Open-PEOPLE
Open Power and Energy Optimization
PLatform and Estimator

QAML: A Multi-Paradigm DSML for Quantitative Analysis of Embedded System Architecture Models

Dominique BLOUIN[1], Eric SENN[1], Kevin ROUSSEL[2] and Olivier ZENDRA[2]

[1]Lab-STICC, Université de Bretagne Sud, Lorient, FRANCE

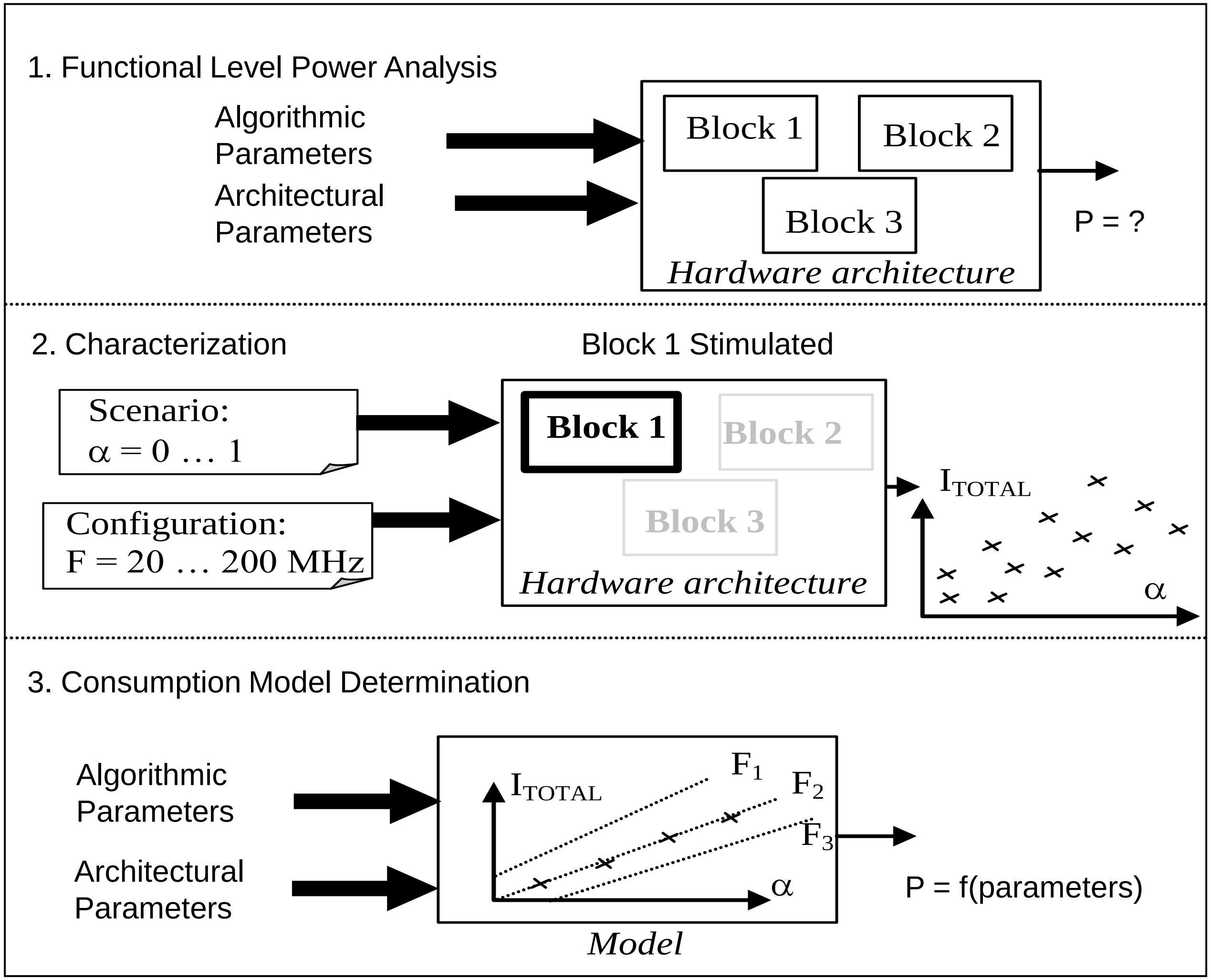
[2]Centre de recherches INRIA Nancy Grand-Est, Nancy, France

Multi-Paradigm Modeling Workshop, Innsbruck, October 1st 2012

AGENDA

- Motivations
- Language Overview
- Static Power Analysis Example.
- Conclusion and Future Work

Functional Level Power Analysis (FLPA)

