



pure::variants



pure-systems

Synchronizer for Telelogic® DOORS

pure::variants - The tool



pure::variants

- **Integration**

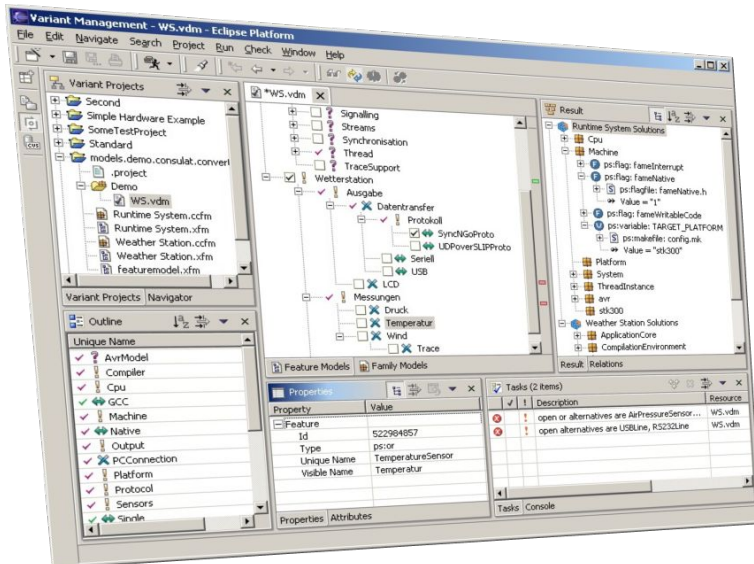
can be integrated into existing development processes independently from used programming language

- **Uses**

automated resolution of relation conflicts
user specific views on variability models

- **Benefits**

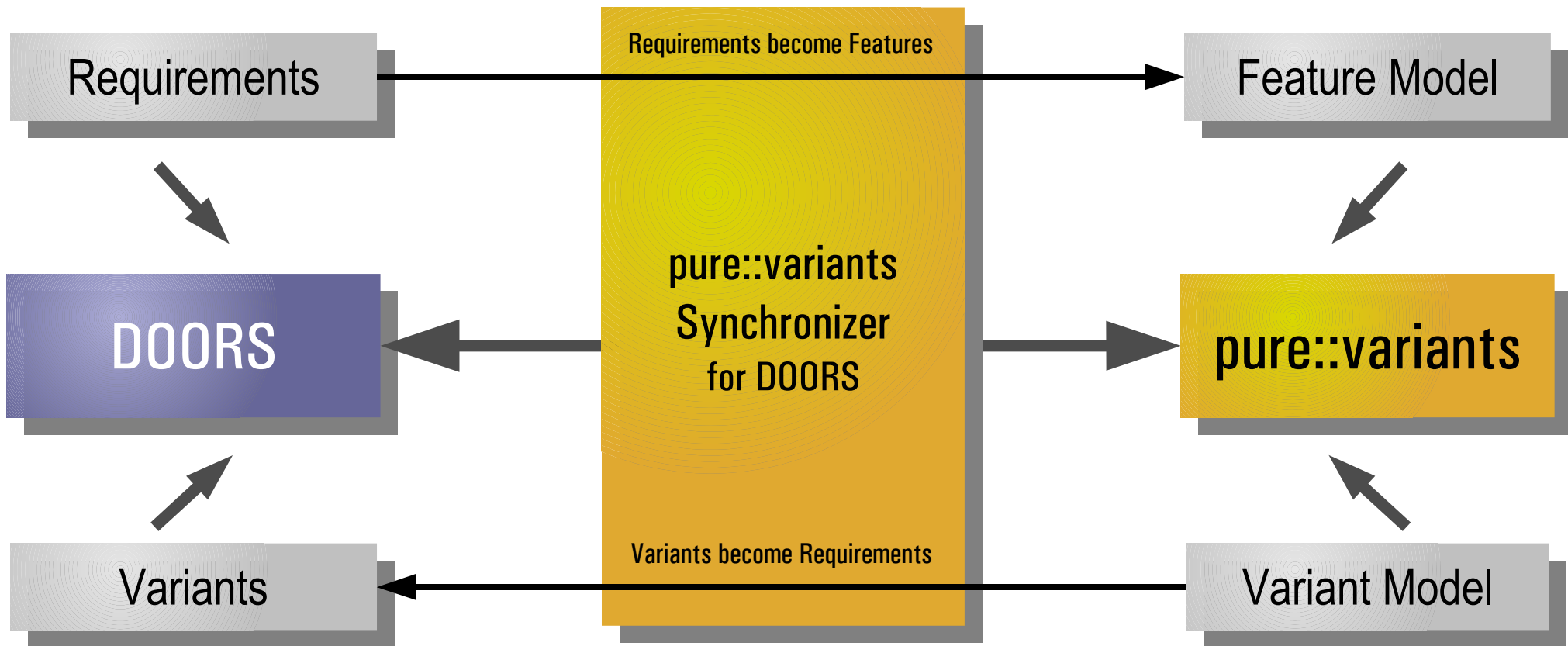
improved complexity management
all benefits of product line based development



