



**pure::variants**



pure-systems

**pure::variants Synchronizer**  
**for Borland® CaliberRM**

# pure::variants – The tool



## pure::variants

- **Integration**

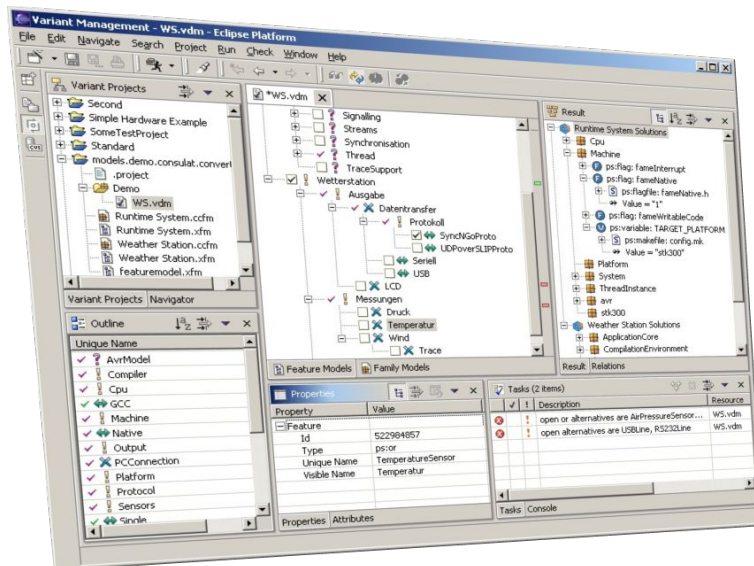
can be integrated into existing development processes  
independently of the programming language used

- **Uses**

automated resolution of relation conflicts  
user-specific views on variability models

- **Benefits**

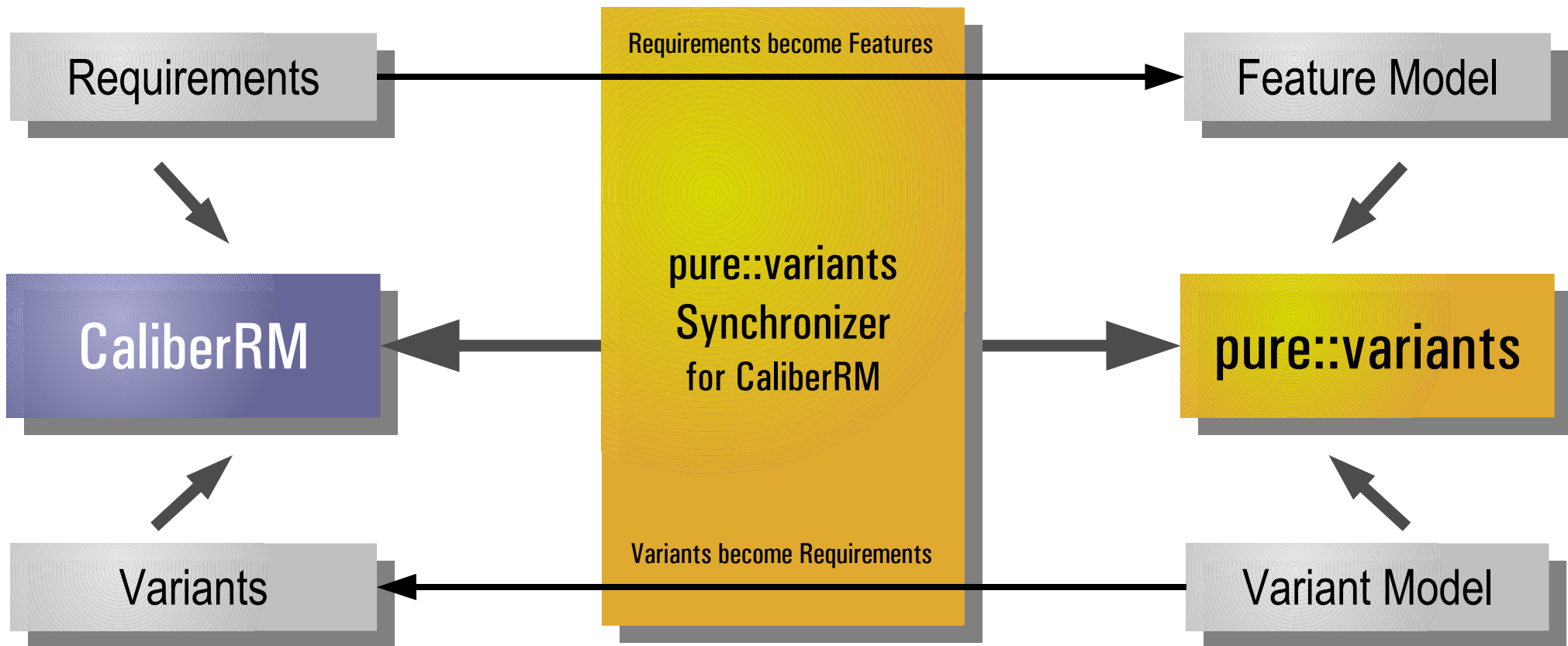
improved complexity management  
all benefits of product line based development