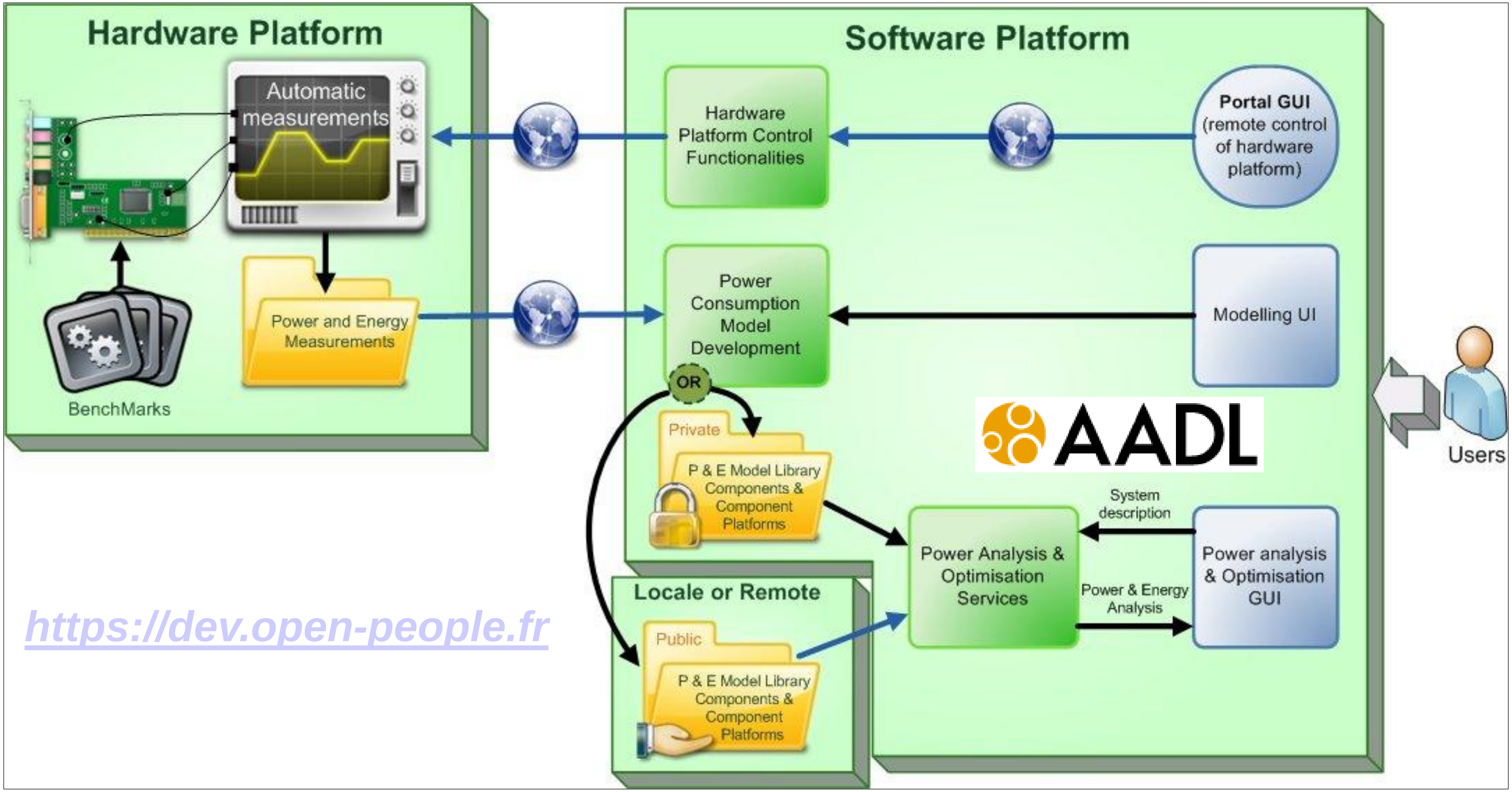


MOTIVATIONS

- Advantages:
    - Fast and accurate estimations. ☺
- Problems:
    - Reduced model applicability. ☹
        - To the specific component model on which measurements were taken.
        - There are many models….
    - Models are integrated in tools where they are expressed with general programming languages (Java, C++) .
    - Analysis tools need to be updated very often (every time a new model becomes available).
    - Analysis tools are difficult to integrate with the design.

Open-PEOPLE

Open Power and Energy Optimization PLatform and Estimator


Hardware Platform
Automatic measurements
BenchMarks
Power and Energy Measurements
Software Platform
Hardware Platform Control Functionalities
Portal GUI (remote control of hardware platform)
Power Consumption Model Development
Modelling UI
OR
Private
P & E Model Library Components & Component Platforms
AADL
Users
System description
Power Analysis & Optimisation Services
Power & Energy Analysis
Power analysis & Optimisation GUI
Locale or Remote
Public
P & E Model Library Components & Component Platforms

https://dev.open-people.fr

Solution: Formally represent analysis models with a dedicated language.

- Analysis results should be computable from the models.

- Allow analysis models to be easily integrated in model based designs.
  - Ideally by the designer.

- Analysis results automatically stored in design models (for verification).

- Provide means to ensure that the analysis results are always consistent with the design.
  - As the design changes, analysis may need to be performed again.

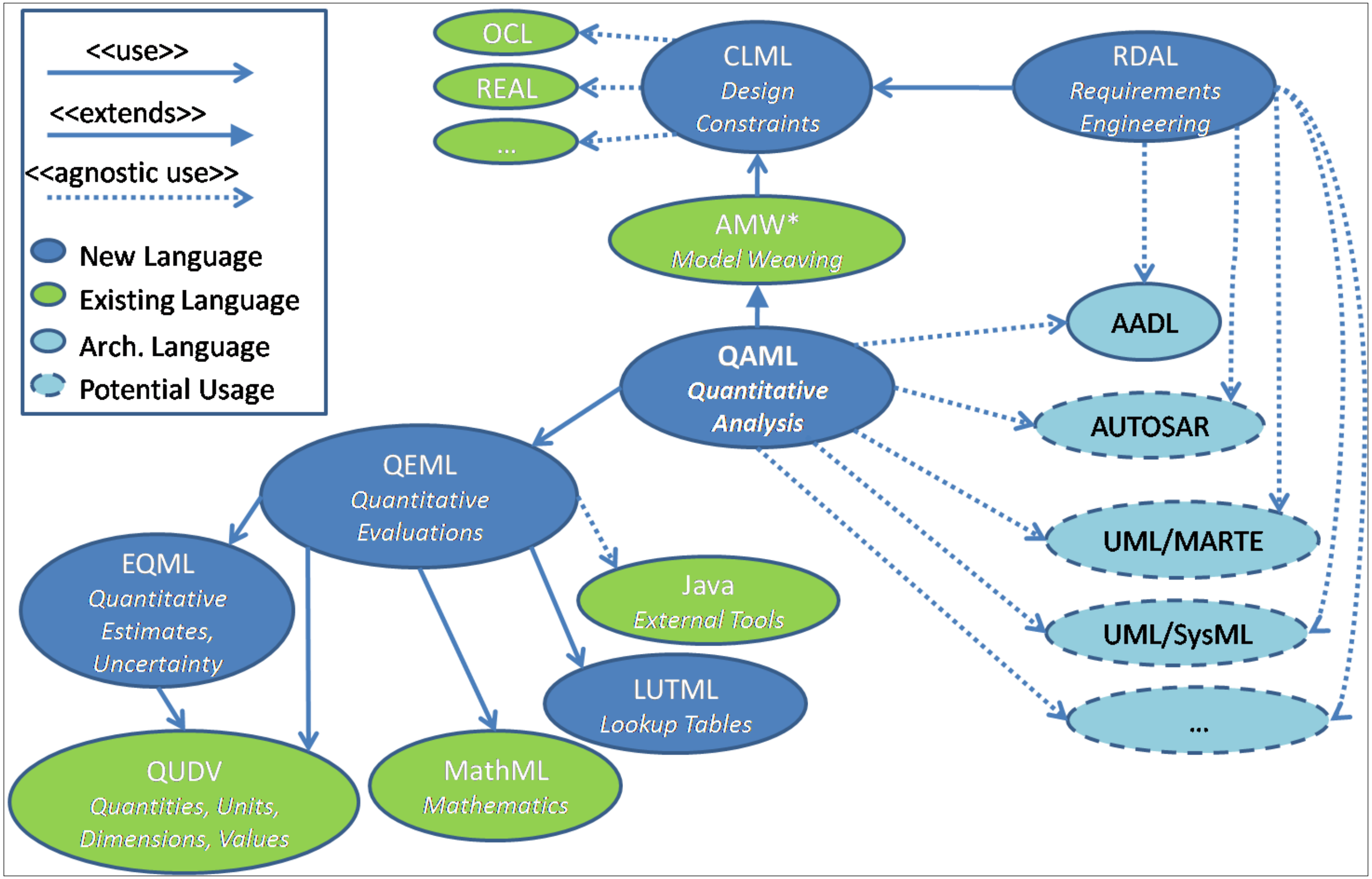
- Allow using external analysis tools.
  - Some analysis can only be expressed with computer programs.
  - Reuse legacy analysis tool.