The Challenge

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

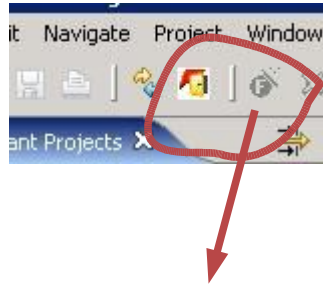
Our Answer



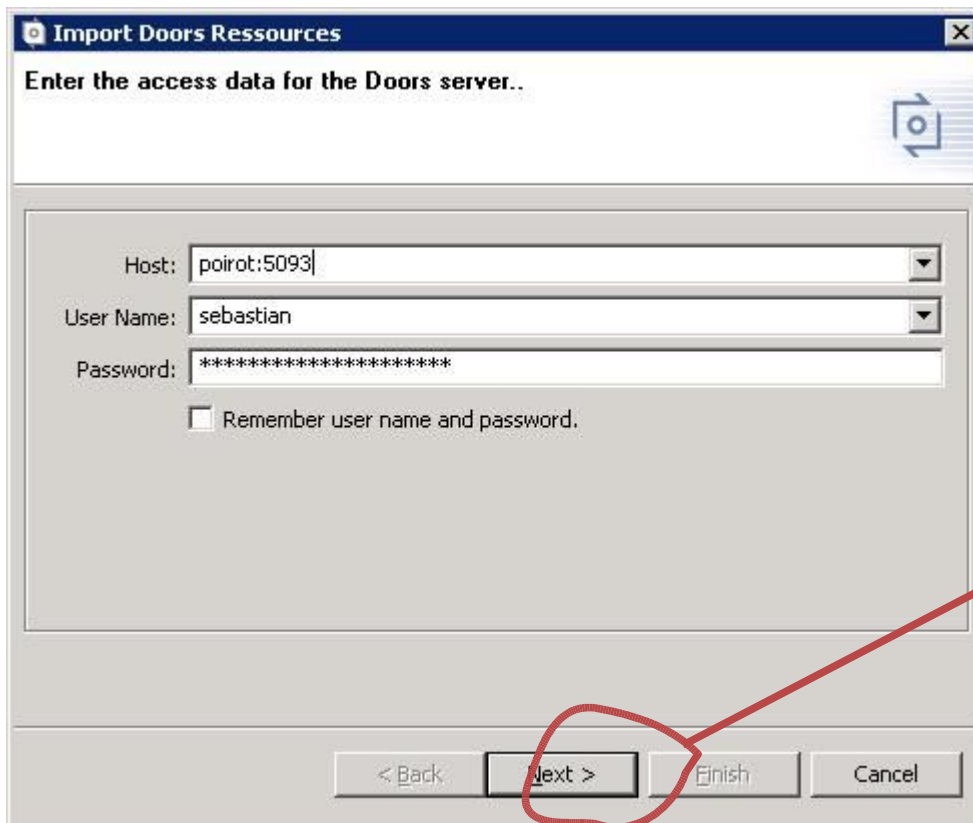
pure::variants Synchronizer for DOORS

- Transforms DOORS Modules into pure::variants feature models
 - Feature models may be updated at any time without loss of information
- Uses pure::variants variability modeling capabilities
 - Powerful rule language
 - Conflict detection and resolution
- Transforms pure::variants Variant Models into DOORS requirements.
- Permits direct integration of variable requirements into software configuration process.

Step 1: Data import from DOORS



The transformation of DOORS Modules is controlled by a pure::variants wizard.



Import Doors Ressources

Enter the access data for the Doors server..

