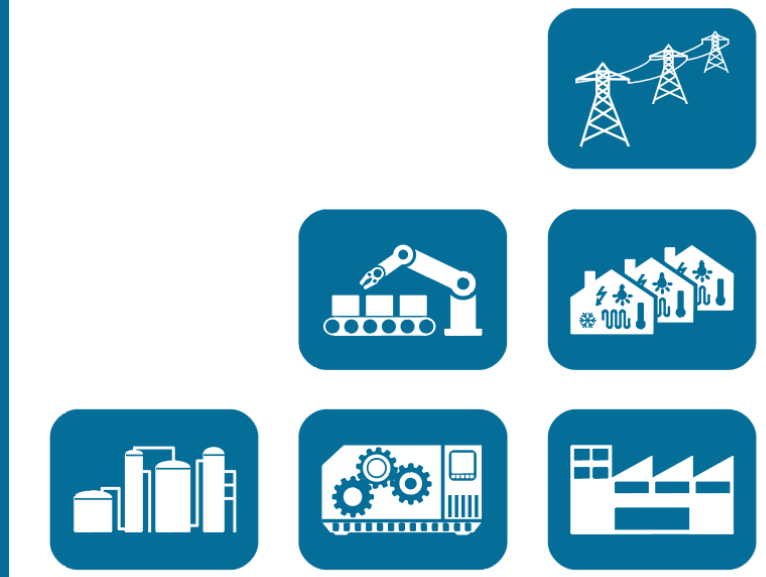
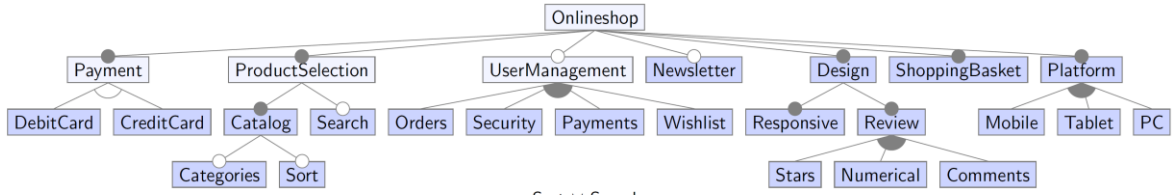
Tutorial on the Universal Variability Language – Session 3



Variability Modeling Approaches

- **Feature modeling (FeatureIDE, UVL, Clafer,…)**
- **Decision modeling (DOPLER, IVML,…)**
- Orthogonal Variability modelling (OVM)
- UML-based variability modelling
- Delta-oriented
- Textual variabi
- Common Varia
- Kconfig (Linux
- Component Definition Language (eCos)



$$\text{Sort} \vee \text{Search}$$
$$\text{Search} \implies \text{Security}$$
$$(\text{Security} \implies \neg\ \text{Payments}) \wedge (\text{Payments} \implies \neg\ \text{Security})$$
$$(\text{Responsive} \implies \text{Mobile}) \wedge (\text{Responsive} \implies \text{Tablet}) \wedge (\text{Responsive} \implies \text{PC})$$
$$(\text{Stars} \wedge \text{Numerical} \wedge \text{Comments}) \vee (\text{Stars} \wedge \text{Numerical} \wedge \neg\ \text{Comments}) \vee (\text{Stars} \wedge \neg\ \text{Numerical} \wedge \text{Comments}) \vee (\neg\ \text{Stars} \wedge \text{Numerical} \wedge \text{Comments})$$

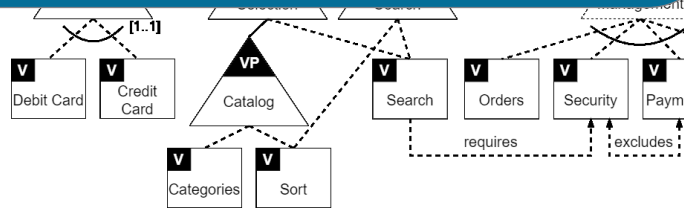
ID	Question	Type	Range	Card.	Constraint/Rule	Visibility
Payment	Which payment methods should be supported?	Enum	DebitCard | CreditCard	1:1		
Search	Should a search function be supported?	Boolean	true | false		if (Search) { UserManagement = Security } if (!Search) { Sort = true }	
Categories	Split products into categories?	Boolean	true | false			
					if (!Sort) { Search = true }	

