Visu-IT! - Tools in ECU Development

Development Process



Franz Lohberger Jul-06 Page: 1

Visu-IT! – ECU Development Process Process flow



ADD – Getting Started Starting ADD



 \circledast All rights with Visu- II: CmbH, also for pending patent applications. All powers of disposition, such as copying and distribution, from Visu $\,$ II:

-IT!-

ADD – Getting Started Container Management



Container can be used to group a set of (calibration-)data. In ADD a **SW specification**/ **SW module** can be modeled by using a container.

ADD allows to:

- create, modify, maintain and delete containers
- create new versions and revisions of existing containers (version management)
- define different roles and lifecycles for the container management

	💱 ADD - Container Management				
	Container Extras Help				
	KMTR (V1.1.1 - fixed)				
	□- □ Container □- □ ■ ADD TEST	Container ID:	KMTR_1		
When opening a container the following	AirRow ThrAgDem AirRow ThrAgDem AirRow ThrAgDem Gaspatizung Carter Strang C	Container Name:	KMTR		
		Short Name:	this is kmtr		
		Based on Container:	KMTR_1.1.1		
	✓ KMTR (V1.1.1 - fixed)	Requestor:	Visu-ITI Status: 🗸 fixed		
	E E Test_Container	Description:	This version of the container introduces		
		Remark:			
			Changed		

ADD – Getting Started Data Objects (1)



The **Interface** of the container is controlled via the tree view on the left side. The (interface-)data can be "INPUT", "OUPTUT" and "LOCAL" (optional).

😽 ADD - Dataobject Definition	
Data Object Container Print Tools Check Help	
🗋 🔦 🗴 🍇 🛲 🗷 🥝	
Container KMTR (V1 1 1 - fixed)	
⊡È ✓ KMTR	map array (1.0, fixed)
tdef ✓ testCDefine (1.0, fixed)	
CAL V map_array (1.0, fixed)	
drebzabl (1.3, draft)	Name: map_array
on (1.2, fixed)	Display name: this is a map array
ONL online2 (1.0, fixed)	Ture
□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □	
tdef ✓ cDefine2 (1.0, fixed)	Status: V fixed V
myOnline2 (1.1, fixed)	X 8
01 √ 02 (1.0, fixed)	
	SIM Data Type
	Base Type: SLONG Y Phys. Unit: E Base Type: SLONG Y Phys. Unit: E Y
	Conv. Type: RationalFunction Conv. Type: RationalFunction
	Conv.: RATIONALFUNCTION_[0;1;0;0;0;1] Conv.: RATIONALFUNCTION_[0;1;0;0;0;1]
	Formatstring: 16.3
	AXIS Dim Avia Varrian Varrian Status Avia size SW Data Ture
	X axis array V1.0 fixed 7 RATIONALFUNCTION [0:1:0:0
	Y · · · · · · · · · · · · · · · · · · ·
1	

ADD – Getting Started Data Objects (2)

VISU^{IT!}

Data Objects (Calibration data):

- can be of the type: MEASUREMENT, AXIS, CHARACTERISTIC and System Constant (defines)
- can have both "SIM"- (simulation, e.g. in Matlab) und "SW"- (software, e.g. in C source code) attributes
- are controlled via a configurable version management (versions, revisions)
- are managed via a configurable lifecycle management ('draft', 'simulation_fixed', 'fixed' and 'obsolete')
 - follow a configurable user and rights management

To add further data objects to a container you can either

- search for existing data objects (see next slide)
- add/create new data objects (or a new version/revision of a data object) (see slide after next)

ADD – Getting Started Data Objects – Search for data objects



KMTR (V2)	in 2 2				
	MADD - Search for Data Obie	cts			X
	Data Object Name (Use Wildcards *agmAck* Display name (Use Wildcards * Ar	s* And ? OR a Cou nd ?)	mma Separated List Of Keywor	ds) Aut Ma Key	comatically With Like tch Case / Word Search comatically With Like tch Case
	Type ↓ cDefine ↓ ↓	Parameter	I Online	⊽ Axis	🔽 Мар
- A	Data Type				
• • • • • • • • • • • • • • • • • • •	Name		Phys. Unit	Base Type Formats	A A X
	Name Name Image: Name Image	Version 5 1.0 c 1.0 c 1.0 c	Phys. Unit State SIM Data Type draft RATIONALFUNCTION draft RATIONALFUNCTION draft RATIONALFUNCTION	Base Type Formats Phys. Unit [0:1 [-] [0:1 [-] [0:1 [-]	Start Search

Franz Lohberger Jul-06 Page: 7

ADD – Getting Started Data Objects – Create new data objects



ADD - Dataobject Definition		
ta Object Container Print Tools	Check Help	
Containe		
KMTR (V2.1.0 40)		
		(new definition)
B OUTPUT RXS axis_array	(1.1, draft)	Data Definition Data Object Information Languages
#def ✓ testCDefine	e (1.0, fixed) (1.1_draft)	Container specific Information
onL ✓ aNewOnlin	ne2 (1.1, fixed)	Name: Classification: OUTPUT
CollE_con ML ✓ drehzahl (1	ICACV_VVV (1.0, draft)	Display name: Limits: no limits
01 (1.2, fixe 01 √ 01 (1.2, fixe	ed) 0, fi 💱 VIT Name B	uilder
PAR p1 (1.0, for	ed) Name Builder:	Autosar 1.0 Pattern: Autosar
	Name:	CoME arAck
#def ✓ cDefine2 (#def ✓ myOnline2	(1. cc	
onL ✓ newInputO	onlin ed) CoME -	
		Phys. Unit:
		(tion V
		Abs (absolute value) Abs (absolute value) Ac (Air Conditioning)
		Ack (acknowledge)
		Act (actual)
		Add (additive) Adh (adhesion)
		Adj (adjustment) Adp (adaptive)
		Adv (advance (early)) Afl (air fuel lean)
		Afr (air fuel rich)

The optional "Name Builder/Checker" component helps applying naming conventions. In the example above, the component supports the **AUTOSAR** naming convention. The component is generic and freely configurable, thus it is also possible to support customer specific naming conventions.