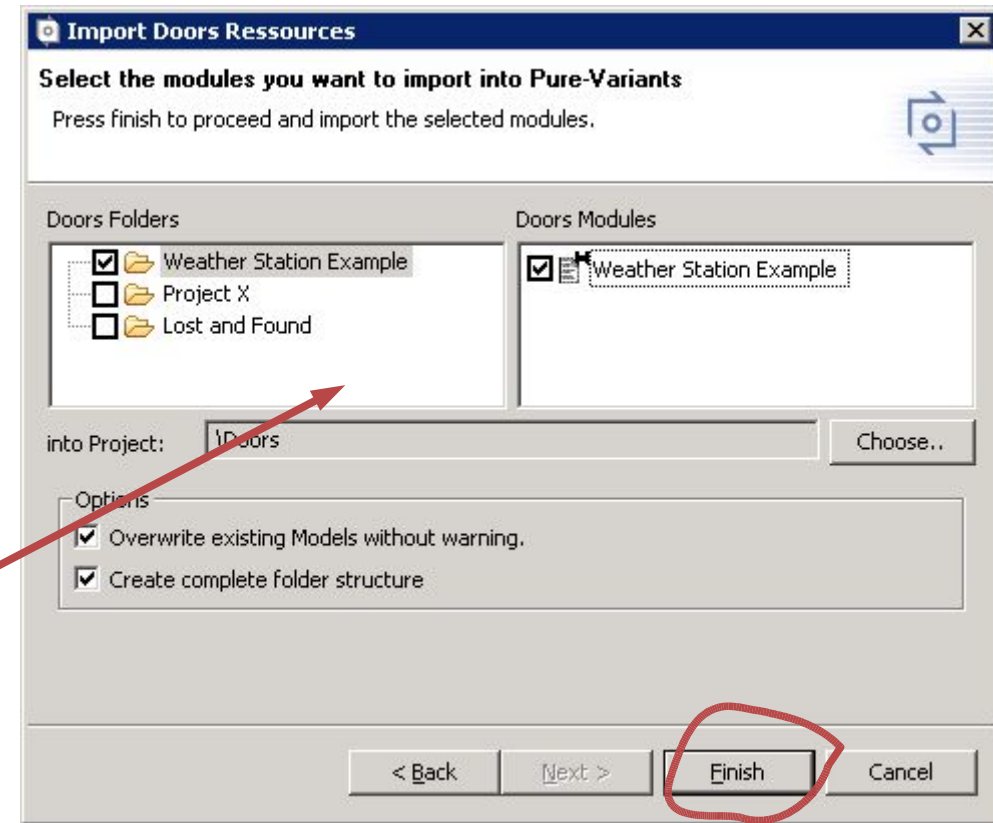
Host: poirot:5093

User Name: sebastian

Password: *****

Remember user name and password.

< Back **Next >** Finish Cancel



Import Doors Ressources

Select the modules you want to import into Pure-Variants

Press finish to proceed and import the selected modules.

Doors Folders

- Weather Station Example
- Project X
- Lost and Found

Doors Modules

- Weather Station Example

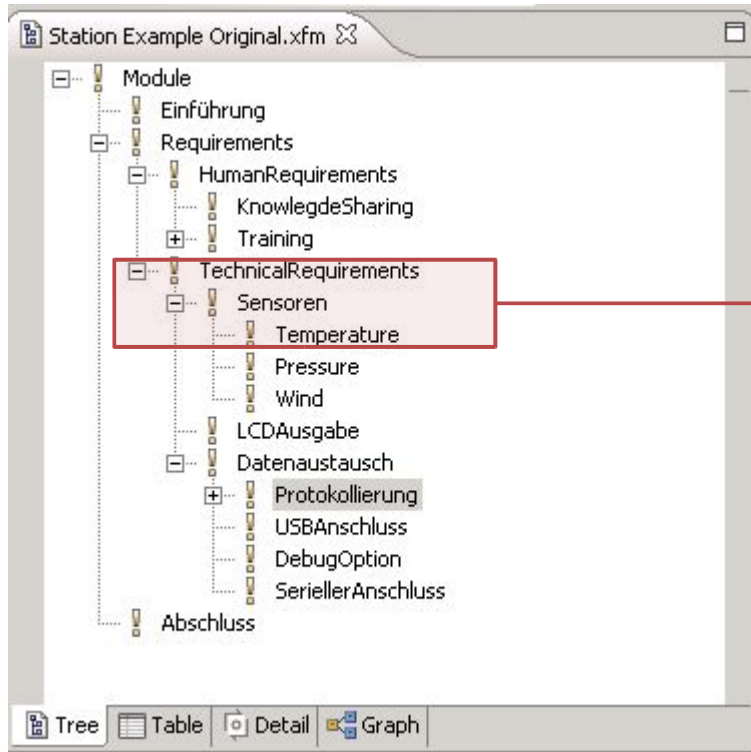
into Project: |Doors Choose..