					if (Security) { disAllow(Payments) }
					if (!Security) { allow(Payments) }
					if (Payments) { disAllow(Security) }	

				ist 1:4	if (!Payments) { allow(Security) }	
				2:3		

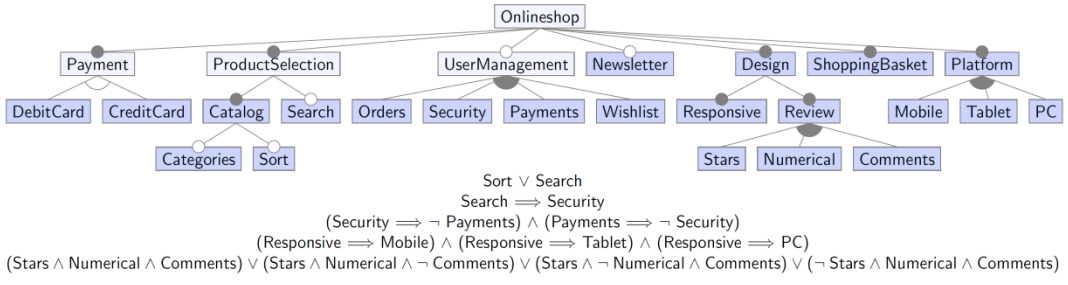
How can researchers and practitioners better understand the strengths and weaknesses of an approach?

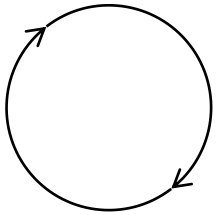
How can we support researchers and practitioners in picking the right approach for their specific use case?




```
menu "Power management and ACPI options"
depends on !X86_VOYAGER

config PM
  bool "Power Management support"
  depends on !IA64_HP_SIM
  ---help---
      "Power Management" means that ...

config PM_DEBUG
  bool "Power Management Debug Support"
  depends on PM

config CPU_IDLE
  bool "CPU idle PM support"
  default ACPI

config PM_SLEEP
  bool
  depends on SUSPEND || HIBERNATION ||
             XEN_SAVE_RESTORE
  default y
...
endmenu
```

Variability Model Transformations





ID	Question	Type	Range	Card.	Constraint/Rule	Visibility
Payment	Which payment methods should be supported?	Enum	DebitCard | CreditCard	1:1		
Search	Should a search function be supported?	Boolean	true | false		if (Search) { UserManagement = Security } if (!Search) { Sort = true }	
Categories	Split products into categories?	Boolean	true | false			
Sort	Should products be sortable?	Boolean	true | false		if (!Sort) { Search = true }	

UserManagement	Which user management options should be added?	Enum	Orders | Security | Payments | Wishlist	1:4	if (Security) { disAllow(Payments) }
					if (!Security) { allow(Payments) }
					if (Payments) { disAllow(Security) }
					if (!Payments) { allow(Security) }	

Newsletter_percent	How many percent off?	Number	0 - 100			
Newsletter_mail	From which mail should the newsletter be sent?	String				
Review	Which review techiques should be supported	Enum	Stars | Numerical | Comments	2:3		