- Do not restrict the usability to a specific ADL (reuse estimation models).

QAML (Quantitative Analysis Modeling Language)

Follow MPM principles to design the language:

- Separation of concerns.
- Develop DSMLs for appropriate domain coverage (avoid accidental complexity).

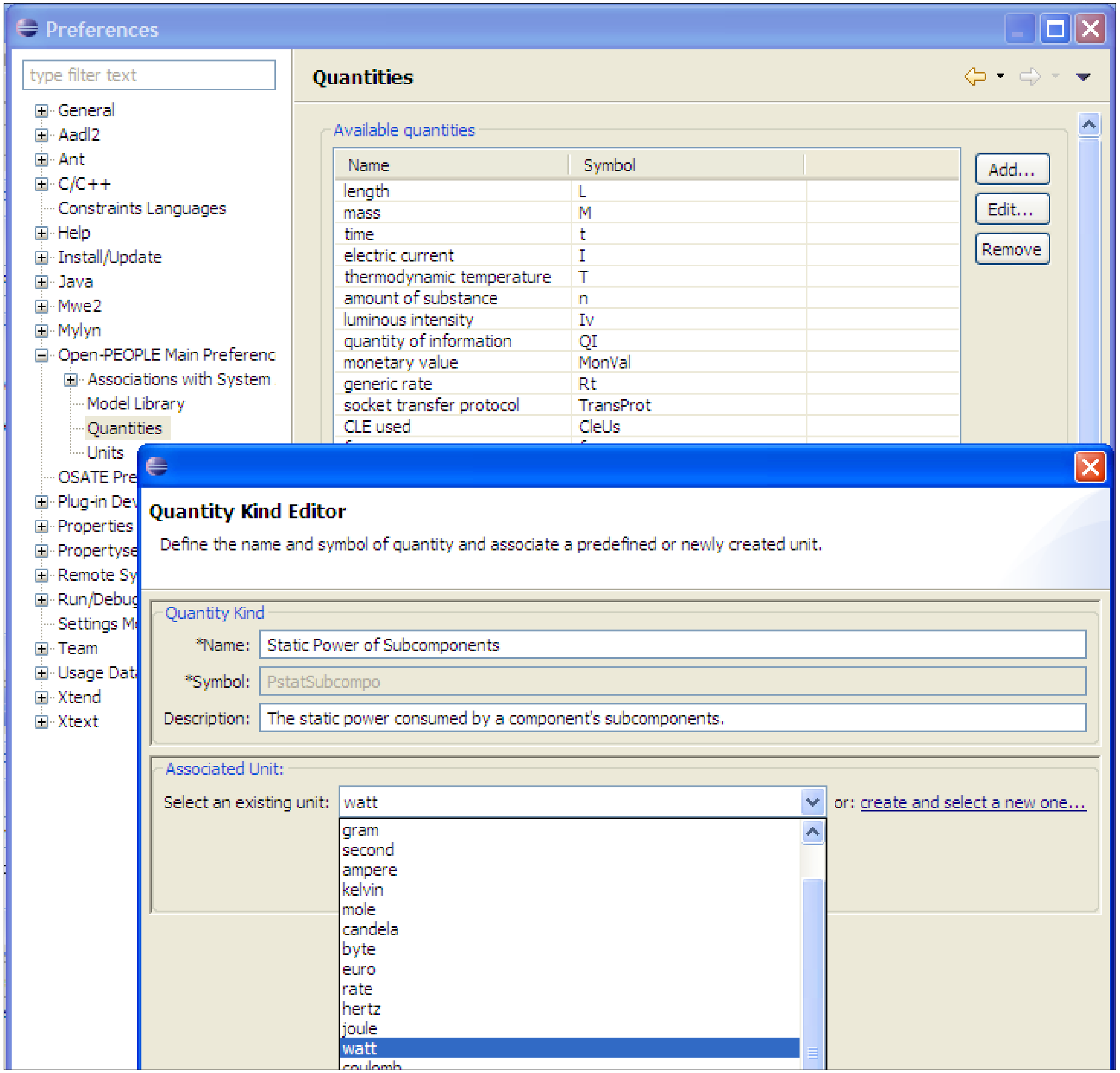


EXAMPLES: STATIC POWER ANALYSIS

Quantity Kinds and Units Definition

QUDV

Quantities, Units, Dimensions, Values



Preferences

type filter text

- General
- Aadl2
- Ant
- C/C++
- Constraints Languages
- Help
- Install/Update
- Java
- Mwe2
- Mylyn
- Open-PEOPLE Main Preferenc
  - Associations with System
  - Model Library
  - Quantities
  - Units
- OSATE Pre
- Plug-in Dev
- Properties
- Propertyse
- Remote Sy
- Run/Debug
- Settings M
- Team
- Usage Dat
- Xtend
- Xtext