Options

- Overwrite existing Models without warning.
- Create complete folder structure

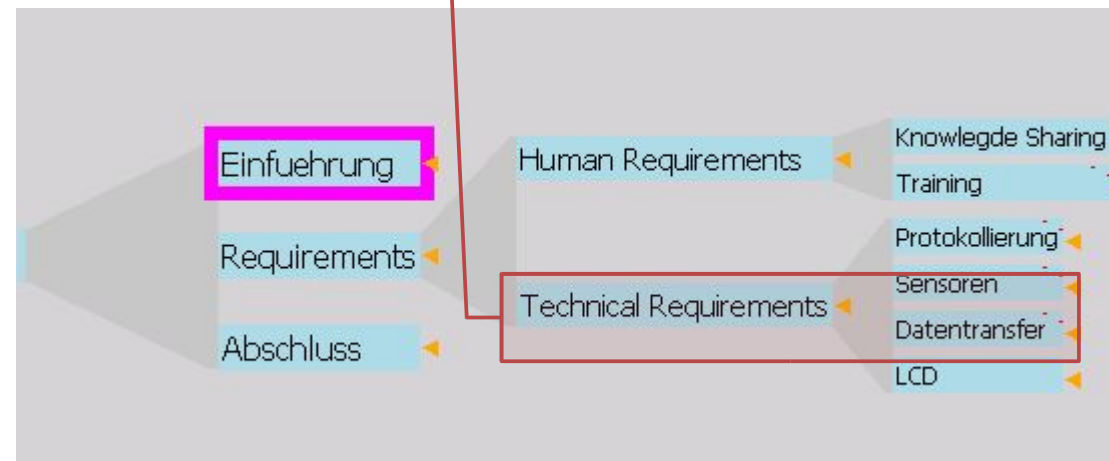
< Back Next > **Finish** Cancel

Initial Representation as a Feature Model

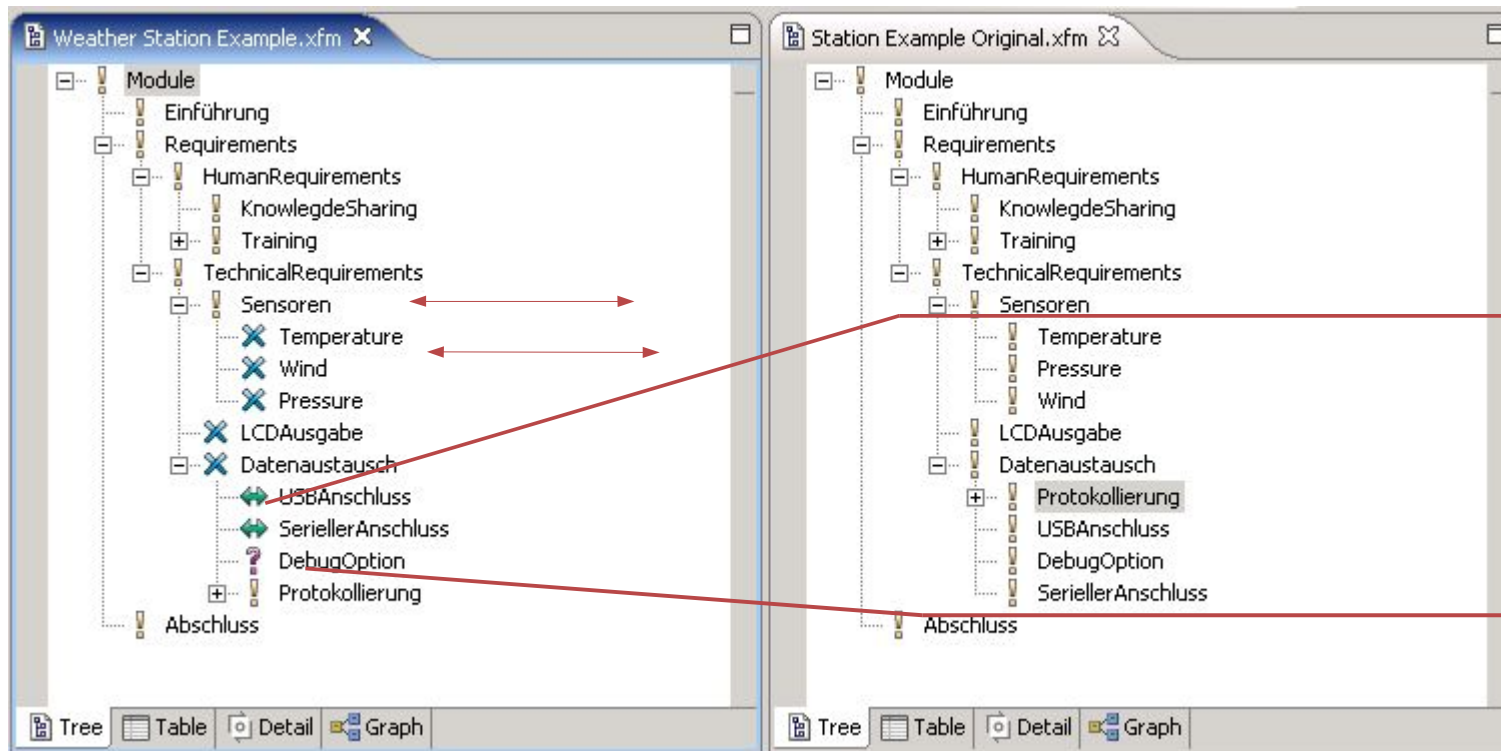


During the transformation the Module hierarchy is used as the Feature model hierarchy. Requirement attributes can also be imported into pure::variants.

All requirements are initially represented as mandatory features.



Step 2: Expressing variability



The Serial and USB connections are mutually exclusive. This is modelled using the *Alternative* feature type.

Since *Debugging* is not mandatory for all products, it becomes an *Option*.

Variability is represented by changing the feature type, by adding restriction rules to features, or through feature relations such as *Mutual exclusion*.

