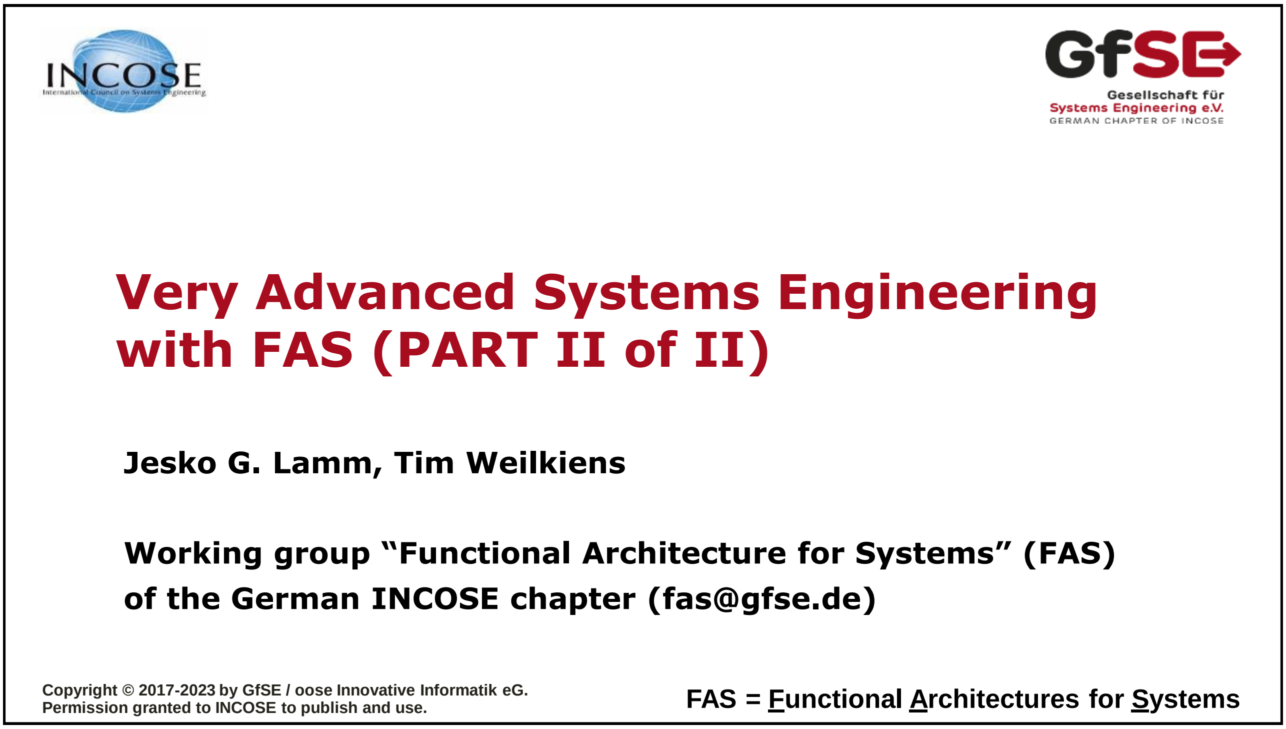
Very Advanced Systems Engineering with FAS (PART II of II)

Tim Weilkiens, Jesko G. Lamm

FAS Working Group of the German Chapter of INCOSE (GfSE)



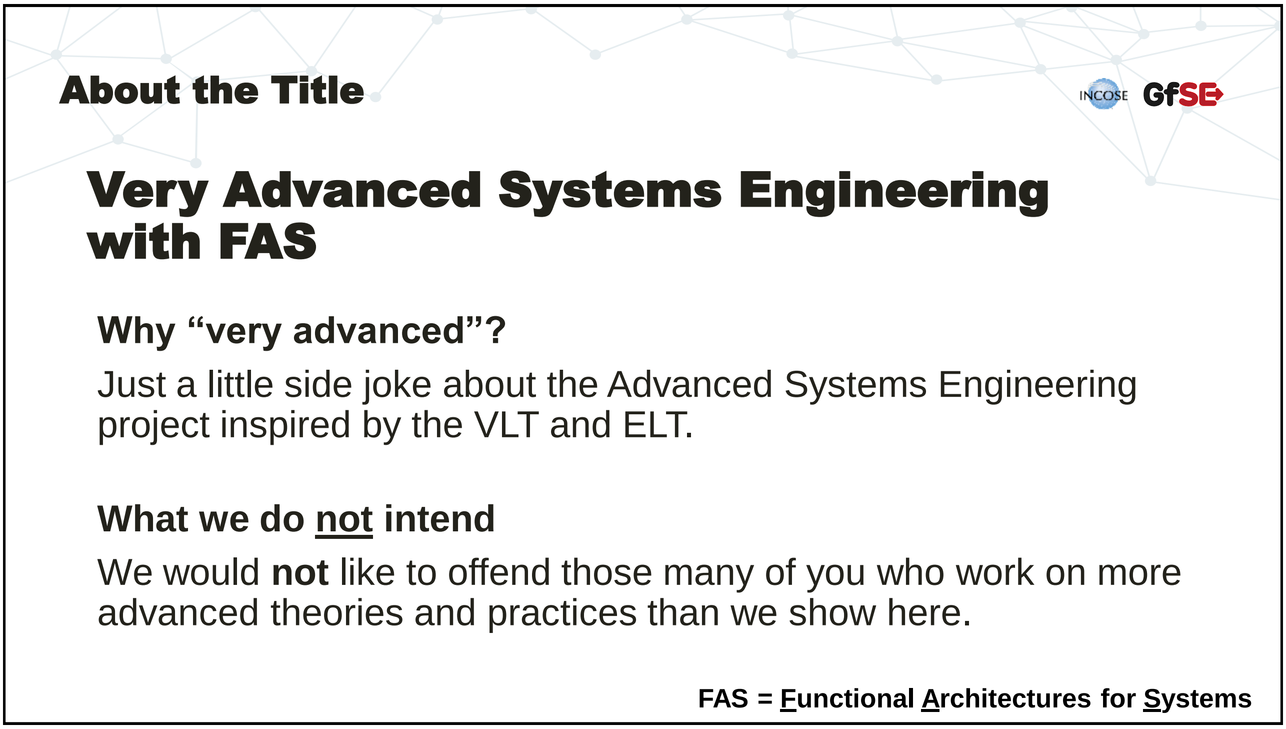
Very Advanced Systems Engineering with FAS (PART II of II)