# The Challenge

- Extension of existing requirements management tools for dealing with variants and variability information:
  - efficiently producing answers for questions like:
    - Which requirement combinations are permitted?
    - Which requirement combinations are used in which product variants? Are there *unused* requirements?
  - applicable to existing requirements data

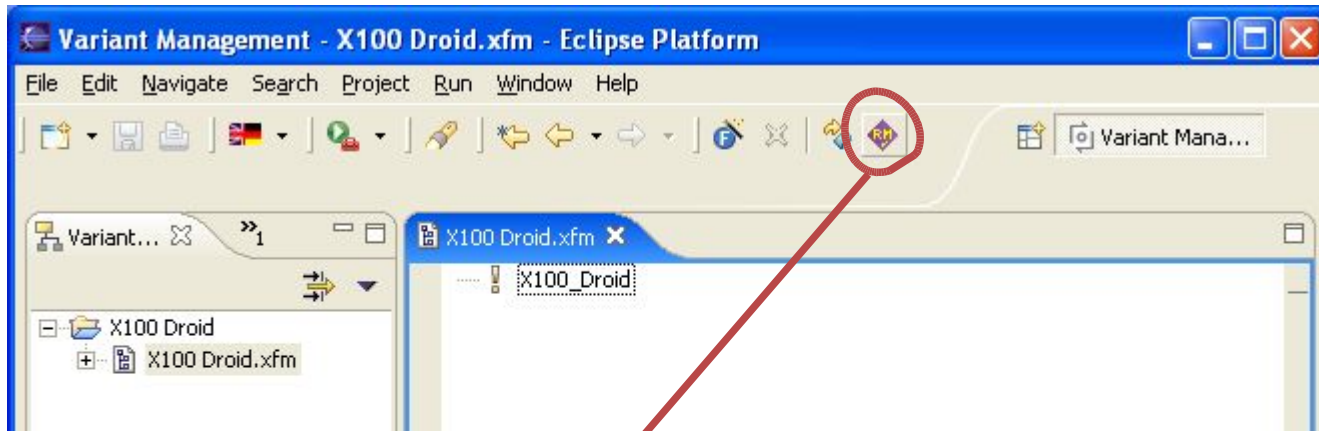
# Our Answer



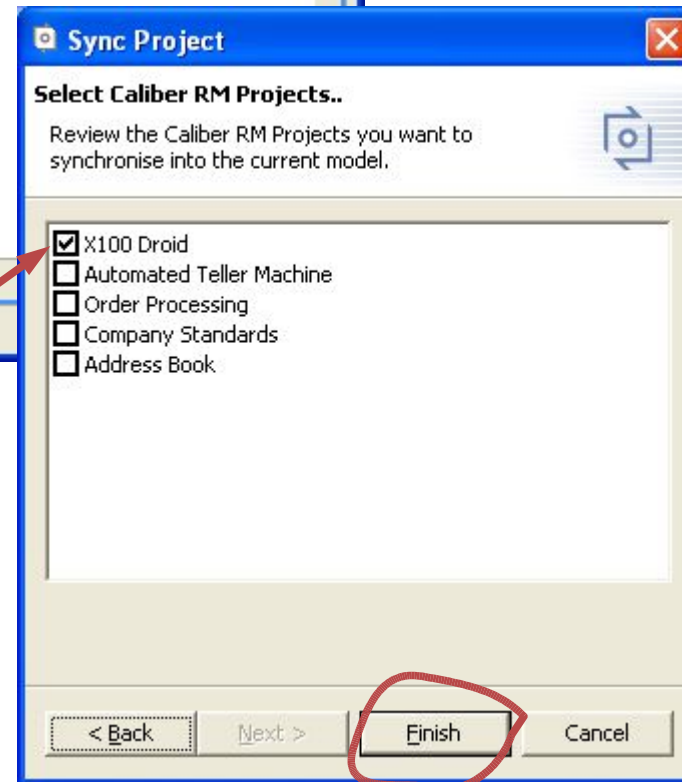
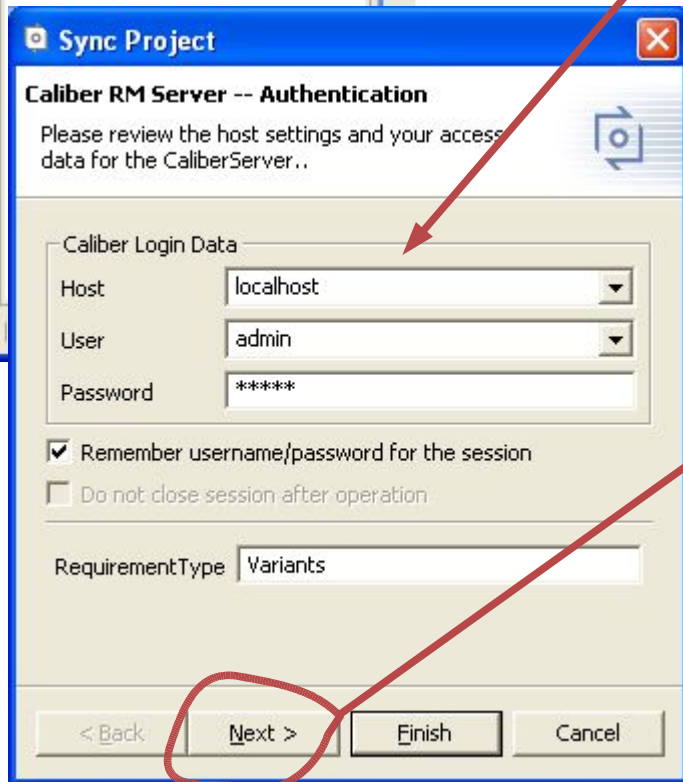
# pure::variants Synchronizer for CaliberRM