The Feature model hierarchy may be freely rearranged and new features may be added to the model to represent more detailed variability information.

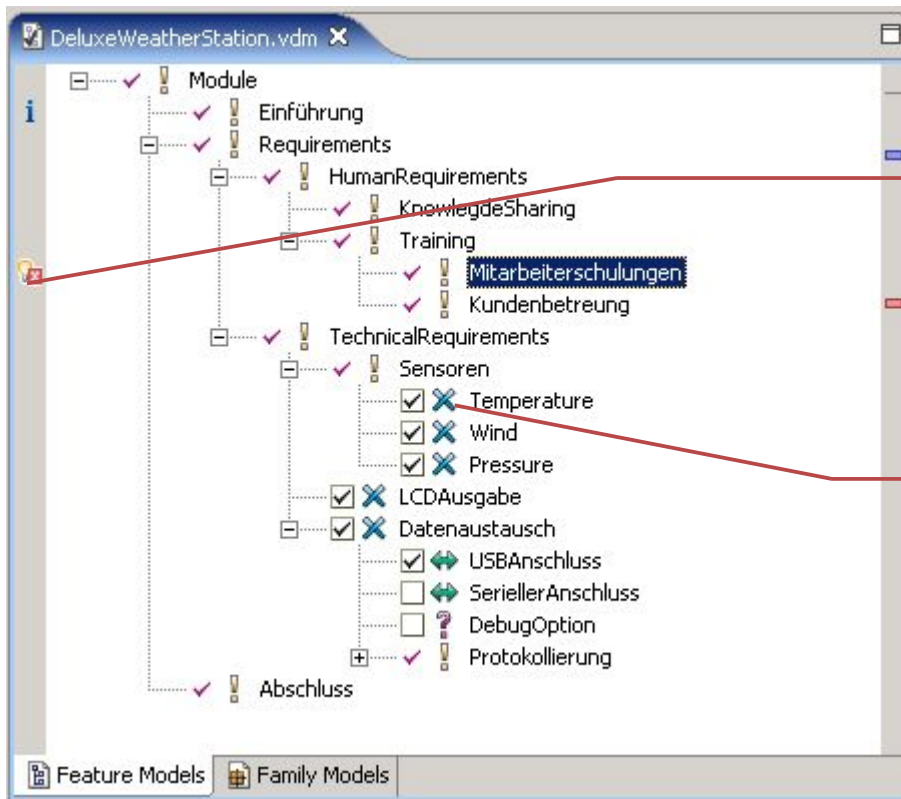
Step 3: Model Update

The screenshot displays the DOORS software interface. On the left, the 'Variant Projects' pane shows a project structure with 'Station Example Original.xfm' selected. The middle pane shows a hierarchical tree of requirements, with 'Verkaufspersonal Schulung' highlighted. The right pane shows a detailed view of the selected requirement, '2.1.2.1 Verkaufspersonal Schulung', with a red arrow pointing from the tree to the detail view.

At any time requirement feature models may be updated from DOORS. During update feature properties and attributes will be changed if necessary.

Deleting or adding requirements in DOORS will also be replicated in feature models. However, the feature hierarchy will not be changed during update.

