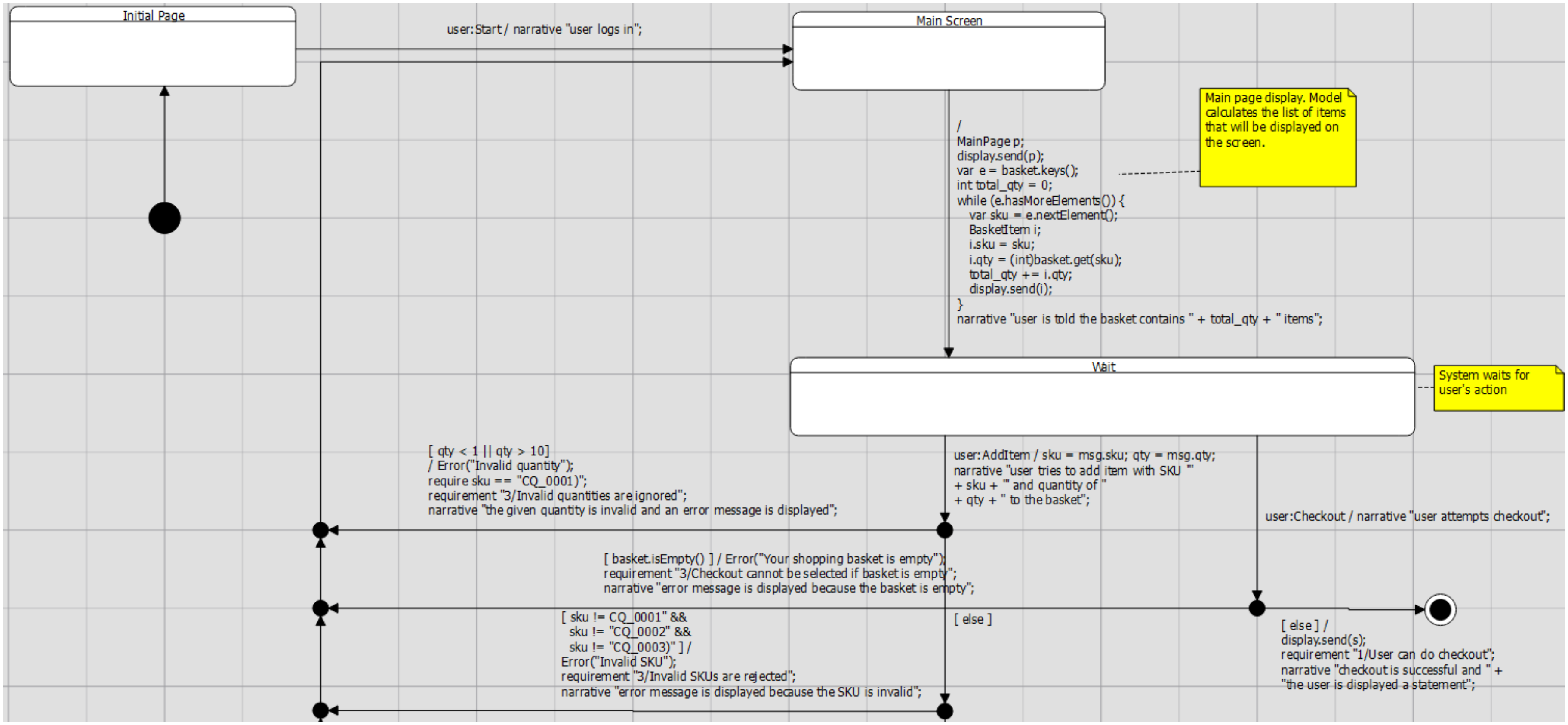
Case 1: Conformiq Creator

Generic
Specific

- A DSL developed for
  - Modeling system operation for *system & system integration & end-to-end testing*
  - First focus on *Enterprise IT applications*, frontends, backends, systems, etc.
  - Target testers and SMEs
- Encodes best practice
  - Provides set of pre-defined modeling building blocks



Modeling before Creator



```
user:Start / narrative "user logs in";
```

```
/
MainPage p;
display.send(p);
var  e = basket.keys();
int total_qty = 0;
while (e.hasMoreElements()) {
   var sku = e.nextElement();
   BasketItem i;
   i.sku = sku;
   i.qty = (int)basket.get(sku);
   total_qty += i.qty;
   display.send(i);
}
narrative "user is told the basket contains " + total_qty + " items";
```

Main page display. Model calculates the list of items that will be displayed on the screen.

System waits for user's action

```
user:AddItem / sku = msg.sku; qty = msg.qty;
narrative "user tries to add item with SKU '"
+ sku + "' and quantity of "
+ qty + " to the basket";
```

```
[ qty < 1 || qty > 10]
/ Error("Invalid quantity");
require sku == "CQ_0001)";
requirement "3/Invalid quantities are ignored";
narrative "the given quantity is invalid and an error message is displayed";
```

```
[ sku != CQ_0001" &&
  sku != "CQ_0002" &&
  sku != "CQ_0003)" ] /
Error("Invalid SKU");
requirement "3/Invalid SKUs are rejected";
narrative "error message is displayed because the SKU is invalid";
```

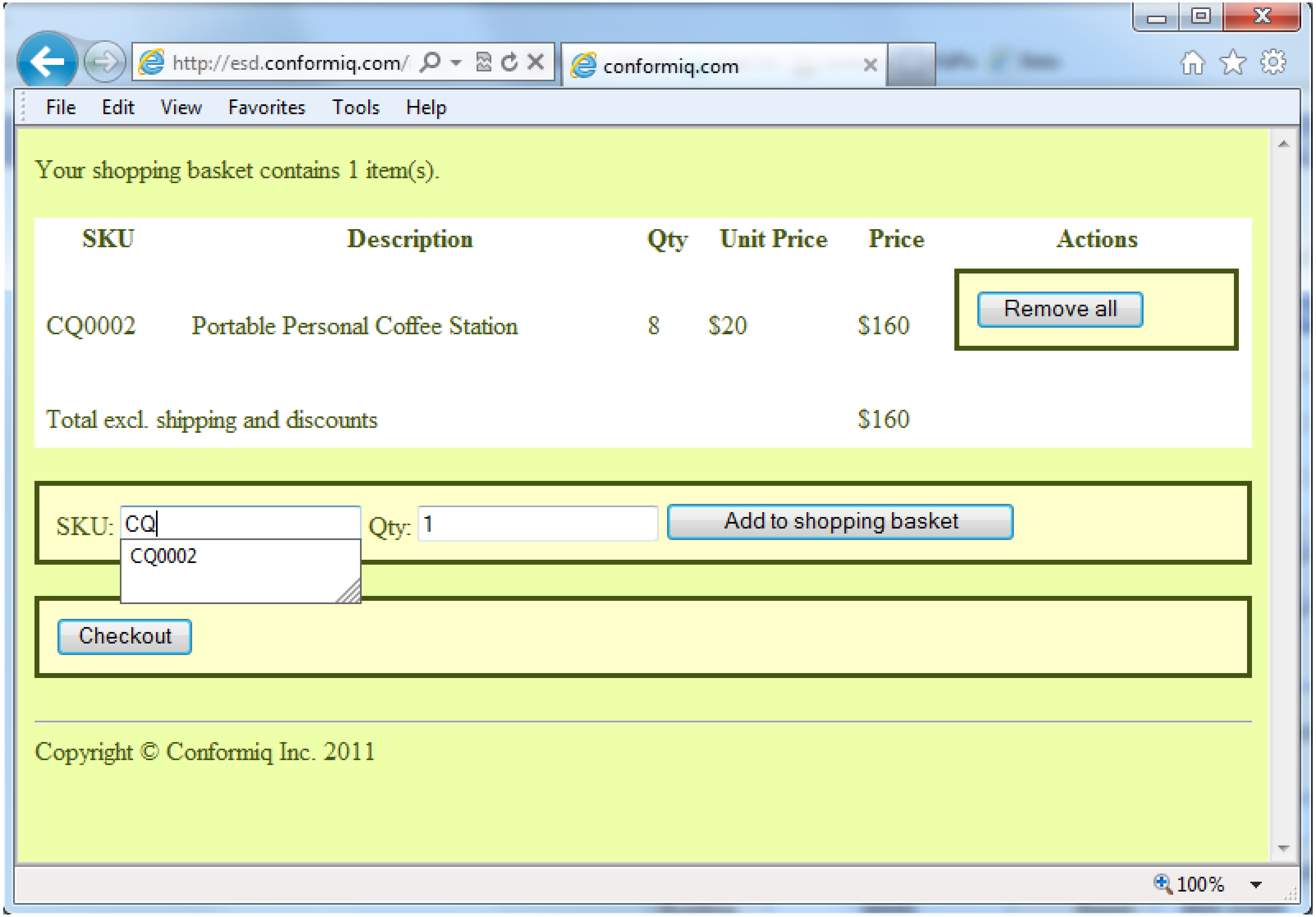
```
[ else ]
```

```
user:Checkout / narrative "user attempts checkout";
```

```
[ basket.isEmpty() ] / Error("Your shopping basket is empty");
requirement "3/Checkout cannot be selected if basket is empty";
narrative "error message is displayed because the basket is empty";
```

```
[ else ] /
display.send(s);
requirement "1/User can do checkout";
narrative "checkout is successful and " +
"the user is displayed a statement";
```