- Transforms CaliberRM projects into pure::variants feature models
  - Updates to feature models possible at any time without loss of information
- Uses variability modeling capabilities of pure::variants
  - Powerful rule language
  - Conflict detection and resolution
- Transforms pure::variants Variant Models into CaliberRM requirements.
- Permits direct integration of variable requirements into software configuration process.

# Step 1: Data import from CaliberRM



The transformation of CaliberRM projects is controlled by a pure::variants wizard.

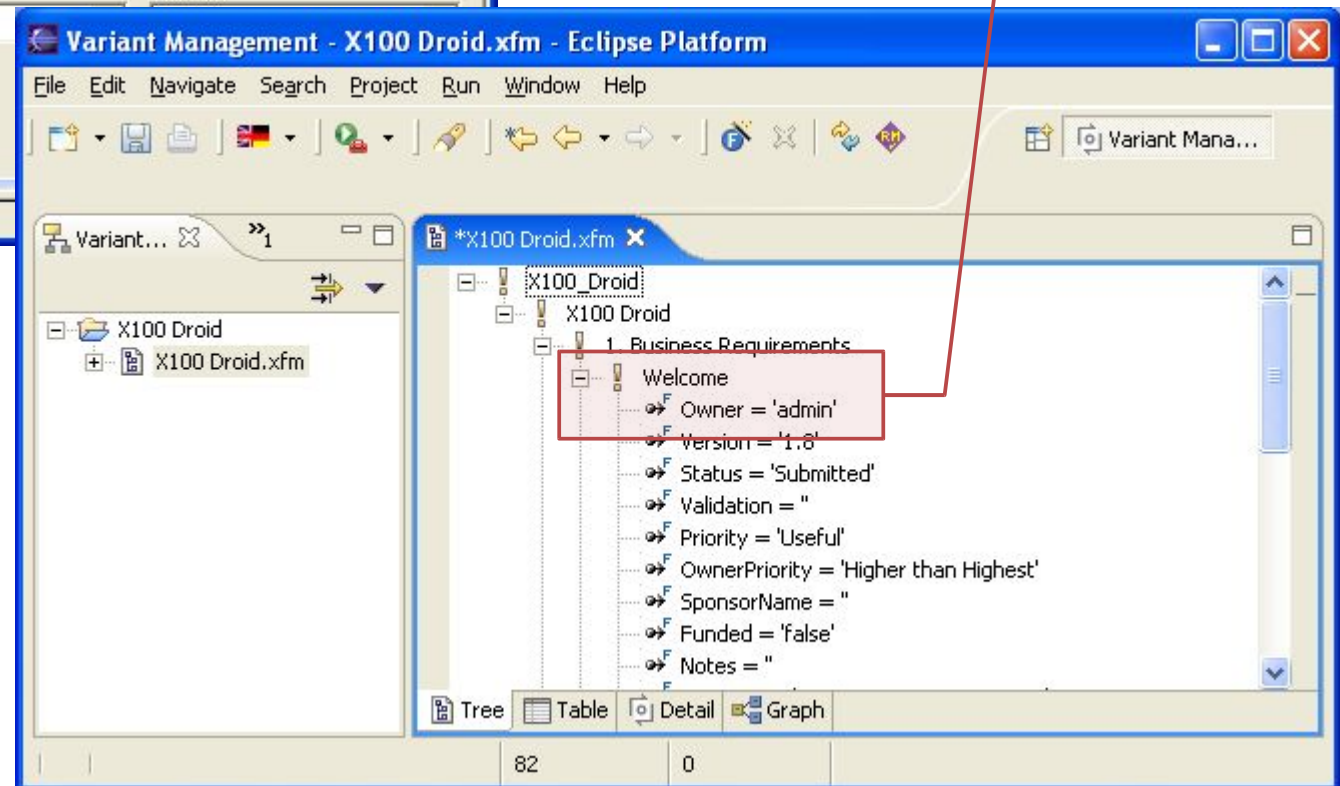


# Initial Representation as Feature Model

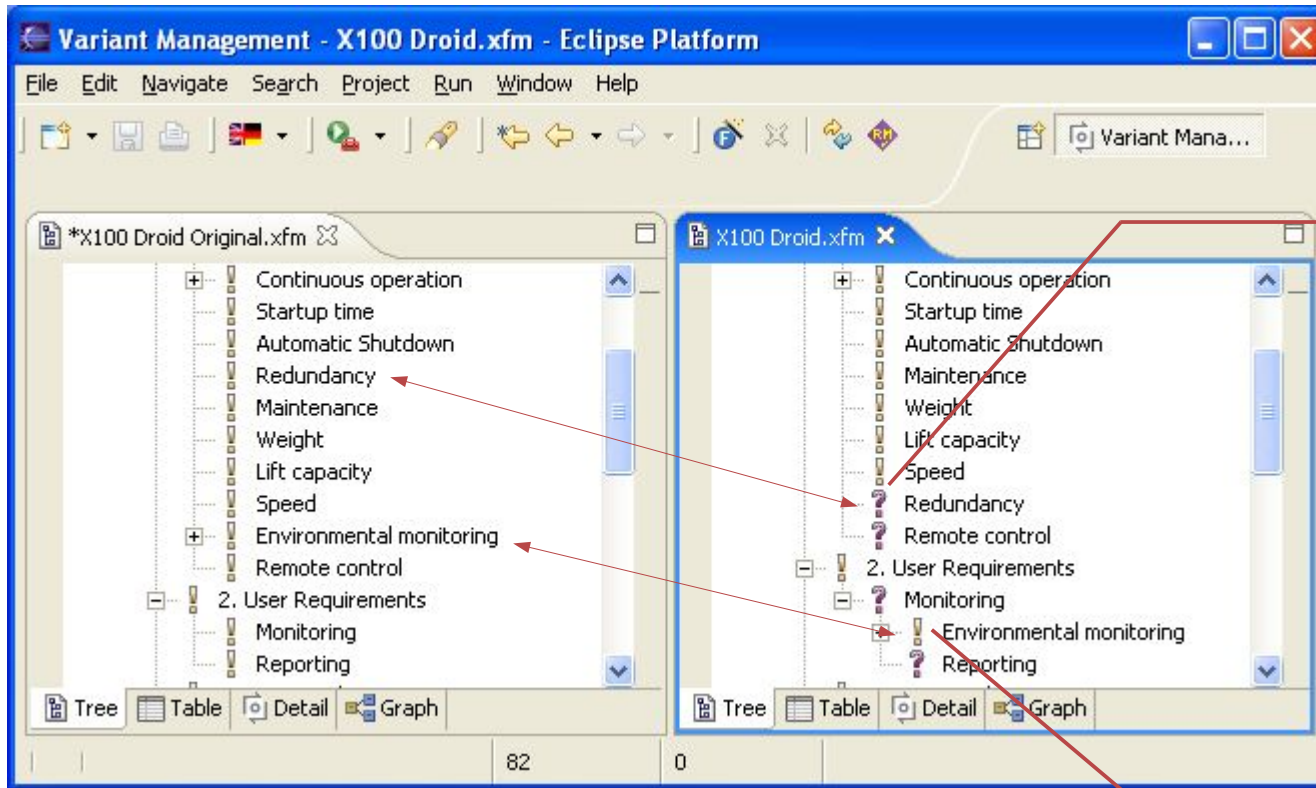


All requirements are represented initially as mandatory features.

The project hierarchy is used as the feature hierarchy during the transformation. Requirement attributes can also be transformed.