Step 4: Variant Definition



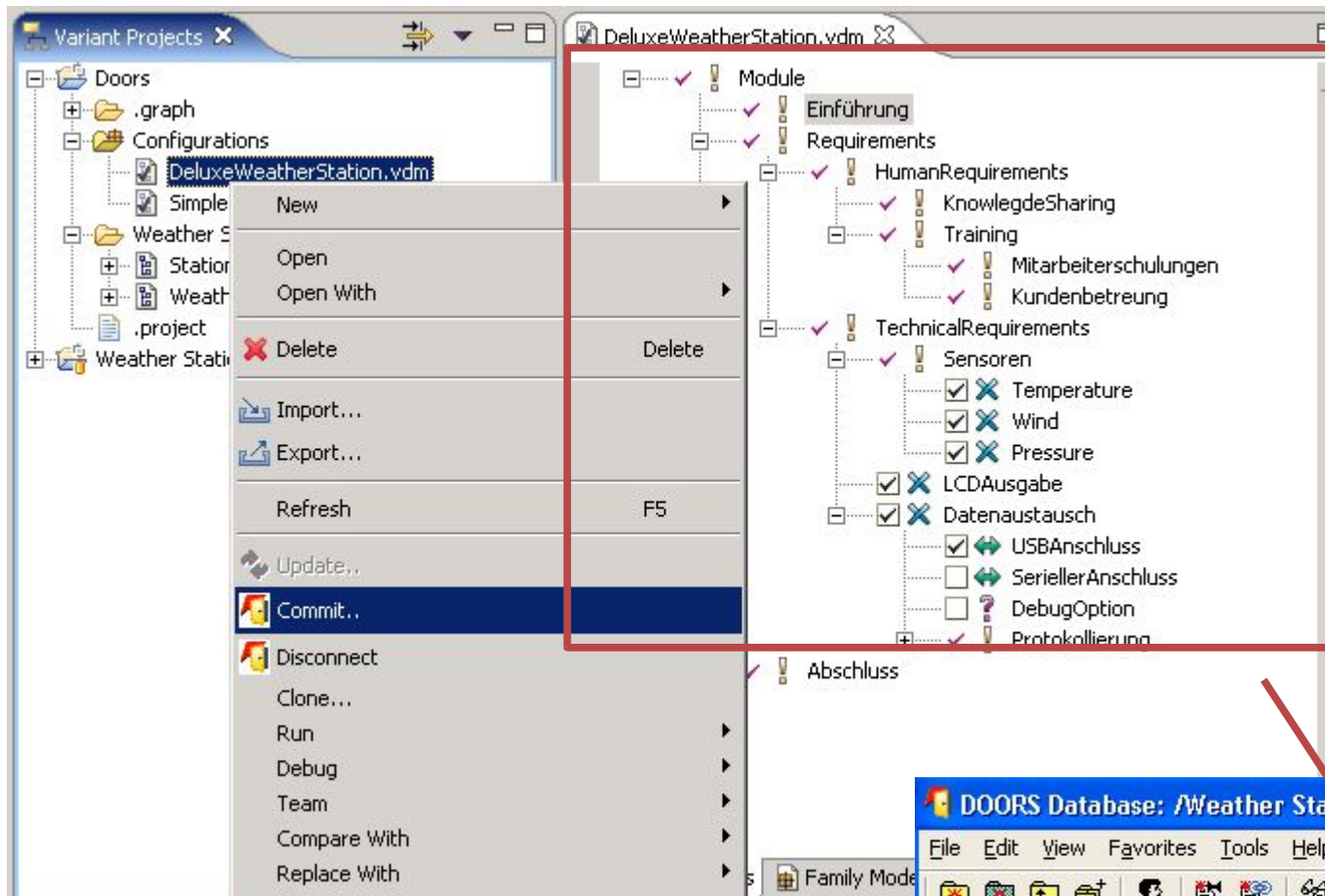
Markers point to configuration problems and offer solutions

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

Variant definition is performed in a special view permitting selection of features from the defined feature models. Problems in the feature selection for a variant 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