The Actual Application to Tested



Your shopping basket contains 1 item(s).

SKU	Description	Qty	Unit Price	Price	Actions
CQ0002	Portable Personal Coffee Station	8	$20	$160	Remove all
Total excl. shipping and discounts				$160	

SKU: CQ
CQ0002

Qty: 1

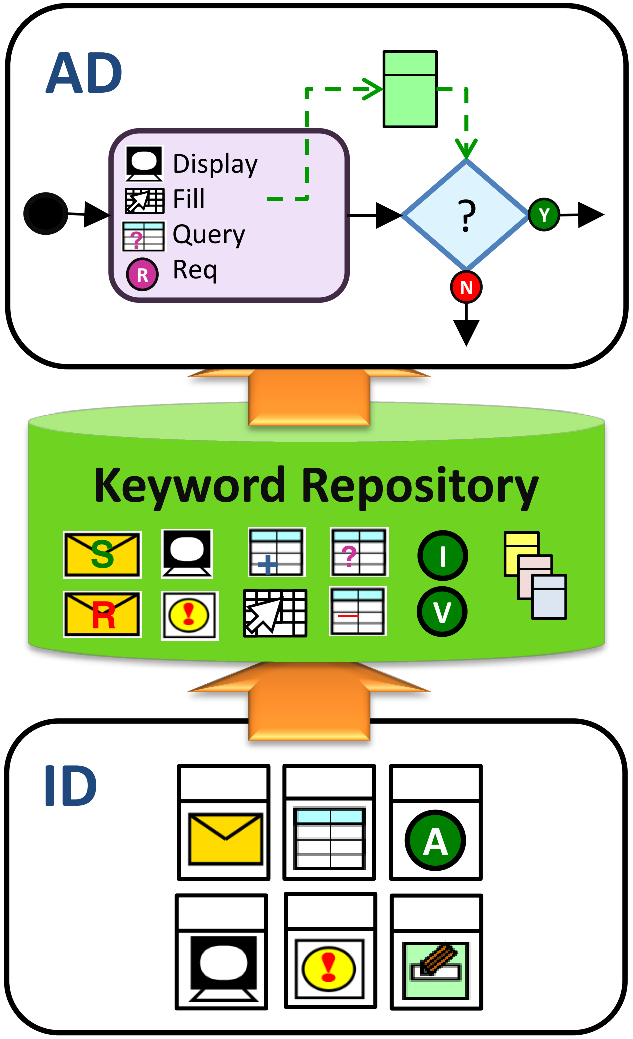
Add to shopping basket

Checkout

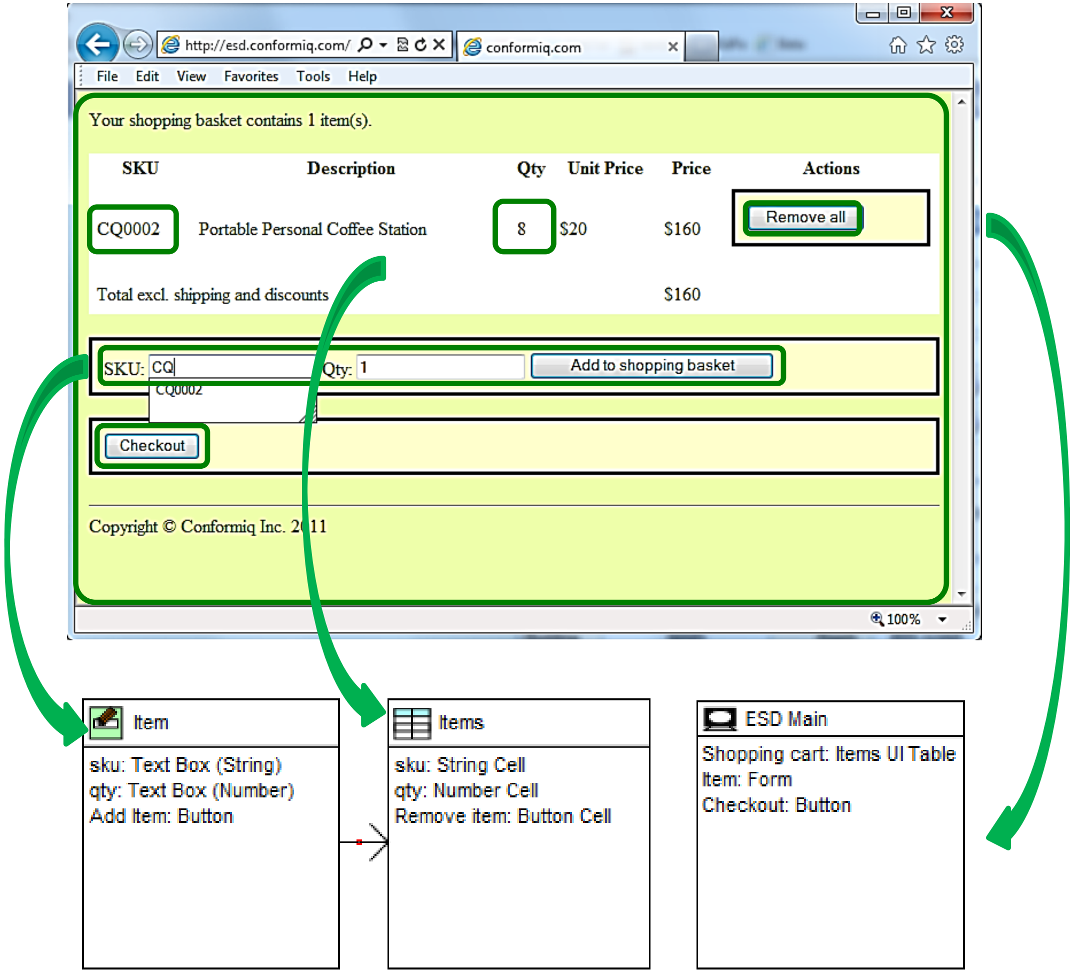
Copyright © Conformiq Inc. 2011

Creator Concepts

- Activity Diagrams
  - Flows specify specific aspects of system operation to be tested
  - Domain specific actions and data objects from keyword repository concretize activities and decisions
- Interface Diagrams
  - Specify external interfaces available for testing based on pre-defined interface objects
  - Are the source for generated actions and data objects



About Interface Diagrams



Your shopping basket contains 1 item(s).