# Step 2: Expressing variability

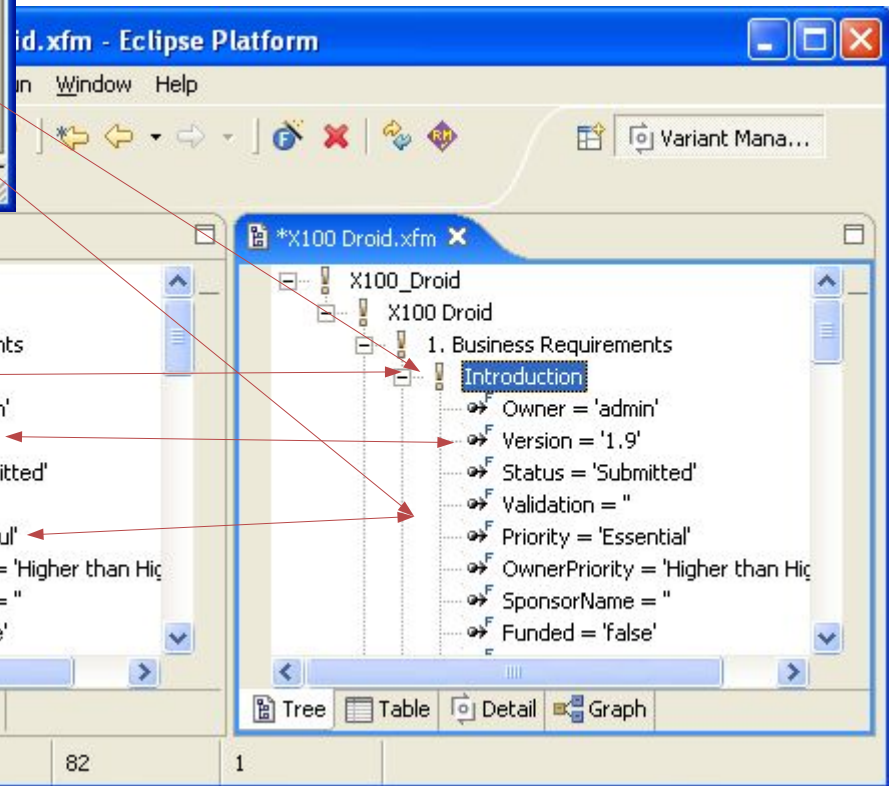
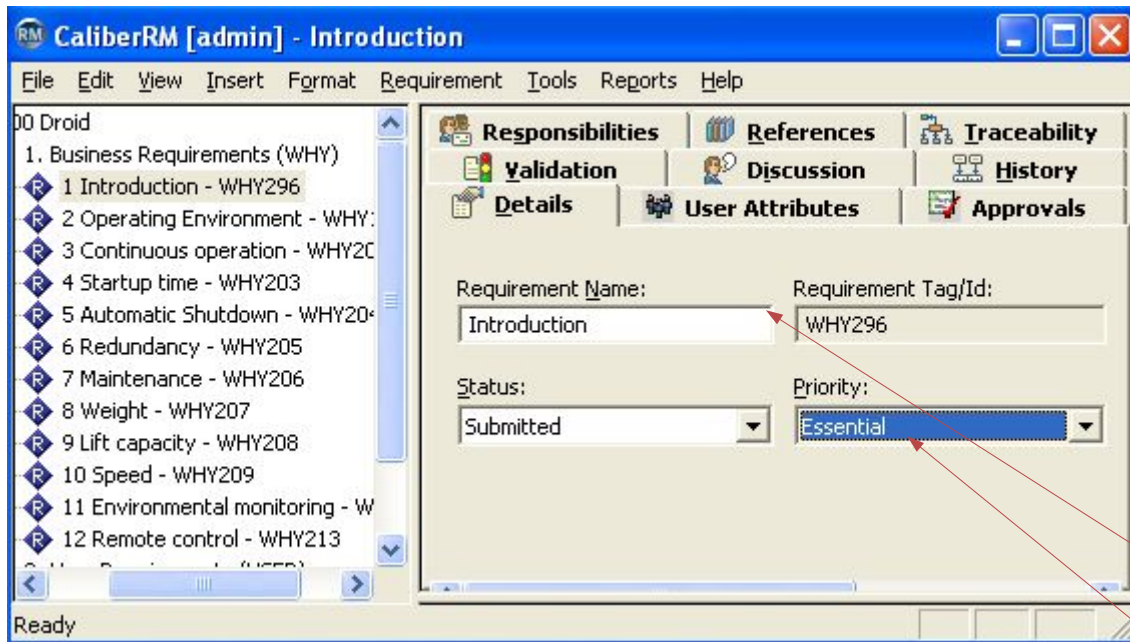


Since "Redundancy" is not mandatory for all products, it becomes an "Option".