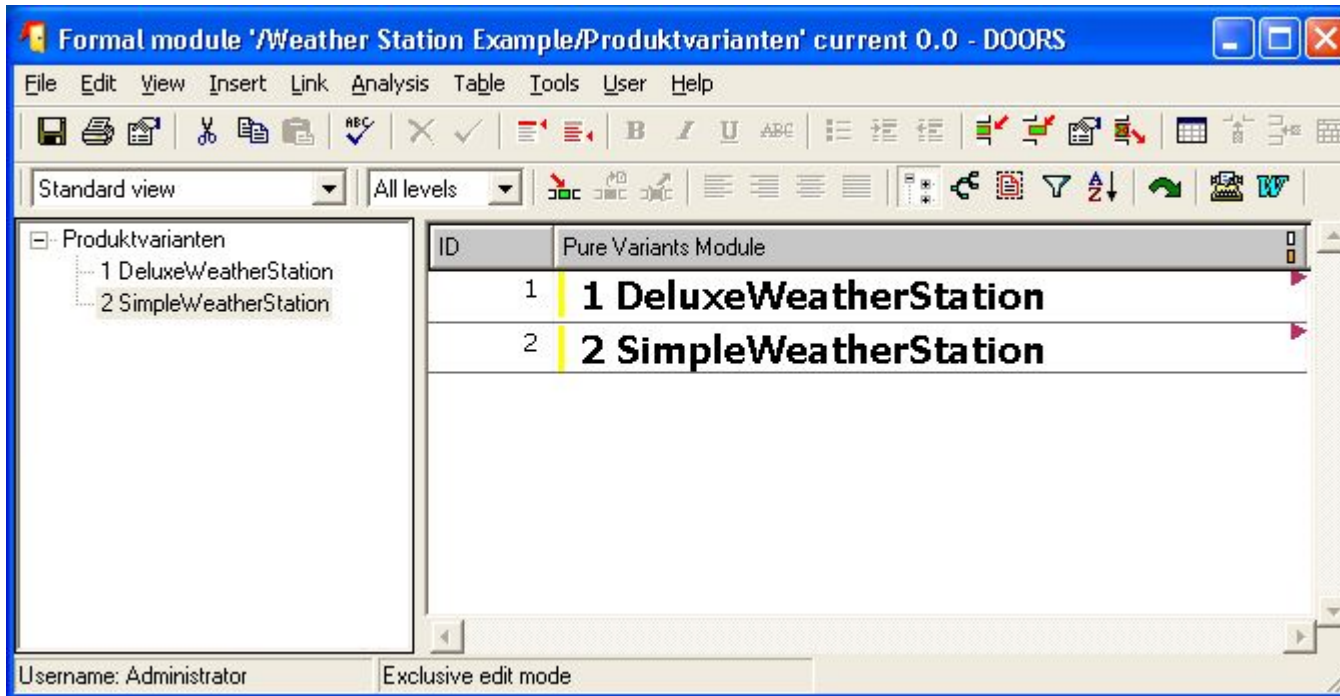
Variants or variant groups can be synchronized back into DOORS.

By default each variant becomes a single requirement (of a specific requirement type) and links to all contained requirements.