Quantities

Available quantities

Name	Symbol
length	L
mass	M
time	t
electric current	I
thermodynamic temperature	T
amount of substance	n
luminous intensity	Iv
quantity of information	QI
monetary value	MonVal
generic rate	Rt
socket transfer protocol	TransProt
CLE used	CleUs

Quantity Kind Editor

Define the name and symbol of quantity and associate a predefined or newly created unit.

Quantity Kind

*Name: Static Power of Subcomponents

*Symbol: PstatSubcompo

Description: The static power consumed by a component's subcomponents.

Associated Unit:

Select an existing unit: watt

or: create and select a new one...

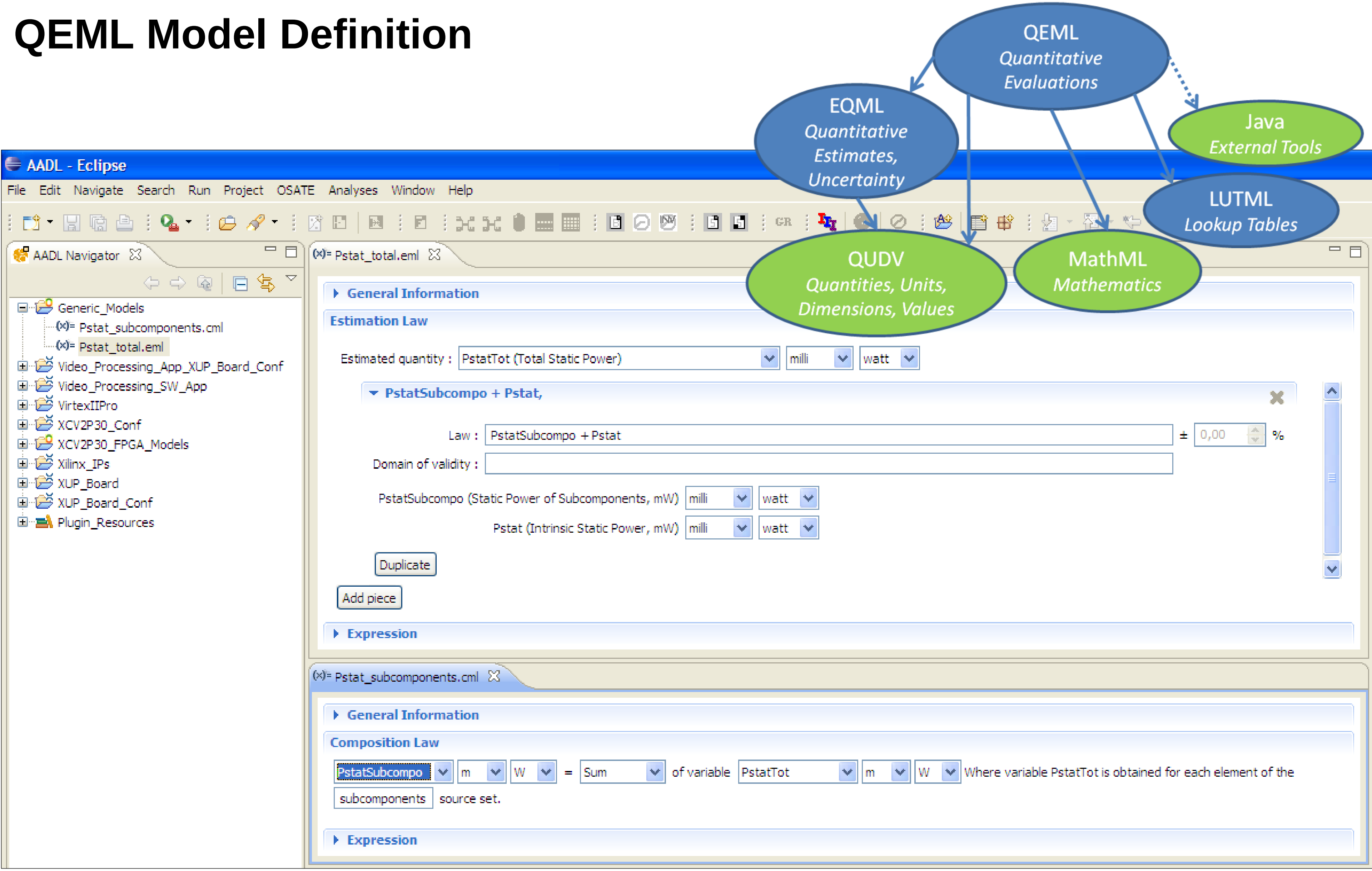
gram
second
ampere
kelvin
mole
candela
byte
euro
rate
hertz
joule
watt
coulomb

$$P_{statSubcompo} = \sum_{subcomponents} P_{statTot}$$

$$P_{statTot} = P_{statSubcompo} + P_{stat}$$

EXAMPLES: STATIC POWER ANALYSIS

QEML Model Definition



Estimation Law

Estimated quantity : PstatTot (Total Static Power) | milli | watt

PstatSubcompo + Pstat,

Law : PstatSubcompo + Pstat ± 0,00 %

Domain of validity :