"Environmental Monitoring" is permitted only if "Monitoring" is selected. Therefore it is moved below "Monitoring"

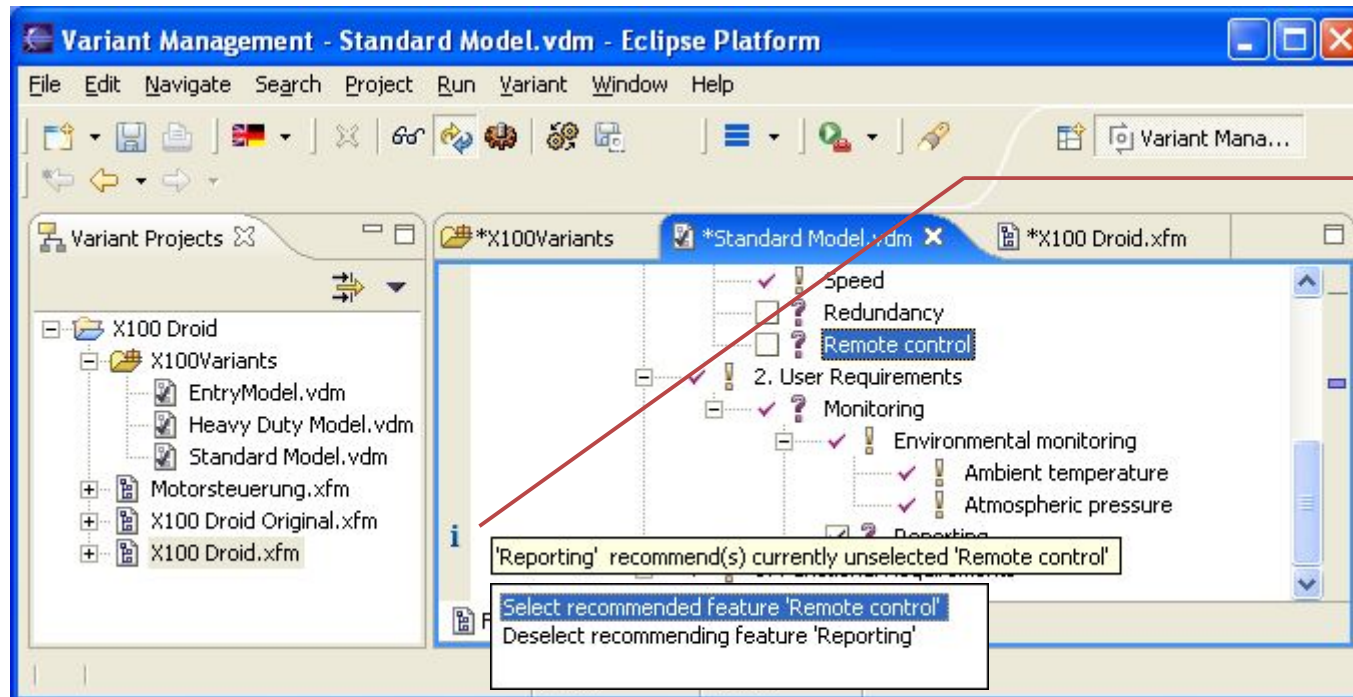
Variability is represented by changing the feature type, by adding restriction rules to features, or through relations like "mutual exclusion". Rearrangement of the feature hierarchy is also possible. If needed, extra features can be included to model more detailed variability information.

# Step 3: Model Update



Feature models may be updated from the CaliberRM project at any time. If necessary, feature properties and attributes will be changed during update. Deletion or addition of requirements in CaliberRM will also be reflected in feature models. However, the feature hierarchy is never changed during update.