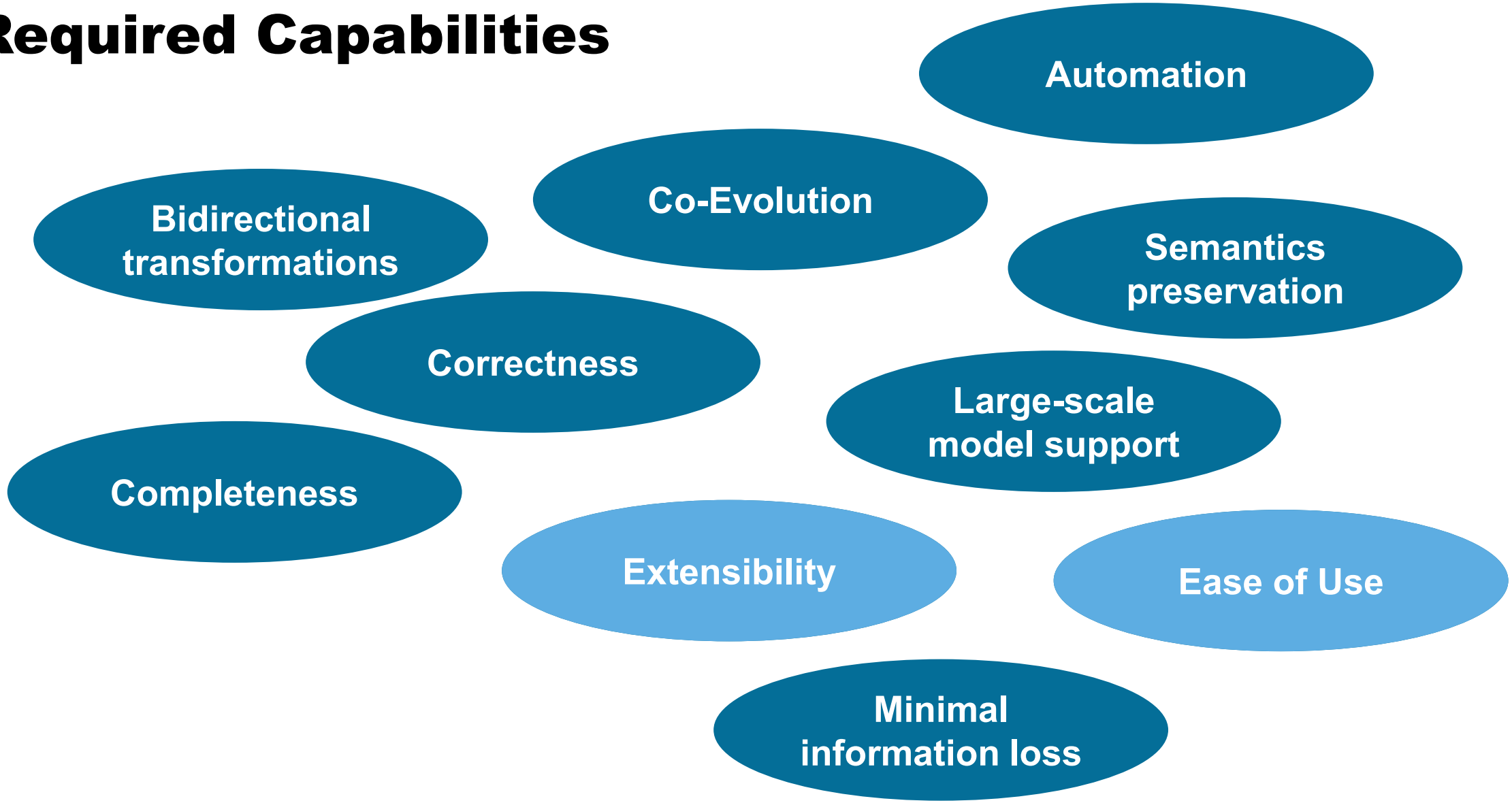
(1) Switching to a different approach without losing invested modeling efforts