SKU	Description	Qty	Unit Price	Price	Actions
CQ0002	Portable Personal Coffee Station	8	$20	$160	Remove all
Total excl. shipping and discounts				$160	

SKU: CQ Qty: 1 Add to shopping basket

CQ0002

Checkout

Copyright © Conformiq Inc. 2011

Item

sku: Text Box (String)
qty: Text Box (Number)
Add Item: Button

Items

sku: String Cell
qty: Number Cell
Remove item: Button Cell

ESD Main

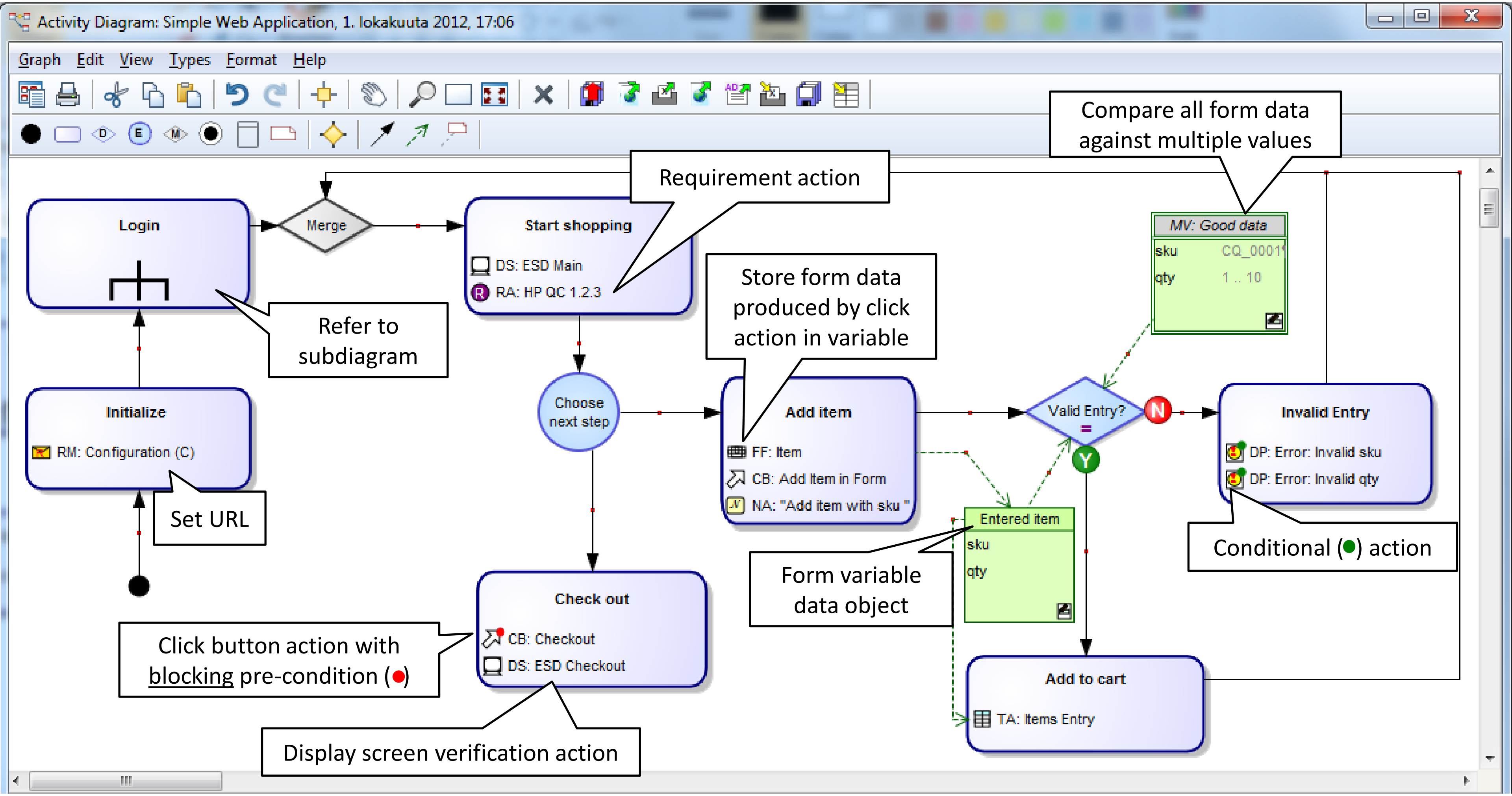
Shopping cart: Items UI Table
Item: Form
Checkout: Button

About Activity Diagrams

Fulfill a dual purpose:

- Specifies “what” is to be tested, i.e., relevant system operation, in terms of flows
  - Using standard concepts of initial, final, activity, decision, event, merge nodes and control flows
- Specifies “how” to test based on action keywords and data objects generated from interface diagrams
  - Actions from action keyword repository refine activity descriptions
  - Data objects refine (graphical) conditions

Activity Diagram Example



Compare all form data against multiple values

Requirement action

Refer to subdiagram

Store form data produced by click action in variable

Set URL

Conditional (●) action

Form variable data object

Click button action with blocking pre-condition (●)

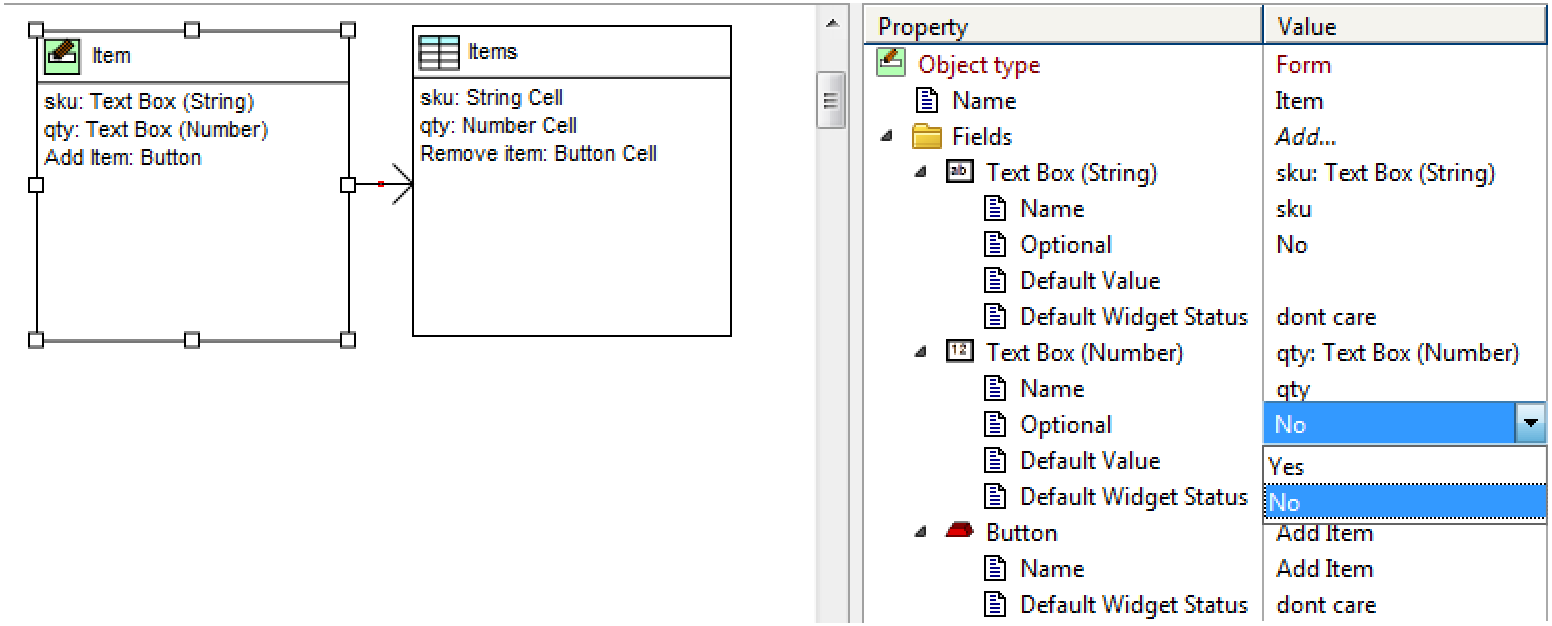
Display screen verification action

Generic vs Domain Specific

Generic Concept	Domain Specific Concept
Class	Message, Screen, Button
integer, boolean, String	Number, Checkbox, Dropdown Box
Receive on a port	Click a button, fill a form, Receive a message
Send from a port	Display a screen, Send a message
Compare each field of a variable to basic value	Compare entire message or form variable against value

Note: Domain = Application Domain and Testing Domain!