PstatSubcompo (Static Power of Subcomponents, mW) milli watt

Pstat (Intrinsic Static Power, mW) milli watt

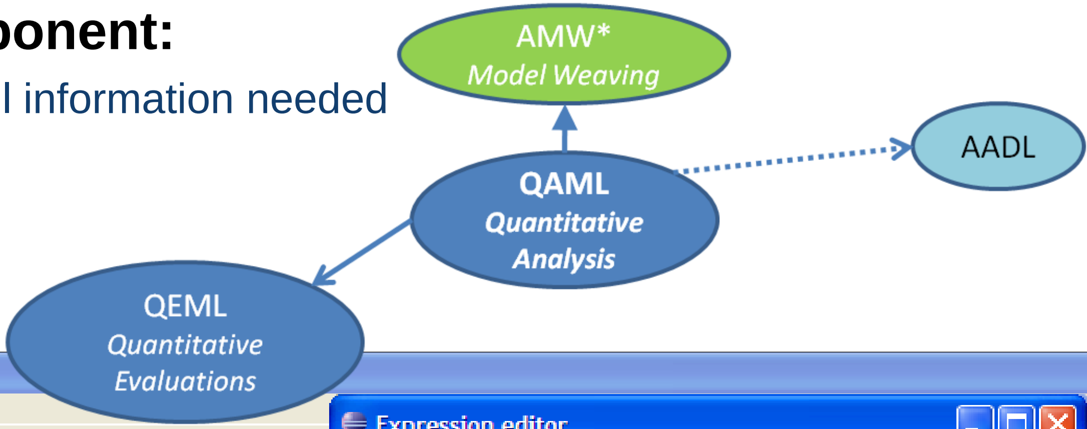
Composition Law

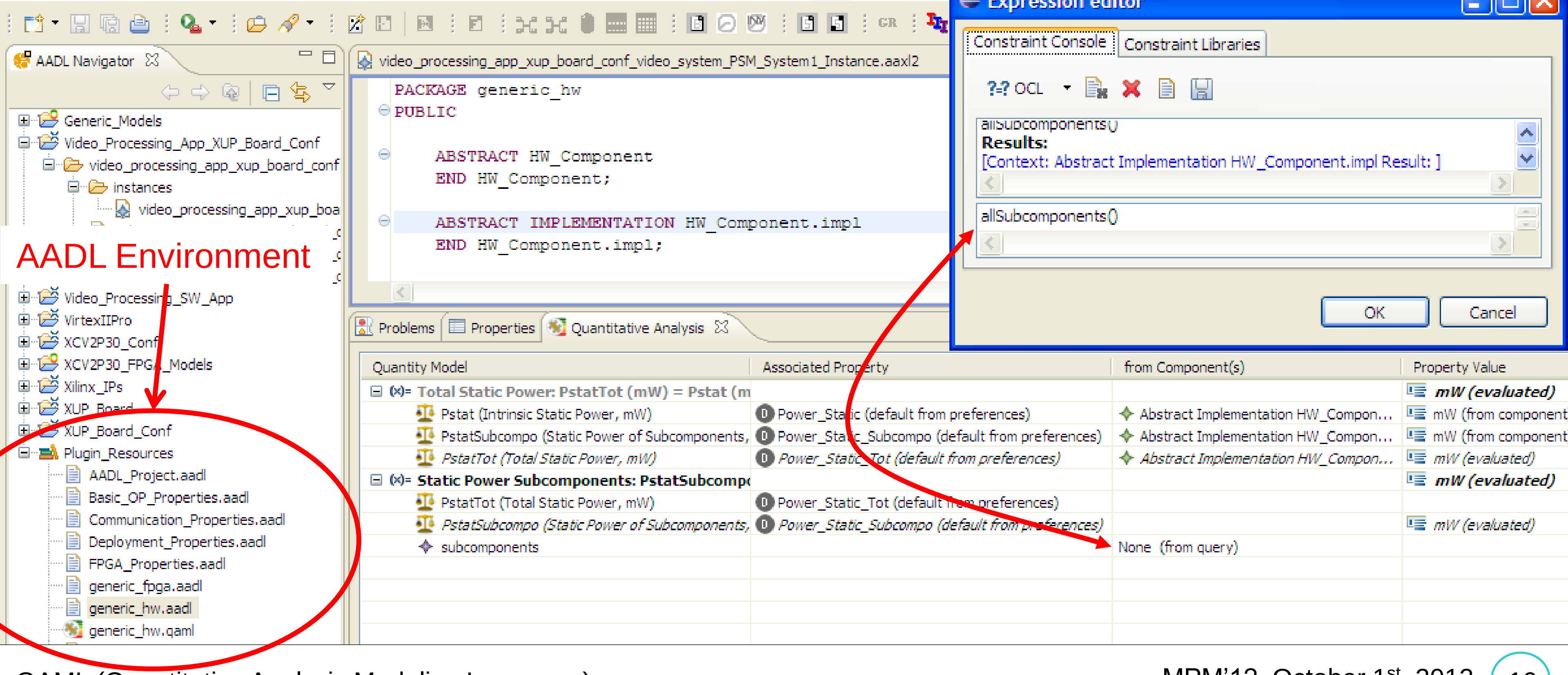
PstatSubcompo m W = Sum of variable PstatTot m W Where variable PstatTot is obtained for each element of the subcomponents source set.

EXAMPLES: STATIC POWER ANALYSIS

Association with an AADL Component:

- At this high level of abstraction **not** all information needed to compute the model is available.