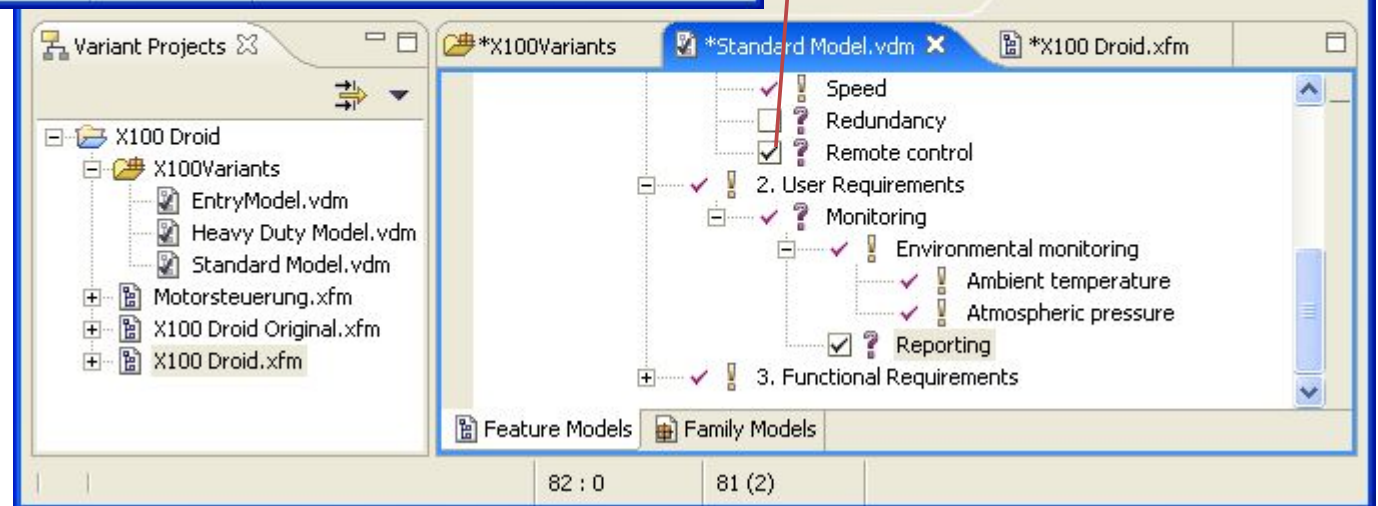
# Step 4: Variant Definition



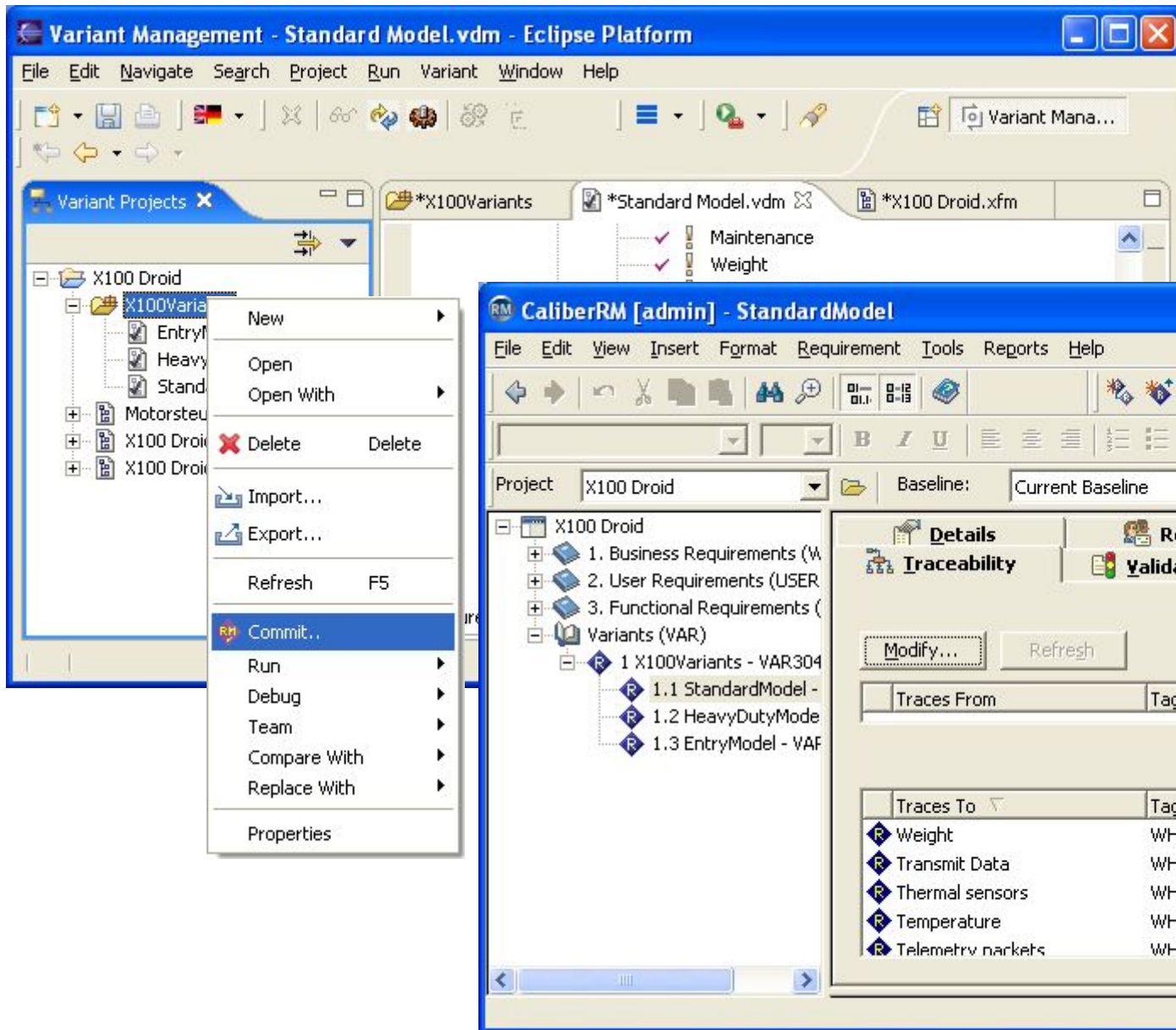
Markers points to configuration problems and offer solutions

Icons in boxes indicate selection state of features (user selected, automatic, ...)