Idea: Simplify, Reduce & Reuse



Item

sku: Text Box (String)
qty: Text Box (Number)
Add Item: Button

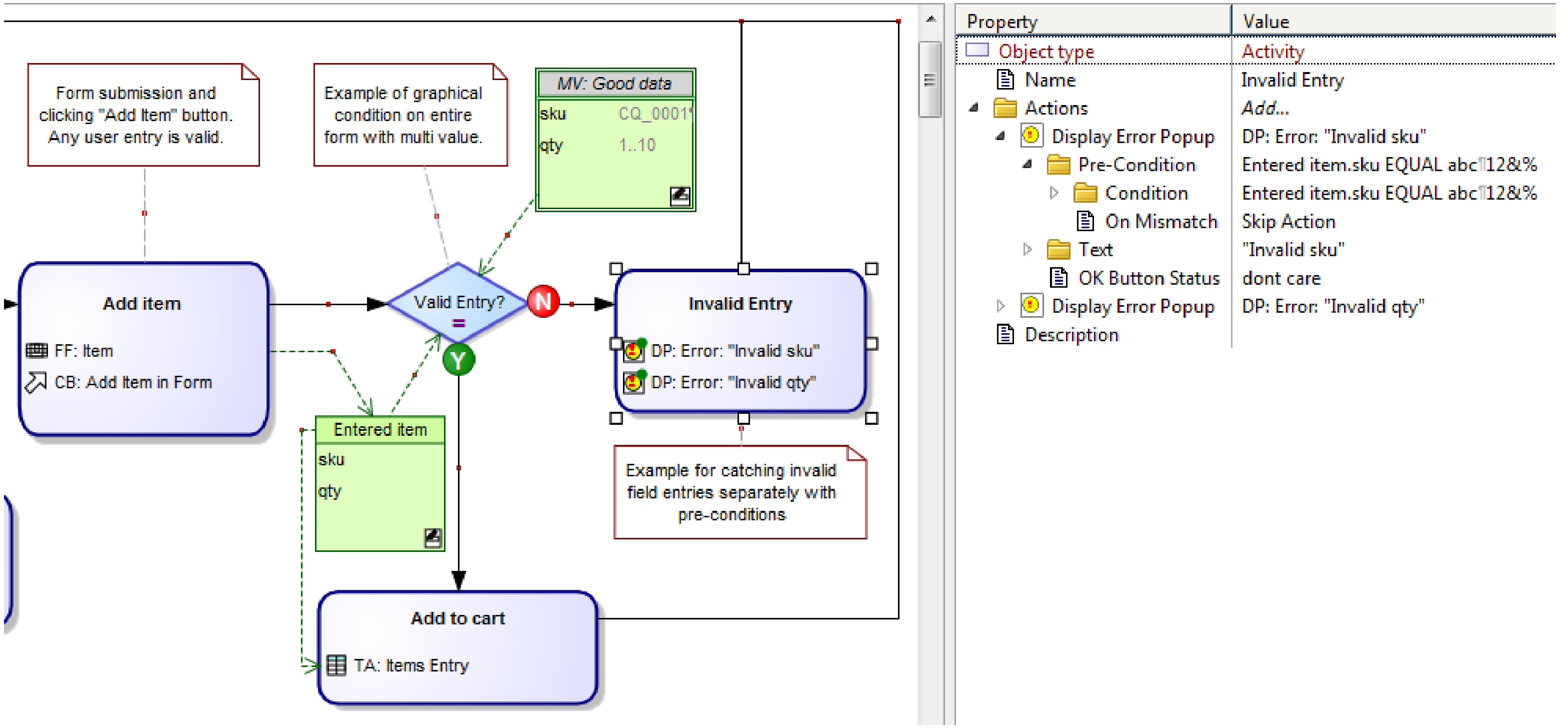
Items

sku: String Cell
qty: Number Cell
Remove item: Button Cell

Property	Value
Object type	Form
Name	Item
Fields	*Add...*
Text Box (String)	sku: Text Box (String)
Name	sku
Optional	No
Default Value	
Default Widget Status	dont care
Text Box (Number)	qty: Text Box (Number)
Name	qty
Optional	No ▾ Yes No
Default Value	
Default Widget Status	
Button	Add Item
Name	Add Item
Default Widget Status	dont care

- Symbols have look & feel closer to application domain
- Abstraction and layering of model information
- Object driven specification enables reuse
- Changes to interfaces are updated in activity diagrams
- Less modeling errors by using “specification by selection”