(2) Experimenting with different approaches before selecting one

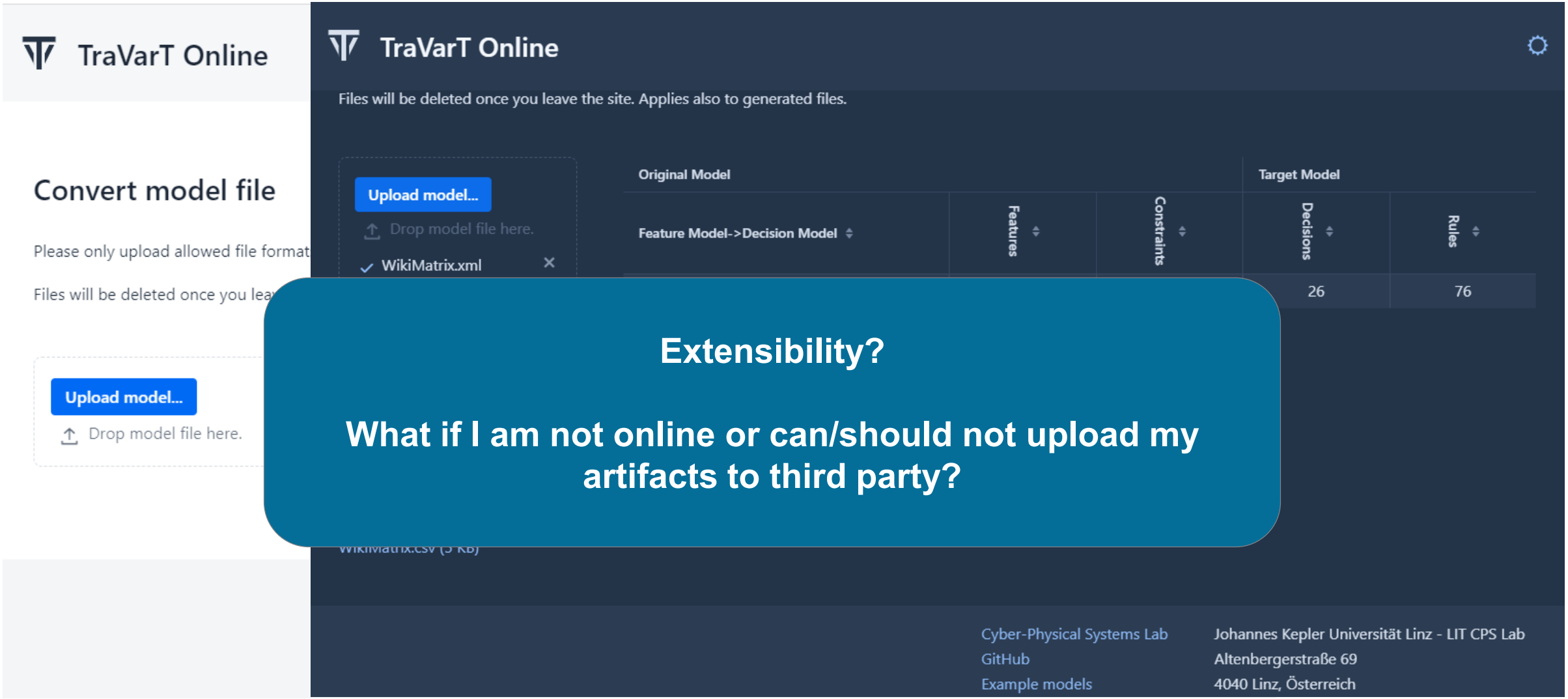
(3) Integrating tools of other approaches, e.g., for analysis