Generic_Models
Video_Processing_App_XUP_Board_Conf
video_processing_app_xup_board_conf
instances
video_processing_app_xup_boa
Video_Processing_SW_App
VirtexIIPro
XCV2P30_Conf
XCV2P30_FPGA_Models
Xilinx_IPs
XUP_Board
XUP_Board_Conf
Plugin_Resources
AADL_Project.aadl
Basic_OP_Properties.aadl
Communication_Properties.aadl
Deployment_Properties.aadl
FPGA_Properties.aadl
generic_fpga.aadl
generic_hw.aadl
generic_hw.qaml

video_processing_app_xup_board_conf_video_system_PSM_System1_Instance.aaxl2

```
PACKAGE generic_hw
PUBLIC

    ABSTRACT HW_Component
    END HW_Component;

    ABSTRACT IMPLEMENTATION HW_Component.impl
    END HW_Component.impl;
```

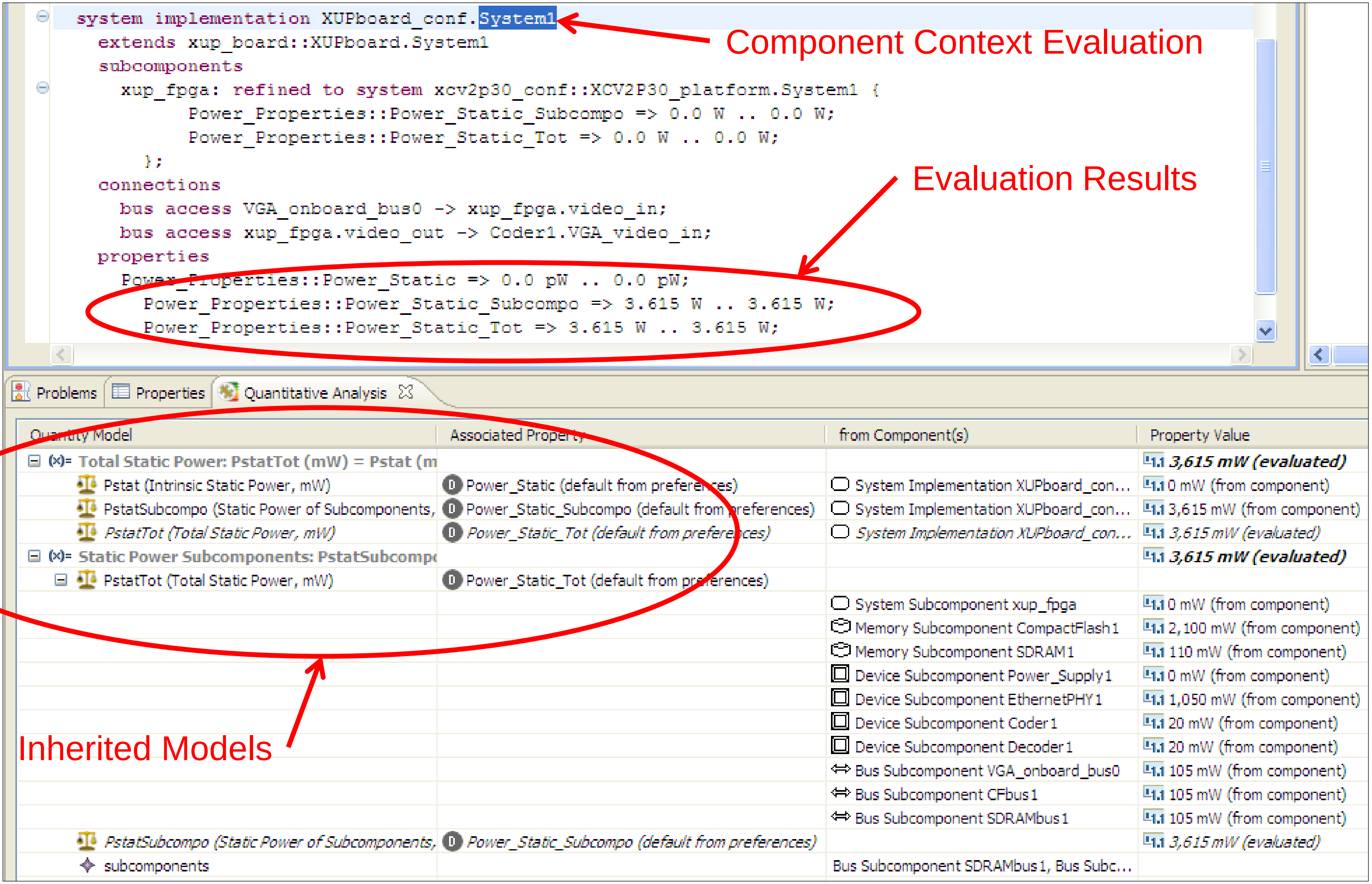
```
allSubcomponents()
Results:
[Context: Abstract Implementation HW_Component.impl Result: ]
```

```
allSubcomponents()
```

Quantity Model	Associated Property	from Component(s)	Property Value
(x)= **Total Static Power: PstatTot (mW) = Pstat (m**			***mW (evaluated)***
Pstat (Intrinsic Static Power, mW)	Power_Static (default from preferences)	Abstract Implementation HW_Compon...	mW (from component
PstatSubcompo (Static Power of Subcomponents,	Power_Static_Subcompo (default from preferences)	Abstract Implementation HW_Compon...	mW (from component
PstatTot (Total Static Power, mW)	*Power_Static_Tot (default from preferences)*	*Abstract Implementation HW_Compon...*	*mW (evaluated)*
(x)= **Static Power Subcomponents: PstatSubcompo**			***mW (evaluated)***
PstatTot (Total Static Power, mW)	Power_Static_Tot (default from preferences)		
PstatSubcompo (Static Power of Subcomponents,	*Power_Static_Subcompo (default from preferences)*		*mW (evaluated)*
subcomponents		None (from query)	

