Using smart packages in your model

To better understand the usability of the smart packages, please see the following case studies.

Case study #1: Gathering use cases

the case study uses the sample project use case diagram.mdzip, which can be found in < MagicDraw installation folder>\samples\diagrams.

For example, we have a lot of use cases owned by different system boundaries in the project.

We need all these use cases to be in a single package, except for several particular ones. Also, we need all newly created use cases to be automatically included into this package.

In this case, the smart packages feature is very useful. We will create a smart package with dynamic contents to gather all use cases in the model, then demonstrate how it manages further changes in the model. Finally, we will create a snapshot of the smart package to have a static list of use cases as a milestone of the model development.

To accomplish this, do the following:

1. Create a smart package named All Use Cases.



2. Define the criteria for gathering the contents of the smart package. Specify the search options to find all use case type elements from the root package *Data*.

Find (Ð	Remove
What:	Q*	•
Type:	UseCase	
Scope:	Data	
	Search in names	🔘 Search in all texts 🚯

3. Expand the contents of the smart package All Use Cases.

🔁 Data		
E- D All Use Cases	«SmartPackage»	
···· 🗢 Add item	requirementUseCase»	
···· 🗢 Add new	customer user «requirementUseCase»	
···· 🗢 Add new	librarian user «requirementUseCase»	
Browse ite	ms «requirementUseCase»	
Browse re	servations «requirementUseCase»	
🗄 🗠 🔿 Cancel res	ervation «requirementUseCase»	
Change p	assword «requirementUseCase»	
Change s	<pre>/stem settings «requirementUseCase»</pre>	
Check sy	stem settings «requirementUseCase»	
O Edit custo	mer's profile «requirementUseCase»	
🗄 🔿 Edit item d	ata «requirementUseCase»	
🗢 Edit librari	an's profile «requirementUseCase»	
O Get custo	mer status summary «requirementUseCase»	
🗄 🔿 Get item d	etails «requirementUseCase»	
🗢 Get notific	ation «requirementUseCase»	
🗄 🔿 Increase r	eading items «requirementUseCase»	
🔿 Login «rea	uirementUseCase»	
🗢 Logout «r	equirementUseCase»	
🗄 🔿 Make rese	rvation «requirementUseCase»	
Order iter	IS	
🗄 🔿 Penalize f	or overdue «requirementUseCase»	
	oan «requirementUseCase»	
🗄 🗠 🔿 Register r	eturn «requirementUseCase»	
🗄 🔿 Remove c	ustomer user «requirementUseCase»	
🗄 🔿 Remove it	em «requirementUseCase»	
Remove li	orarian user «requirementUseCase»	
	an item «requirementUseCase»	
🗄 🖾 MagicLibrary		
Remove li Remove li And the second sec	orarian user «requirementUseCase» an item «requirementUseCase» bassword and Change system settings from the co	ntents. See the contents of the smart packag cases in the cell of the Excluded Elements
2000CCIIL	~~~~~~	*****
	Change password [MagicLibrary::System]	n Access]
Excluded Elements	Change system settings [MagicLibrary::S	System Settings]

	uded from the contents appear in it after being mandally added.	ļ
Additional Elements	 Change system settings [MagicLibrary::System Settings] Change password [MagicLibrary::System Access] 	
Excluded Elements		

8. Create a new use case named Filter by author under the system boundary Item Browser. See the new use case in the smart package All Use Cases.

E Data
🛱 🗔 All Use Cases «SmartPackage»
Add item «requirementUseCase»
Add new customer user «requirementUseCase»
Add new librarian user «requirementUseCase»
Browse items «requirementUseCase»
Browse reservations «requirementUseCase»
• Cancel reservation «requirementUseCase»
Check system settings «requirementUseCase»
Edit customer's profile «requirementUseCase»
🖽 🗢 Edit item data «requirementUseCase»
Edit librarian's profile «requirementUseCase»
Iter by author
Get customer status summary «requirementUseCase»
Get notification «requirementUseCase»
Create a snapshot of the smart package All Use Cases to have a static list of u

9. cases as a milestone of the model development.

🖃 🗠 🗖 🗠 🗠 All Use Cases «SmartPackage» 庄 🗔 Snapshot of All Use Cases «SmartPackage 🗄 🔁 MagicLibrary

10. Expand the contents of the snapshot and see that it equals the contents of the smart package All Use Cases.

Case study #2: Performing the requirements coverage analysis

the case study uses the sample project hybrid sport utility vehicle.mdzip, which can be found in < MagicDraw installation folder>\samples\SysML, if the

SysML plugin is installed.

Here is another case where the smart package feature is very helpful: we need to have all unsatisfied requirements in a separate package. Also, we need requirements to automatically disappear from this package after becoming satisfied.

We will create a smart package with dynamic contents to gather all the unsatisfied requirements in the model and a dependency matrix for performing the requirements coverage analysis. Then we will demonstrate how both the smart package and the dependency matrix reflect the transition of a requirement to satisfy.