Variants are defined in a special view permitting selection of features from the defined feature models. Problems in the feature selection are displayed and, if possible, resolved automatically. It is possible to attach any number of variant configurations to a requirements feature model.



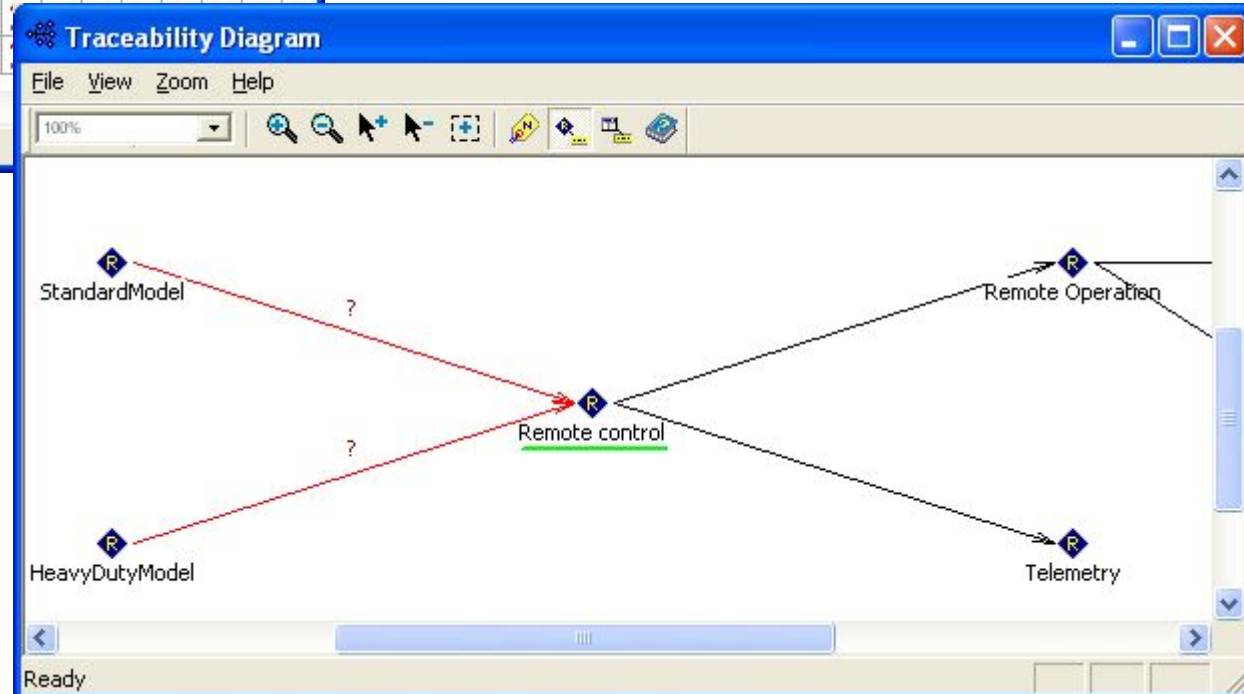
# Step 5: Variant Synchronization



Variants or variant groups can be synchronized back into CaliberRM projects. By default, each variant becomes a single requirement (of a specific requirement type) and points with Traces to all contained requirements. Synchronization is initiated from pure::variants.

# Variant Representation in CaliberRM

	Memory	Titanium outer shell	Exposed parts	A-D Converters	Telemetry	Thermal sensors	Elbow Joint	Grasp Rotation	Grasp pressure	Mechanical Grasp	Load capacity	Mechanical Arm	Location sensing	Pattern matching reso	Pattern matching spee	Preemption	Autonomous Operation	Batteries	Temperature	Humidity
X100Variants																				
EntryModel	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
HeavyDutyModel	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
StandardModel	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?



After synchronizing variant models back into CaliberRM users can analyse variant specific relations. Optional generation of variant specific "copies" of requirement projects is also possible.

# pure::variants Synchronizer for CaliberRM

- Extension module for pure::variants
- Compatibility:
  - pure::variants 2.0 Developer Edition/Server Edition and newer (Win32, Linux, MacOS X)
  - CaliberRM 2005 (6.0) (support of older versions available on request)
- Availability: now

# More Information



pure-systems



pure::variants

- **Telephone**

+49 391 5445 69 -0

the fast path for all your questions

- **Internet**

[www.pure-systems.com](http://www.pure-systems.com)

here you'll find additional information about pure::variants and pure-systems GmbH

- **e-mail**

[info@pure-systems.com](mailto:info@pure-systems.com)