Evaluable inherited QAML Model

- At this lower level of abstraction **all** information needed to compute the model is available.



```
system implementation XUPboard_conf.System1
  extends xup_board::XUPboard.System1
  subcomponents
    xup_fpga: refined to system xcv2p30_conf::XCV2P30_platform.System1 {
          Power_Properties::Power_Static_Subcompo => 0.0 W .. 0.0 W;
          Power_Properties::Power_Static_Tot => 0.0 W .. 0.0 W;
      };
  connections
    bus access VGA_onboard_bus0 -> xup_fpga.video_in;
    bus access xup_fpga.video_out -> Coder1.VGA_video_in;
  properties
    Power_Properties::Power_Static => 0.0 pW .. 0.0 pW;
      Power_Properties::Power_Static_Subcompo => 3.615 W .. 3.615 W;
      Power_Properties::Power_Static_Tot => 3.615 W .. 3.615 W;
```

Component Context Evaluation

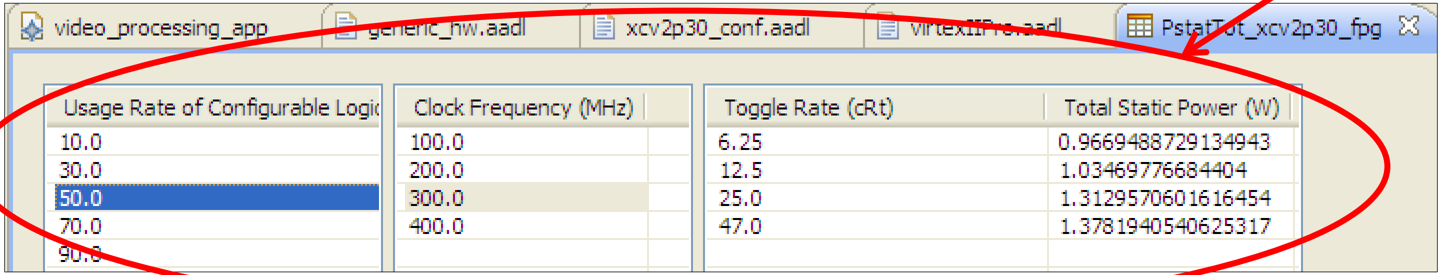
Evaluation Results

Quantity Model	Associated Property	from Component(s)	Property Value
⊟ (x)= Total Static Power: PstatTot (mW) = Pstat (m			*3,615 mW (evaluated)*
Pstat (Intrinsic Static Power, mW)	Power_Static (default from preferences)	System Implementation XUPboard_con...	0 mW (from component)
PstatSubcompo (Static Power of Subcomponents,	Power_Static_Subcompo (default from preferences)	System Implementation XUPboard_con...	3,615 mW (from component)
PstatTot (Total Static Power, mW)	*Power_Static_Tot (default from preferences)*	*System Implementation XUPboard_con...*	*3,615 mW (evaluated)*
⊟ (x)= Static Power Subcomponents: PstatSubcompo			*3,615 mW (evaluated)*
⊟ PstatTot (Total Static Power, mW)	Power_Static_Tot (default from preferences)		
		System Subcomponent xup_fpga	0 mW (from component)
		Memory Subcomponent CompactFlash1	2,100 mW (from component)
		Memory Subcomponent SDRAM1	110 mW (from component)
		Device Subcomponent Power_Supply1	0 mW (from component)
		Device Subcomponent EthernetPHY1	1,050 mW (from component)
		Device Subcomponent Coder1	20 mW (from component)
		Device Subcomponent Decoder1	20 mW (from component)
		Bus Subcomponent VGA_onboard_bus0	105 mW (from component)
		Bus Subcomponent CFbus1	105 mW (from component)
		Bus Subcomponent SDRAMbus1	105 mW (from component)
PstatSubcompo (Static Power of Subcomponents,	*Power_Static_Subcompo (default from preferences)*		*3,615 mW (evaluated)*
subcomponents		Bus Subcomponent SDRAMbus1, Bus Subc...	

Inherited Models

FPGA Lookup Table Based Model

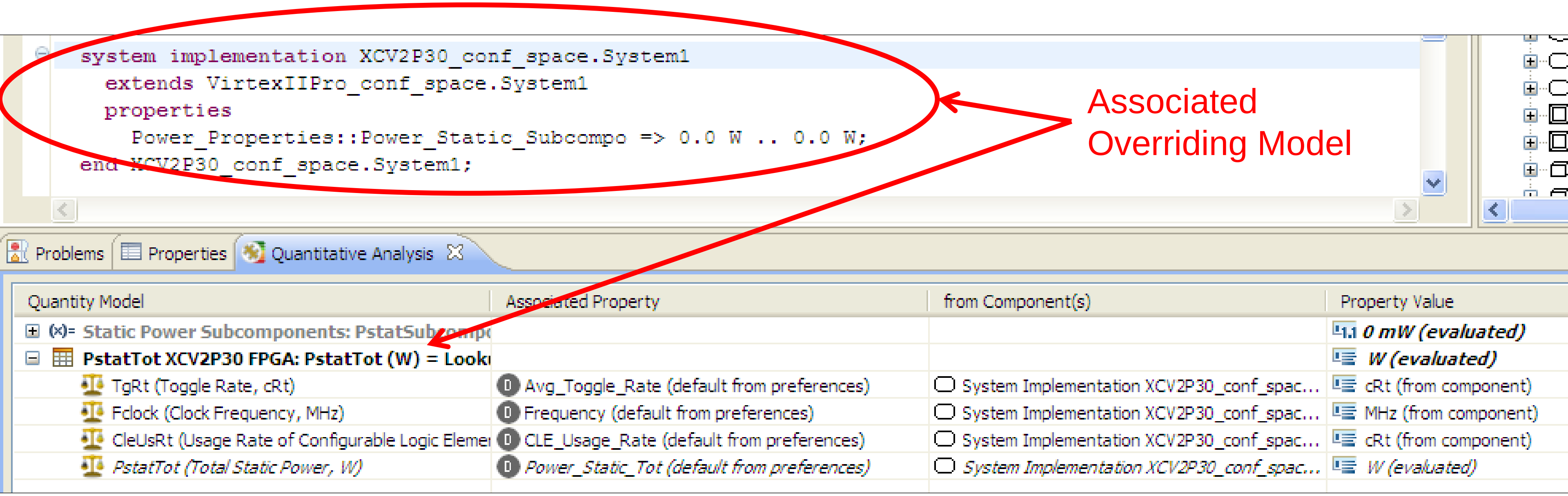
Multi-dimension data table



Usage Rate of Configurable Logic
10.0
30.0
50.0
70.0
90.0

Clock Frequency (MHz)
100.0
200.0
300.0
400.0

Toggle Rate (cRt)	Total Static Power (W)
6.25	0.9669488729134943
12.5	1.03469776684404
25.0	1.31295706016454
47.0	1.3781940540625317