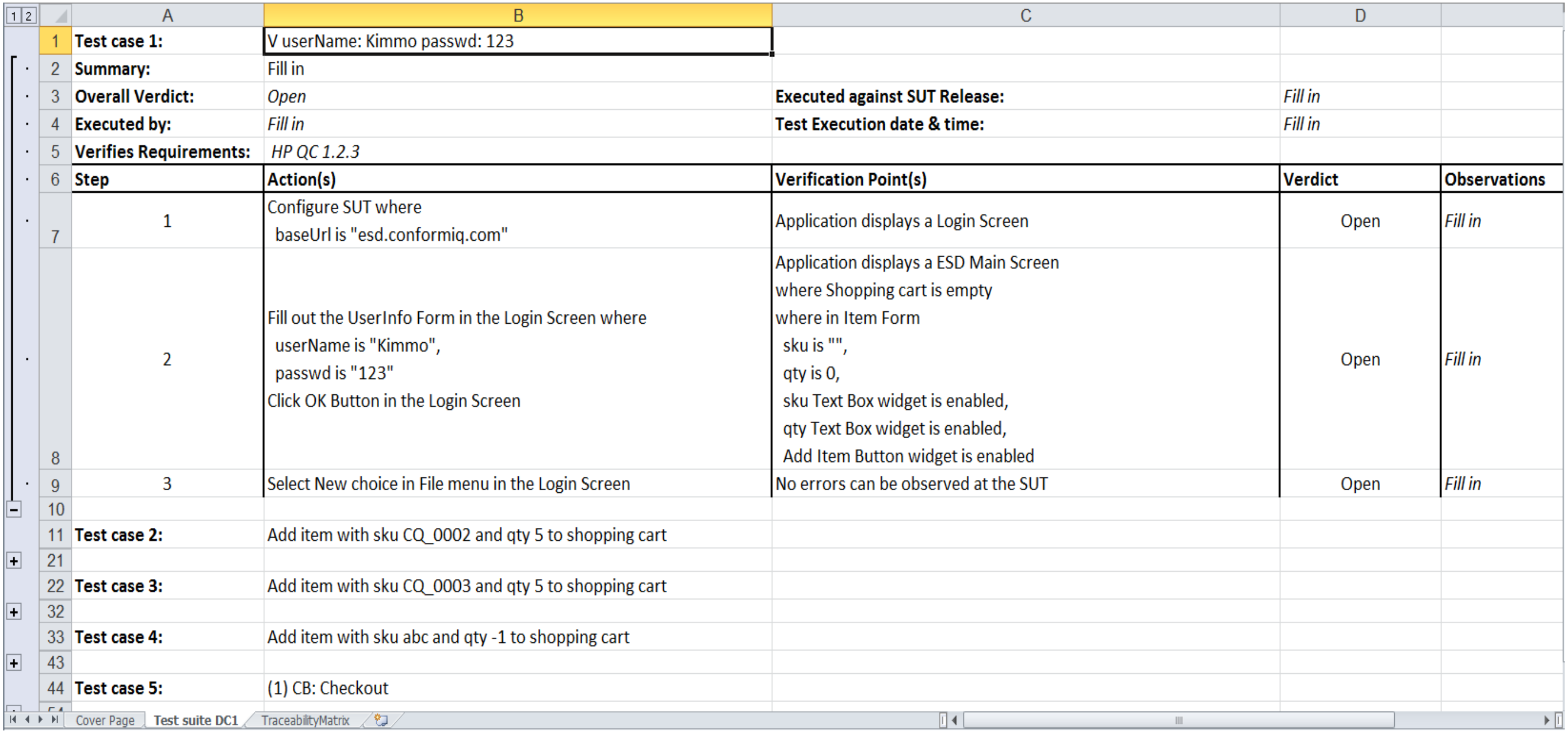
Modeling for Testing


Form submission and clicking "Add Item" button. Any user entry is valid.
Example of graphical condition on entire form with multi value.
MV: Good data
sku
CQ_0001
qty
1..10
Add item
FF: Item
CB: Add Item in Form
Valid Entry?
=
N
Y
Invalid Entry
DP: Error: "Invalid sku"
DP: Error: "Invalid qty"
Entered item
sku
qty
Example for catching invalid field entries separately with pre-conditions
Add to cart
TA: Items Entry

Property	Value
Object type	Activity
Name	Invalid Entry
Actions	*Add...*
Display Error Popup	DP: Error: "Invalid sku"
Pre-Condition	Entered item.sku EQUAL abc¶12&%
Condition	Entered item.sku EQUAL abc¶12&%
On Mismatch	Skip Action
Text	"Invalid sku"
OK Button Status	dont care
Display Error Popup	DP: Error: "Invalid qty"
Description	

- Work with complete data object values
- Enable use wildcards
- Visual indication of pre-conditions

What do Generated Tests look like?



	A	B	C	D	
1	**Test case 1:**	V userName: Kimmo passwd: 123			
2	**Summary:**	Fill in			
3	**Overall Verdict:**	*Open*	**Executed against SUT Release:**	*Fill in*	
4	**Executed by:**	*Fill in*	**Test Execution date & time:**	*Fill in*	
5	**Verifies Requirements:**	*HP QC 1.2.3*			
6	**Step**	**Action(s)**	**Verification Point(s)**	**Verdict**	**Observations**
7	1	Configure SUT where baseUrl is "esd.conformiq.com"	Application displays a Login Screen	Open	*Fill in*
8	2	Fill out the UserInfo Form in the Login Screen where userName is "Kimmo", passwd is "123" Click OK Button in the Login Screen	Application displays a ESD Main Screen where Shopping cart is empty where in Item Form sku is "", qty is 0, sku Text Box widget is enabled, qty Text Box widget is enabled, Add Item Button widget is enabled	Open	*Fill in*
9	3	Select New choice in File menu in the Login Screen	No errors can be observed at the SUT	Open	*Fill in*
10					
11	**Test case 2:**	Add item with sku CQ_0002 and qty 5 to shopping cart			
21					
22	**Test case 3:**	Add item with sku CQ_0003 and qty 5 to shopping cart			
32					
33	**Test case 4:**	Add item with sku abc and qty -1 to shopping cart			
43					
44	**Test case 5:**	(1) CB: Checkout			

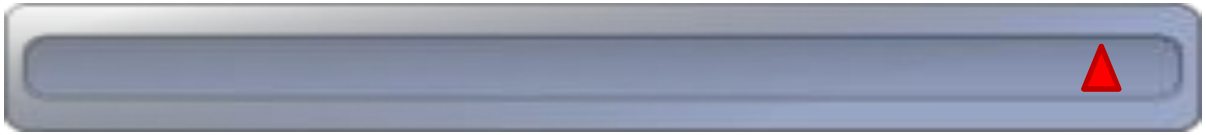
… or VB or Java or Perl or Pyton or TTCN-3 or etc

1st Industrial Feedback on Creator

- Doubled productivity over conventional UML/ Java based automated test design solution
- Training need reduced from 4 weeks to 4 days
- Subject Matter Experts (SMEs) and manual testers are able to model for testing
- Ecosystem from conventional automated test design approach could be reused