Let's do the following:

- 1. Create a smart package named Unsatisfied Requirements.
- 2. To gather the contents of the smart package, add a new script operation (in the Expert mode of the Query dialog) and define the following OCL 2.0 expression as the criteria:

SysML::Requirements::Requirement::allInstances()->select(r|not r.supplierDependency->exists(d|d. oclIsKindOf(SysML::Satisfy)))

	Script 😗	Remove
	Language: OCL2.0 Body:	•
	SysML::Requirements::Requirement::allInstances()->select (r not r.supplierDependency->exists(d d.odIsKindOf(SysML::Sat	tisfy)))
	Check OCL syntax	

3. Expand the contents of the smart package Unsatisfied Requirements.

	E 🔁 Data
	🖶 🥕 Relations
	庄 📩 avs «validationSuite»
	🖶 💼 HSUVModel
	🗄 👘 ModelingDomain
	🗗 💭 Unsatisfied Requirements «SmartPackage»
	Braking «Requirement»
	Capacity «Requirement»
	Eco-Friendiness «Requirement»
	III Ergonomics «Requirement»
	E FuelEconomy «Requirement»
	PassengerCapacity «Requirement»
	⊕ ■ Performance «Requirement»
	PowerSourceManagement «Requirement»
	⊕
	SafetyTest «Requirement»
4. (Create a dependency matrix and define the following criteria:
	 Specify Requirement as the row element type
	 Specify Block as the column element type

- Specify the smart package *Unsatisfied Requirements* as the row scope
 Specify the package *HSUV Structure* as the column scope
 Specify the Satisfy relationship as dependency criteria
 In the Direction drop-down list, select Column to row.
 Criteria

Row Element Type:	Requirement	. (Column Element Type:	Block	
Row Scope:	Unsatisfied Requirements	. (Column Scope:	HSUV Structure	
Dependency Criteria:	Satisfy	. [Direction: Column to r	ow 👻 Show Elements: 🗚	-

5. On the dependency matrix, create a Satisfy relationship between the block BrakePedal and the requirement Braking.

	E.	· 🚞	HSU	JV St	ruct	ure-												
		Accelerator	AutomotiveDomain	BatteryPack	BodySubsystem	📃 BrakePedal	BrakeSubsystem	CAN_Bus	ChassisSubsystem	Differential	ElecPower	ElectricalPowerController	ElectricCurrent	ElectricMotorGenerator	Front Wheel	Fuel	EuelFlow	FuelInjector
🖃 💭 Unsatisfied Requirements																		
🎫 Braking																		1
🛱 🖓 🖾 Capacity								Сге	ate	Ne	w ((Colu	mn	Tol	Row	1)		
📧 CargoCapacity							3	Sat	isfy									2
🗈 FuelCapacity										h	5							3
🗈 PassengerCapacity																		3
📧 CargoCapacity																		2
Eco-Friendiness																		3
Emissions																		200

The requirement becomes satisfied and thus disappears from the contents of the smart package *Unsatisfied Requirements* and from the dependency matrix as well.

Case Study #3: Configuration management of the complex system - creating dynamic configuration catalogs

The efficient configuration management process is a challenge in the evolution of any industrial scale product family. Smart packages are a real-life out of the box solution supporting the configuration management approach.

Now study the case that illustrates the efficient management of the complex system configurations with the help of smart packages. We have a library (static package) of system components, which we need to see in several different views of the model, that is, catalogs, according to their characteristics. Using the catalogs will not extend the scope of the model, since they do not require duplication of the elements.



We will create two dynamic system configuration catalogs, that is, smart packages with dynamic contents, to gather the servers from the library *TI* Hardware according to the configuration version defined in a tag value of their specification.

Do the following:

1. Create two packages: TI12 Catalog and TI14 Catalog.

2. In each package, create a smart package named Servers.

🖃 🕂 📩 Hardware Library
🖻 🛅 TI 12 Catalog
Servers
🛱 🗁 🛅 TI 14 Catalog
Servers
🖻 🛅 TI Hardware
🗄 🖷 Cabinets
🗄 ··· 🛅 Displays
🗄 🖷 Servers
DELL R710
DELL R715
DELL R810
DELL R815
SUN FIRE T 1000
SUN FIRE T2000
SUN FIRE X2100 M2
SUN FIRE X2200 M2
🗄 🛅 Switches

3. Define criteria for gathering the contents of the smart package Servers in TI12 Catalog. Specify search options to find in the package TI Hardware, all block type elements with tag value Used In=TI12.

	Find 🕕		Remove
	Operation Nam	e: Find	
		-	
	What:	Q *	•
	Type:	Block	
	Scope:	TI Hardware	
	Properties: 🕕	Used In=TI12	
		Include elements from standard/system	profiles
		Search data unused in diagrams	
i	Define criteria for	gathering the contents of the smart package Ser	vers in TI14 Cata

4. Define criteria for gathering the contents of the smart package Servers in TI14 Catalog. Specify search options to find in the package TI Hardware, all block type elements with tag value Used In=TI14.

1	Find 🕤		Remove
(Operation Nam	e: Find	
1	What:	Q *	•
1	Гуре:	Block	
5	Scope:	TI Hardware	
F	Properties: 🕤	Used In=TI14	
		 Include elements from standard/system Search data unused in diagrams 	profiles



the block SUN FIRE T1000 appears in both TI12 Catalog and TI14 Catalog, since it has both tag values Used In=TI12 and Used In=TI14.

6. Also, you can add a block to a smart package manually. Just drag the block to the smart package.

Related Pages

- Specifying criteria for querying model
- Model Elements
- Stereotype
- Package
- Working with Profiles