(4) Transforming custom-developed variability artifacts into well-knowns

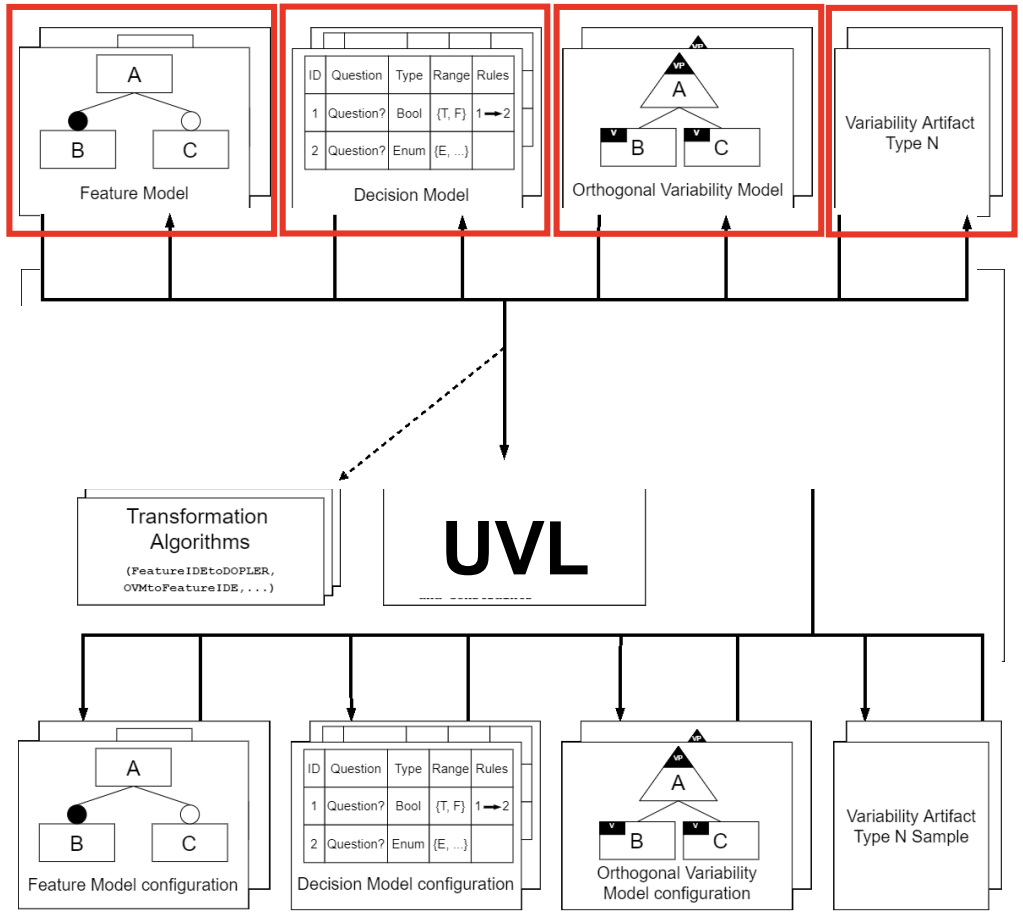
Required Capabilities



WebTraVarT


TraVarT Online
TraVarT Online
Files will be deleted once you leave the site. Applies also to generated files.
Convert model file
Please only upload allowed file format
Files will be deleted once you lea
Upload model...
Drop model file here.
Upload model...
Drop model file here.
WikiMatrix.xml
Original Model
Target Model
Feature Model->Decision Model
Features
Constraints
Decisions
Rules
26
76
Extensibility?
What if I am not online or can/should not upload my artifacts to third party?
Cyber-Physical Systems Lab
GitHub
Example models
Johannes Kepler Universität Linz - LIT CPS Lab
Altenbergerstraße 69
4040 Linz, Österreich

TraVarT: Transforming Variability Artifacts




```
Usage: travart [-hVv] [COMMAND]
TraVarT main command to transform and validate variability artifacts.
  -h, --help      Show this help message and exit.
  -v, --verbose   Enable verbose log information during execution.
  -V, --version   Print version information and exit.
Commands:
  transform  Transforms the given variability artifacts into another type.
  plugin     Lists the available plugins for the TraVarT command.
```

Github: https://github.com/SECPS/TraVarT - WebTraVarT: https://litcps.jku.at/travart/

Practical Part I

Open TRAVART folder and transform some artifacts

Implementing a TraVarT Plugin

```java
public interface IPlugin<T> extends ExtensionPoint {

    /**
     * Returns the transformer of the plugin to transform the variability model.
     *
     * @return the transformer of the plugin to transform the variability model.
     */
    IModelTransformer<T> getTransformer();

    /**
     * Returns the reader of the plugin to read the variability model from the file
     * system.
     *
     * @return the reader of the plugin to read the variability model from the file
     *         system.
     */
    IReader<T> getReader();

    /**
     * Returns the statistics of the plugin to get the statistics the variability
     * model.
     *
     * @return the statistics of the plugin to get the statistics the variability
     *         model.
     */
    IStatistics<T> getStatistics();

    /**
     * Returns the writer of the plugin to write the variability model to the file
     * system.
     *
     * @return the writer of the plugin to write the variability model to the file
     *         system.
     */
    IWriter<T> getWriter();

    /**
     * Returns the variability model type name.
     *
     * @return the name of the variability model type.
     */
    String getName();

    /**
     * Returns the version of the plugin.
     *
     * @return the version of the plugin.
     */
    String getVersion();

    /**
     * Returns a unique ID of the plugin, such that it can be identified.
     *
     * @return the unique ID of the plugin.
     */
    String getId();

    /**
     * Returns a iterable of file extensions for which this plugin is applicable.
     *
     * @return a unmodifiable list of file extensions.
     */
    Iterable<String> getSupportedFileExtensions();
}
```

Practical Part III

Practical Example FeatureIDE

Future Work

- Make todays elements (cli and plugin extension support) publicly available
- Make further plugins publicly available
- Extend CLI commands
  - Validating and evaluating the created variability artifacts
  - Option for one-way or roundtrip focused transformation
  - Option to log statistical information
  - Option to create different types of artifacts at once

What would you need from an CLI to enable your use case?