Case 2: Elektrobit Military radio

Generic

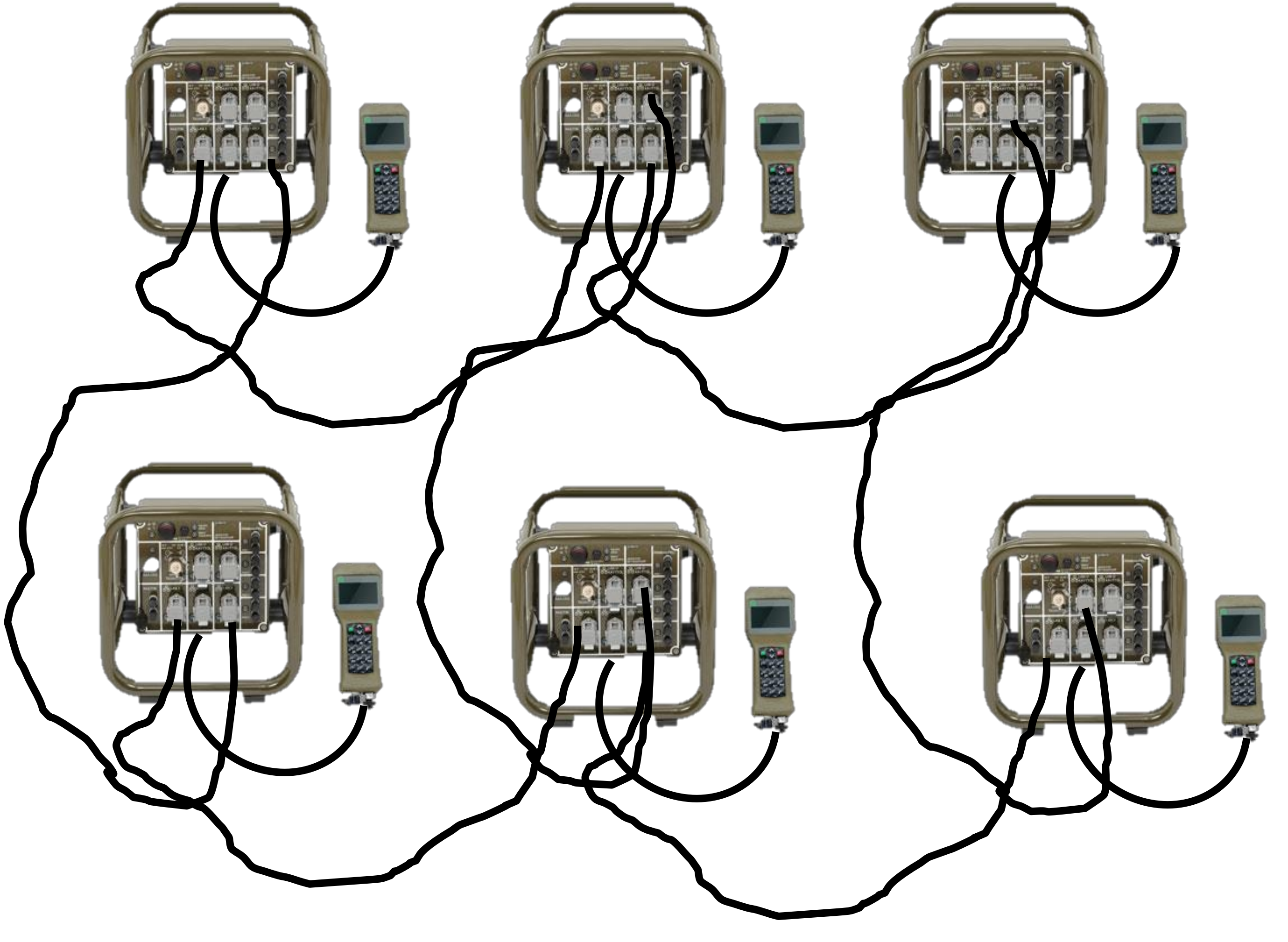


Specific

EB Tough VoIP Features

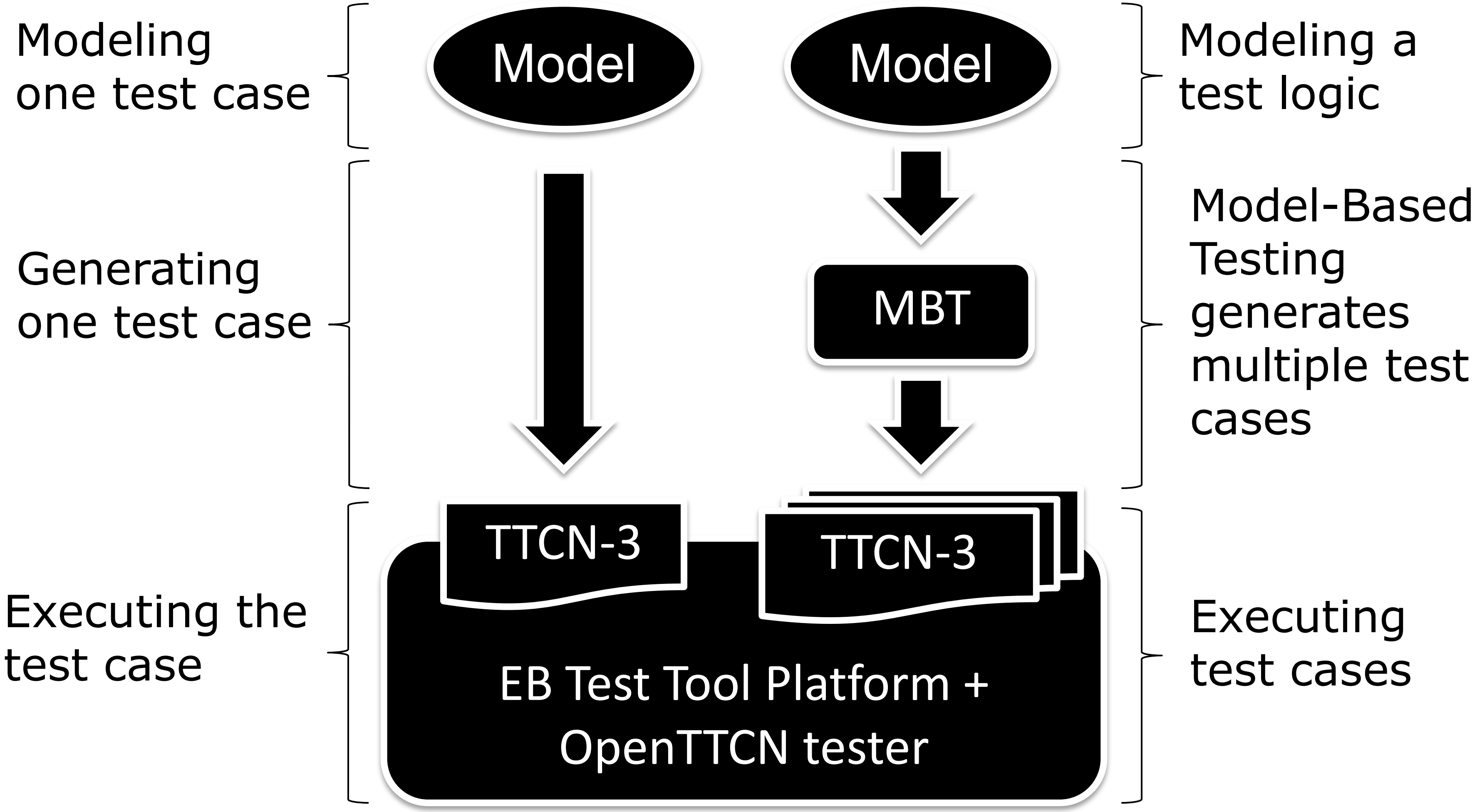
- Tough VoIP is a wired phone that is using UDP/IP network for connection
- Manufacturer: Elektrobit
- Main features:
  - Easy configuration
  - Point-to-Point call
  - All call
  - War-proof device
  - As simple as possible

Testing problem

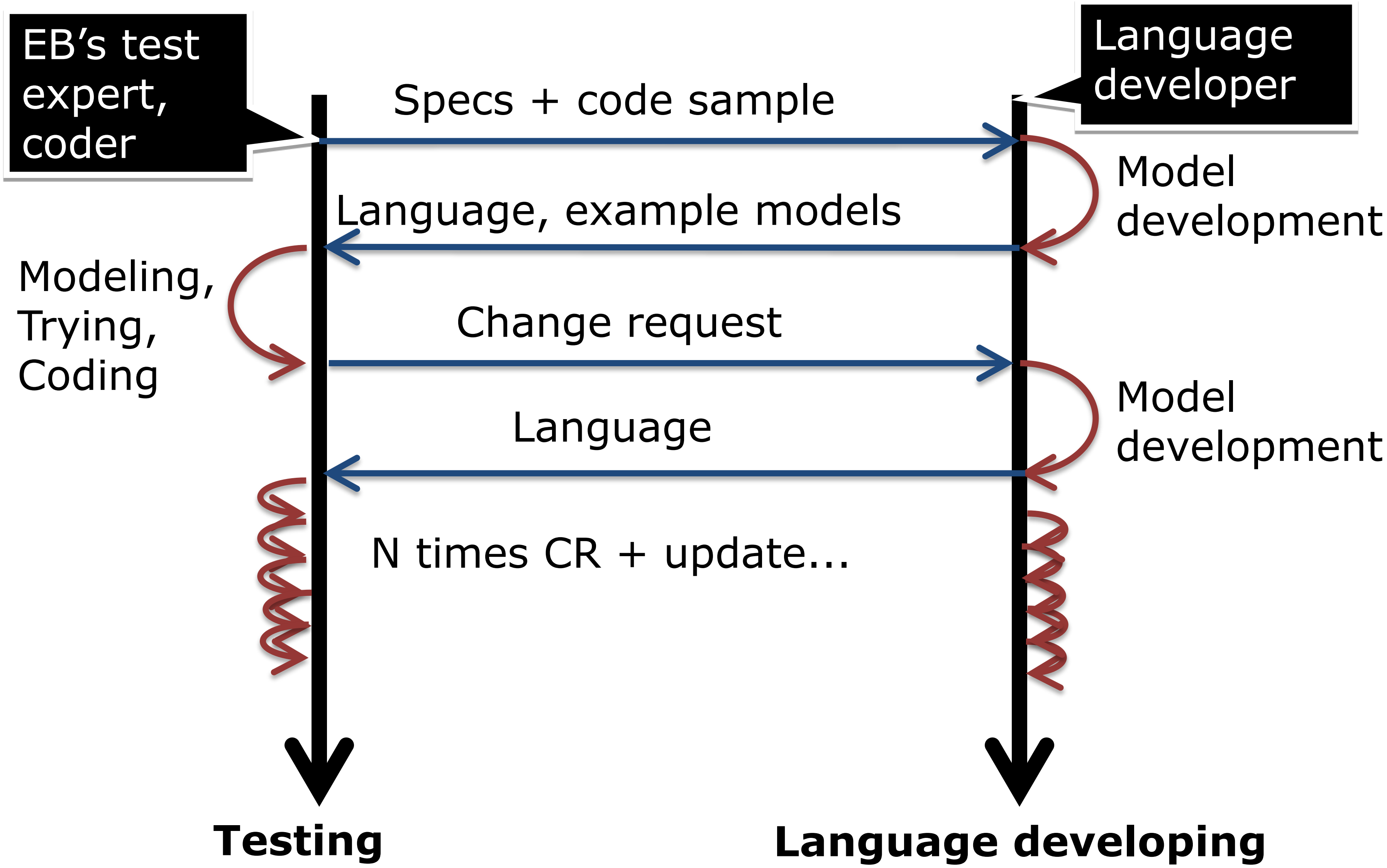


ETC...