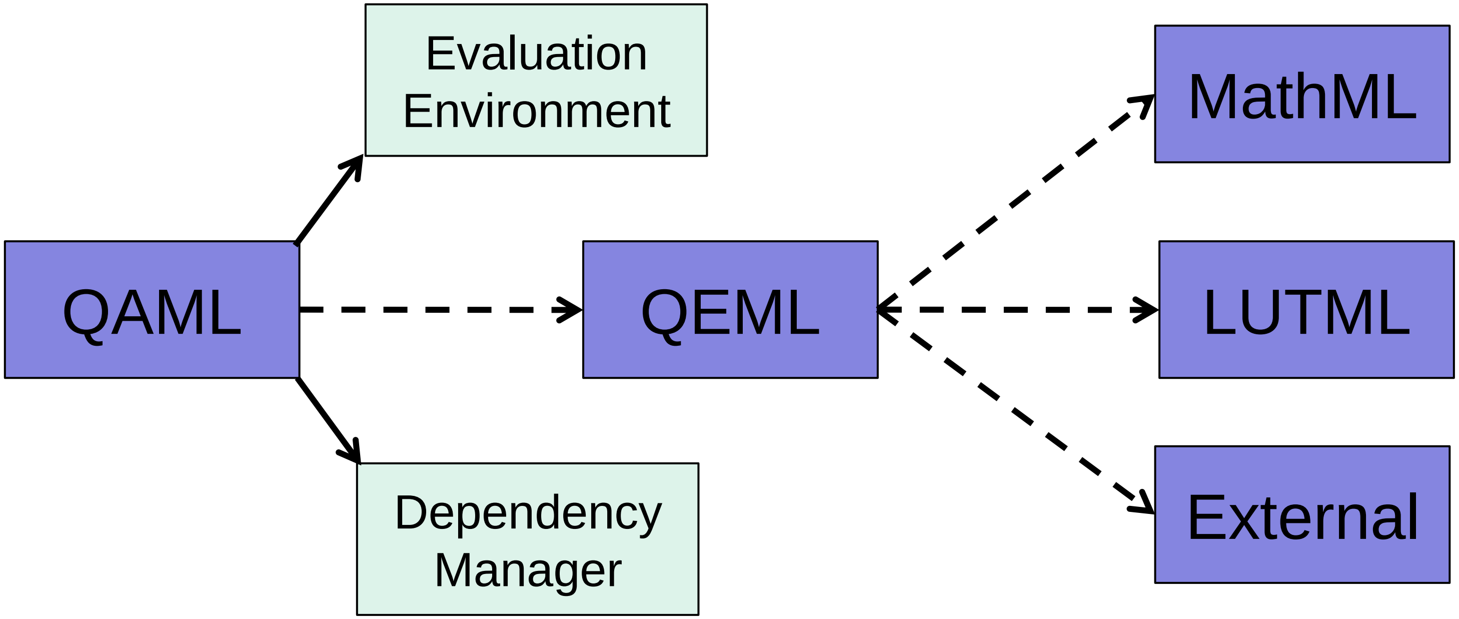
```
system implementation XCV2P30_conf_space.System1
  extends VirtexIIPro_conf_space.System1
  properties
    Power_Properties::Power_Static_Subcompo => 0.0 W .. 0.0 W;
end XCV2P30_conf_space.System1;
```

Associated Overriding Model

Quantity Model	Associated Property	from Component(s)	Property Value
⊞ (x)= Static Power Subcomponents: PstatSubcompo			0 mW (evaluated)
⊟ PstatTot XCV2P30 FPGA: PstatTot (W) = Looku			W (evaluated)
TgRt (Toggle Rate, cRt)	Avg_Toggle_Rate (default from preferences)	System Implementation XCV2P30_conf_spac...	cRt (from component)
Fclock (Clock Frequency, MHz)	Frequency (default from preferences)	System Implementation XCV2P30_conf_spac...	MHz (from component)
CleUsRt (Usage Rate of Configurable Logic Eleme	CLE_Usage_Rate (default from preferences)	System Implementation XCV2P30_conf_spac...	cRt (from component)
PstatTot (Total Static Power, W)	*Power_Static_Tot (default from preferences)*	*System Implementation XCV2P30_conf_spac...*	*W (evaluated)*

SEMANTICS

- Semantics composed from the semantics of the sub-languages.
- QAML is an interpreted language.
  - Interpreter developed in Java.
- Composite interpreter calling sub languages interpreters.
  - Sub-languages and their interpreters can be reused independently.



CONCLUSION

- MPM methodology provided nice language architecture favoring separation of concerns and reuse of existing DSMLs.
  - Saved tremendous modeling efforts and ensured proper domain coverage.
- QAML can be used for many analyses besides power consumption:
  - Equation based model.
  - Simulation results (LUT).
  - Integration of legacy analysis tools.
- Analysis results maintained consistent with design models.
- Opportunity to represent *formal* components data sheets.

FUTURE WORK

- Improve inter model consistency (what to do when refferred model elements are deleted, changed, etc).
- Use QAML for other analyses than power consumption.
- Extend QAML for non numerical properties.
- Use QAML with other ADLs (Autosar, SysML, etc.).
- Tool available from https://dev.open-people.fr/wiki/OPSWP-Release
- Next releases should include:
  - QAML automated evaluation.
  - Uncertainty management.
  - External tool evaluation descriptor.