ADD – Getting Started

<u>Use case:</u> Create a new function/module (1)



Steps in ADD:

- 1. Create a new container/module/specification Status: 'draft'
- 2. Define the interface of the container
 - define the INPUTs
 - -> reference existing data objects
 - e.g. select the correct ENGINE_SPEED from the global database for your function (e.g. 32bit, etc.)
 - specify the **OUTPUT**s and **LOCAL**s
 - -> use existing data objects (from previous or related functions) and/or add new data objects

The status of the new data objects is: 'draft'

3. "Simulate" your function

Before you can simulate your function, you have to set the simulation-related attributes (SIM-attributes). After that, you can set the status of the data objects to 'simulation fixed' and export the container and the data objects to Matlab in order to simulate the functionality.

-> see http://www.visu-it.de/add, section "Interfaces & Links", section "Matlab"

ADD – Getting Started

<u>Use case:</u> Create a new function/module (2)



4. "Implement/code" your function Before you can create real C code, you have to set the software-related attributes (SW-attributes). After that, you can set the status of the data objects to 'fixed' and export the container and the data objects to DDS (and Ascet and TargetLink) in order to code/implement the functionality.

Import the function into DD [*]	f:\dd	s_d' Intents: ADD - Impe	ort a cont	tainer	? 🛛
😴 DDS - DDS Project, Version		OPTIONS	TYPES	VALUES	~
Project Filters Configuration View Window Help		DDContainer	String	KMTR2.1.0	
🖹 🛱 🖬 Export Filter 🕨 🕨		DDContainerImportMode	Choice		
Import Filter Compare/Merge		DDProxyDefinitionsToUnit	String	ſ	
Other Import ASAP2		DDProxyAttributesToUnit	String		=
Editor Import IEEE-695 (I3E)		InputDefinitionsAndAttributesTerUnit	String		
Active Project Settings		InstrEntitiesBasedOnProxiesToUnit	String		
Application Settings ADD - Import a container	data	GenerateInstrEntitiesBasedOnProxies	Choice	CreationRule_ADDName	
Import Init Values (CVX 1.0)		GenerateFunctionBasedOnDDContainer	Bool	• Yes O No	
	9	ProjectDelivery	String		
	10	OverwriteConfigurationValues	Bool	O Yes No	
	11	KeepObsoleteDefinitions	Bool	O Yes No	
	12	ConvertObsoleteDDProxyAttributes	Bool	OYes	
	13	RemoveObsoleteDDProxyAttributes	Bool	O Yes No	
	14	SetDefaultLimits	Bool	O Yes No	
specify the name (version	15	DisplayNameExtension	String		
specify the name (version,	16	BitmaskForBooleans	Bool	OYes No	✓
revision) of the function	-DDCo -Gener	ntainer KMTR2.1.0 -GenerateInstrEntities ateFunctionBasedOnDDContainer	BasedOnPro	oxies CreationRule_ADDName	 N
			OK	Cancel <u>Save As</u>	<u>R</u> un Filter
					Franz Jul-06 Page: 10

ADD – Getting Started <u>Use case:</u> Create a new function/module (3)



Page: 11

😾 DDS - DDS Project, Version	
Project Filters Configuration View Window Help	
1 🛱 🚅 🔲 🖸 🗸 🛥 🛷 🖍 🤶 🥂	
😽 '~ADD_drehzahl' [online data]	
Erefor System View ▲	Definition Calib Init Used by Contain Sel-Lists Source Sections Desc
E⊷ E KMTR2.1.0	Conversion: LINEAR [0, 0H] [8000, FFH]
ADD_axis_array	Formatstring: Phys.Unit: [-]
-#def myOnline2	Online update index:
ADD_newinputonine	
⊡- tC P Output	Flags, Attributes
	Lower limit (hex): (phys):
~ADD_CoME_arAckAc	
ADD_LOIE_concAcv	opper limit (riex). j (priys). j
~ADD_drehzahl	
	Bit mask:
ADD_online2	
⊕ • ttp* Local	
	< []

<u>Note:</u> In DDS, all data objects and attributes which are defined and specified in ADD are shown in green color and are readOnly! ADD is the 'master' for these attributes.

5. Container lifecycle

Similar to data objects, the container is/can be also set to 'simulation fixed' when the simulation has been successfully done. The container can be set to 'fixed' when all data objects of the container are also set to 'fixed'.

ADD – Getting Started

Use case: Create a new version/revision of a function

Steps in ADD:

- 1. Create a new version/revision of an existing container Status: 'draft'
- 2. Define the (new) interface of the container, add new data objects, remove obsolete data objects, create new versions/revisions of existing data objects etc. Note: When only a new revision of the container is created, some modifications are not allowed.

3. "Simulate" your function ...identical to the use case "Create a new function/module"

4. "Implement/code" your function

...identical to the use case "Create a new function/module" Note that DDS provides a smart "Upgrade" mechanism!

5. Container lifecycle

...identical to the use case "Create a new function/module"