Two language solution



Language development



EB's test expert, coder

Language developer

Specs + code sample

Model development

Language, example models

Modeling, Trying, Coding

Change request

Model development

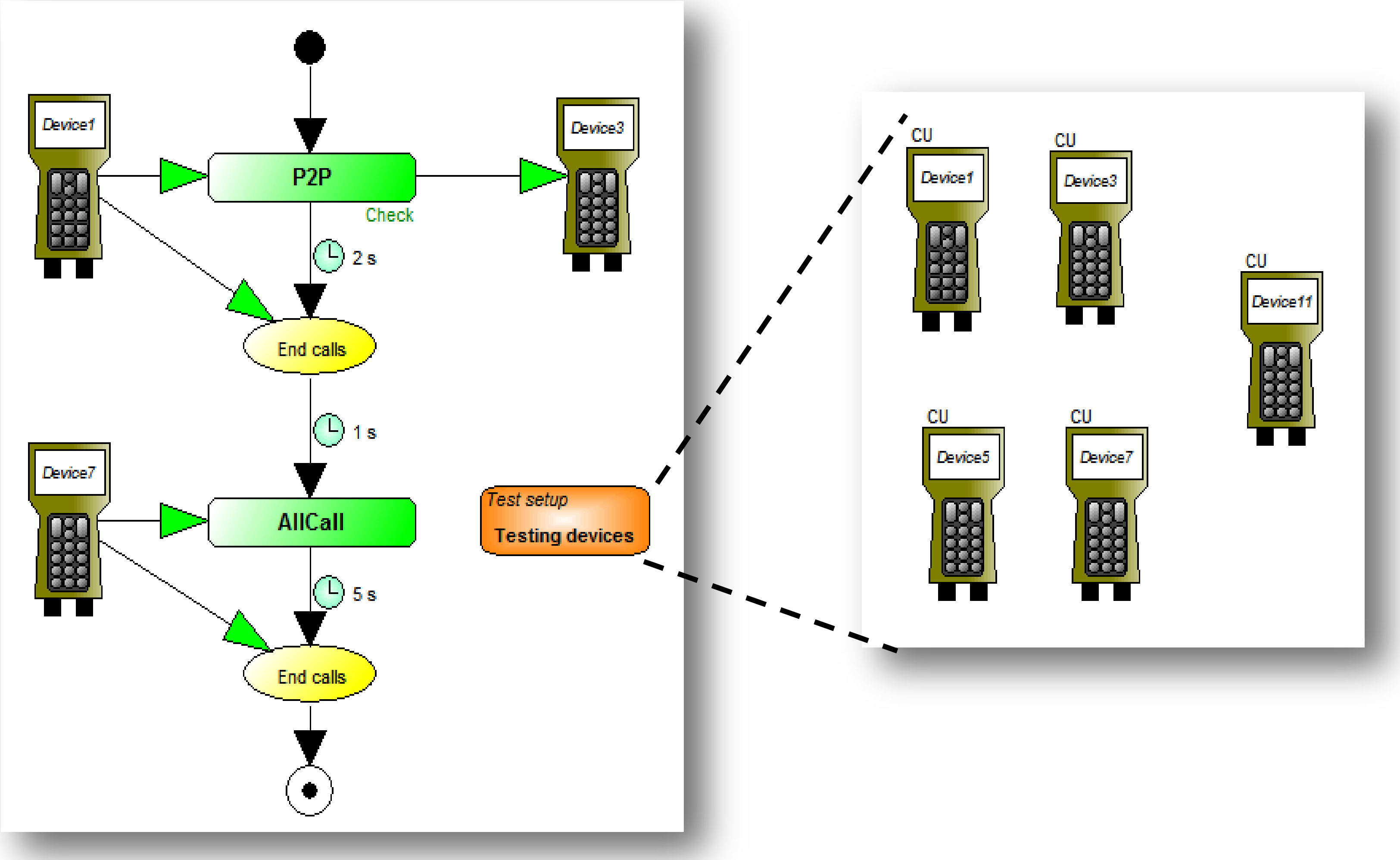
Language

N times CR + update…

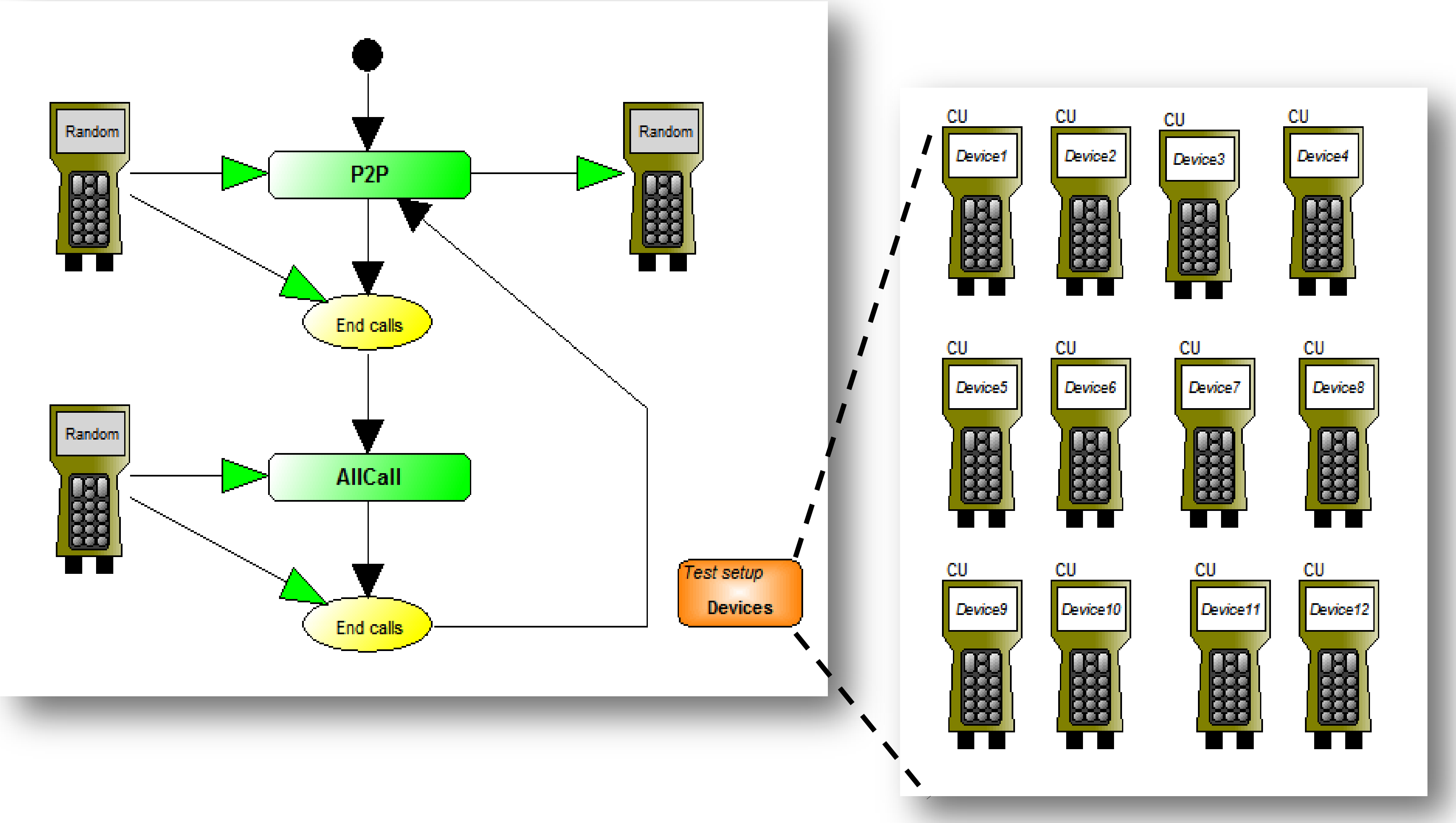
Testing

Language developing

Model example 1: Modeling test cases



Model example 2: Modeling for test generation

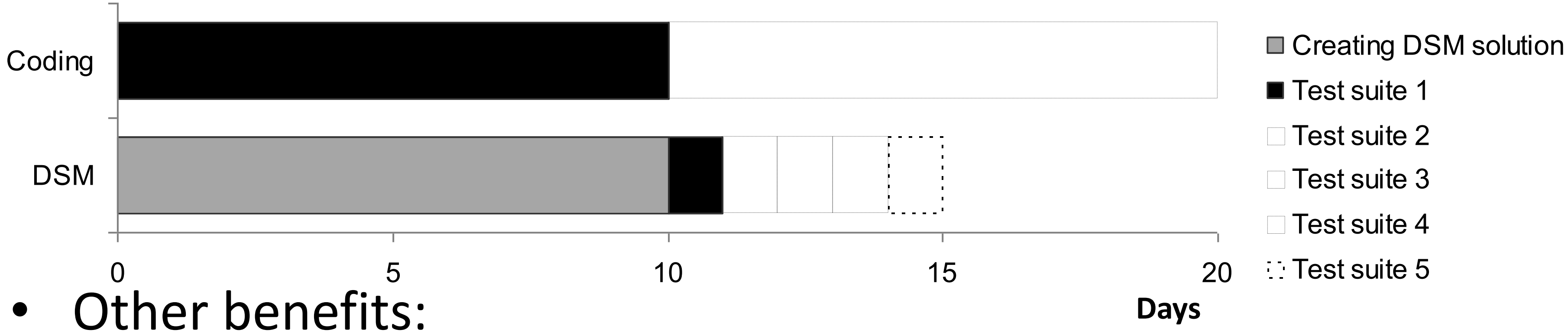


How to get started on a DSL design

- Define
  - Concepts
  - Rules
  - Symbols
  - Generators
- Focus on how you think about a problem not how you (re)solve or describe it today
  - DSLs are not effective as graphical general purpose programming languages

Experiences

- About 10 times faster with modeling
- Set-up time estimation:
  - 2 weeks for the first version
  - 1 more week for making it better



- Other benefits:
  - Visualization makes it easy to understand
  - Easy test configuration
  - Test coverage dramatically increase with MBT
  - Mass testing with MBT models
  - No special skills needed for creating test cases

Results of combining DSLs + MBT

The case studies show:

- Easier adoption
  - Better acceptance, short ramp up
- Significantly faster model development
  - Higher abstraction leads to improved productivity
  - Automation of model creation
  - Immediate feedback & guidance during model creation
- Wider model accessibility
  - Visualization makes it easier to understand
  - Domain experts can participate
  - *Customers* can review models!

Summary

- Classic DSLs benefits found to be applicable in testing
  - Driven by fully automatic model transformations
  - Prevent illegal model construction & enforce methodology
- Challenge: Keep DSL lean *and* expressive
  - Leanness yields simplicity but too lean may lead to rejection!
  - Important to use tools that enable flexibility by allowing language evolution
- We believe DSL driven MBT will establish itself as the next step in evolution of software testing

How to get started: Concepts

- What are the different object types?
  - Example: Screen, forms, widgets, messages
- What are their properties? What kind of values can they take? What is really relevant for testing?
  - Example: Dependencies between form fields? Yes
  - Example: Screen where button is located? Yes
  - Example: Pixel location of a button? No
  - Example: Underlying data base table structure? No
- What is the mapping domain concepts to concepts in the general purpose language?
  - Example: Button click maps to receiving a class

How to get started: Rules

- How many objects can exist?
  - Example: Only one starting point
- How can objects be connected?
  - Example: Only input actions can produce data
- Which property values have to be unique?
  - Example: Screen and form names
- What are valid property values?
  - Example: Only optional fields can be omitted
- When is a diagram ready for test generation?