Synchronization is initiated from pure::variants



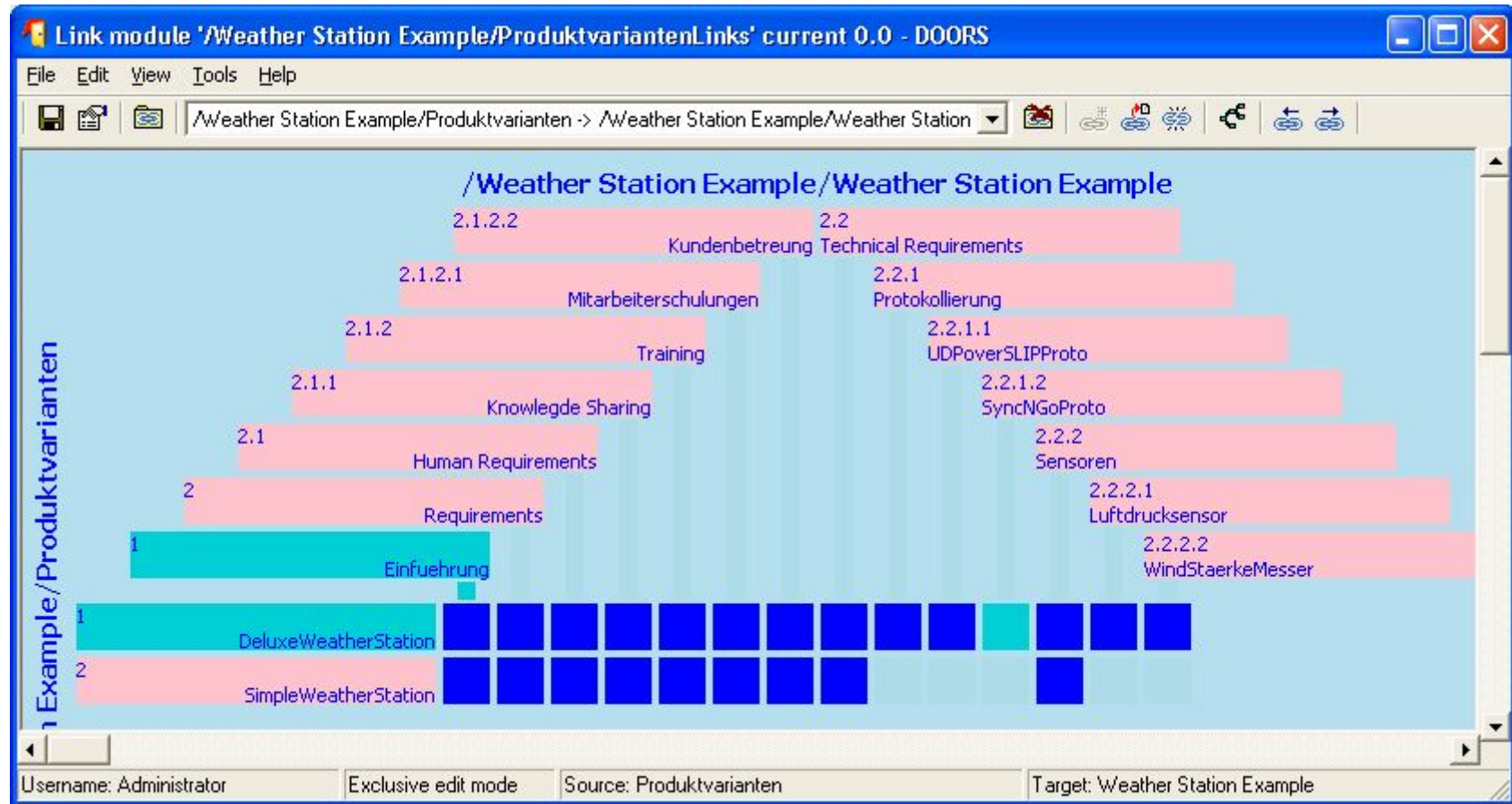
Variant Representation in DOORS



- 1: Einfuehrung
- 2: Requirements
- 3: LCD - Ausgabe: Standard Flüssigk
- 4: Datenaustausch: Es muss mindeste
- 6: USB Anschluss: Dieser Anschluss
- 8: Technical Requirements
- 9: Human Requirements
- 11: Training: Jeder Mitarbeiter und
- 12: Knowlegde Sharing: Da Mitarbeite
- 13: Kundenbetreuung
- 14: Mitarbeiterschulungen
- 15: Abschluss
- 16: Sensoren
- 18: Temperature: Dieser Sensor misst
- 19: Wind: Dieser Sensor misst die Wi
- 20: Pressure: Dieser Sensor misst de
- 21: Protokollierung: Es ist unbeding
- 23: UDPOverSLIPProto

By synchronising variants back into DOORS, users can now analyse variant-specific relations.

Variant Representation in DOORS



By synchronising variants back into DOORS, users can now analyse variant-specific relations.

Variant Representation in DOORS

The image displays three screenshots of the DOORS Database interface, illustrating how requirement modules are represented in different variants. Each screenshot shows a tree view on the left and a table on the right. Red boxes highlight the 'DOORS Links' and 'WeatherStationExample' entries in the table.

Screenshot 1 (Top Left): DOORS Database: /WeatherStationExample - DOORS. Location: /WeatherStationExample. The table shows:

Name	Type
DOORS Links	Link
WeatherStationExample	Formal

Screenshot 2 (Top Right): DOORS Database: /WeatherStationVariants/DeluxeWeatherStation/Weath... Location: /WeatherStationVariants/DeluxeWeatherStation/W... The table shows:

Name	Type
DOORS Links	Link
WeatherStationExample	Formal

Screenshot 3 (Bottom): DOORS Database: /WeatherStationVariants/SimpleWeatherStation/Weath... Location: /WeatherStationVariants/SimpleWeatherStation/We... The table shows:

Name	Type
DOORS Links	Link
WeatherStationExample	Formal

Generation of variant specific *copies* of requirement Modules is also possible. These contain only the objects belonging to the variant and their superordinate objects.

Other Capabilities

- Support for requirement variation
 - Automatic conversion to alternatives in the Feature model
 - Multi-Level Variation
 - Selective import of the original requirement attributes
- Automatic connection of originals and copies through Links.
- Selection of the attributes to import on a per Module basis

pure::variants Synchronizer for DOORS

- Extension module for pure::variants
- Compatibility:
 - pure::variants 2.0 Server Edition/Developer Edition and newer (Win32, Linux, MacOS X)
 - Doors 8.0, [7.1] (support of older versions available on request)
- Availability: now

For More Information



pure-systems



pure::variants

- **Telephone**

+49 391 5445 69 -0

the fast path for all your questions

- **Internet**

www.pure-systems.com

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

- **e-mail**

info@pure-systems.com