  - Example: At least one input and verification action

How to get started: Symbols

- What type of diagrams are needed?
- Which objects are important to visualize in which diagram or at all?
  - Example: Author of a diagram
- What is the absolutely essential information important to get first understanding?
  - Example: Action has a pre-condition
- How should the information be represented?
  - Example: Symbol color, shape versus text

How to get started: Generators

- What type of information is needed to be generated?
  - Example: Code for test generation
  - Example: Model documentation
  - Example: “Live” model analysis
- In which order should objects be traversed to produce the generated code?
- How should property values be processed and converted to produce best target code?
- How to structure and modularize generator code to maximize reuse?

Thank you!

- Questions, comments, counter arguments, own experiences…

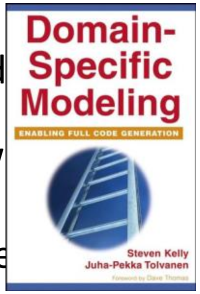
- Contact
  - Juha-Pekka Tolvanen [jpt@metacase.com]
  - www.metacase.com

MetaCase

  - Stephan Schulz [stephan.schulz@conformiq.com]
  - www.conformiq.com

CONFORMIQ
Automated Test Design™

References [1/2]



- M. Blackburn, P. Denno, Virtual Design and Verification of Cyber-Physical Systems: Ind Process Plant Design, Procedia Computer Science 28, Elsevier, 2014
- S. Kelly, J.-P. Tolvanen, “Domain-Specific Modeling: Enabling Full Code Generation”, W 2008. http://dsmbook.com
- O.-P. Puolitaival et al, “Utilizing Domain-Specific Modeling for Software Testing”, Proce of VALID, October 2011
- U. Oligschläger, “Modell-gestütztes Framework für das Testen von Mess- und Automatisierungssoftware für Prüfstände der Automobilindustrie“ [in German], GI TAV#34 Report, February 2013
- Industrial presentations and tutorials at ETSI User Conferences on MBT
  - http://www.model-based-testing.de/mbtuc11/program.html
  - http://www.elvior.com/model-based-testing-uc-2012/program
  - http://ucaat.etsi.org/2013/program.html
- MBT community http://model-based-testing.info/
- ETSI MBT Standardization
  - http://portal.etsi.org/portal/server.pt/community/MTS/323
  - MBT Modeling ES 202 951 http://pda.etsi.org/pda/queryform.asp
- *“Functional Testing Tools Are Not Enough.”*, Forrester Research Inc. Report, Testing Tools Landscape, 2010
  - Summary available via www.conformiq.com

ETSI ES 202 951 V1.1.1 (2011-07)

Methods for Testing and Specification (MTS);
Model-Based Testing (MBT);
Requirements for Modelling Notations

References [2/2]

- EADS, www.metacase.com/papers/MetaEdit_in_EADS.pdf
- NSN, Architecture in the language, www.metacase.com/cases/architectureDSMatNSN.html
- Nokia, www.metacase.com/papers/MetaEdit_in_Nokia.pdf
- Panasonic, Proceedings of Domain-Specific Modeling, 2007, www.dsmforum.org/events/DSM07/papers/safa.pdf
- Polar, Proceedings of Domain-Specific Modeling , 2009, www.dsmforum.org/events/DSM09/Papers/Karna.pdf
- USAF, ICSE, http:*//dl.acm.org/citation.cfm?id=227842*