Jesko G. Lamm, Tim Weilkiens

Working group "Functional Architecture for Systems" (FAS) of the German INCOSE chapter (fas@gfse.de)

FAS = Functional Architectures for Systems

1



About the Title

Very Advanced Systems Engineering with FAS

Why "very advanced"?

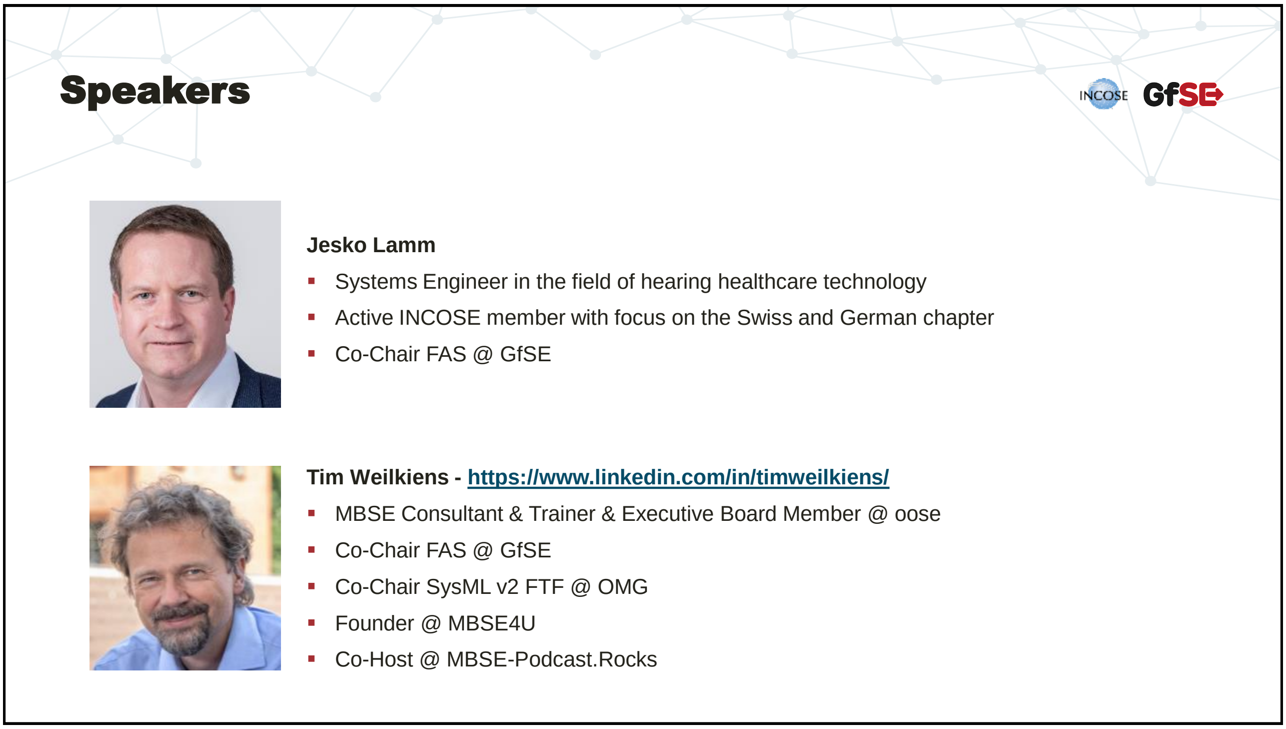
Just a little side joke about the Advanced Systems Engineering project inspired by the VLT and ELT.

What we do not intend

We would **not** like to offend those many of you who work on more advanced theories and practices than we show here.

FAS = Functional Architectures for Systems

2



Speakers

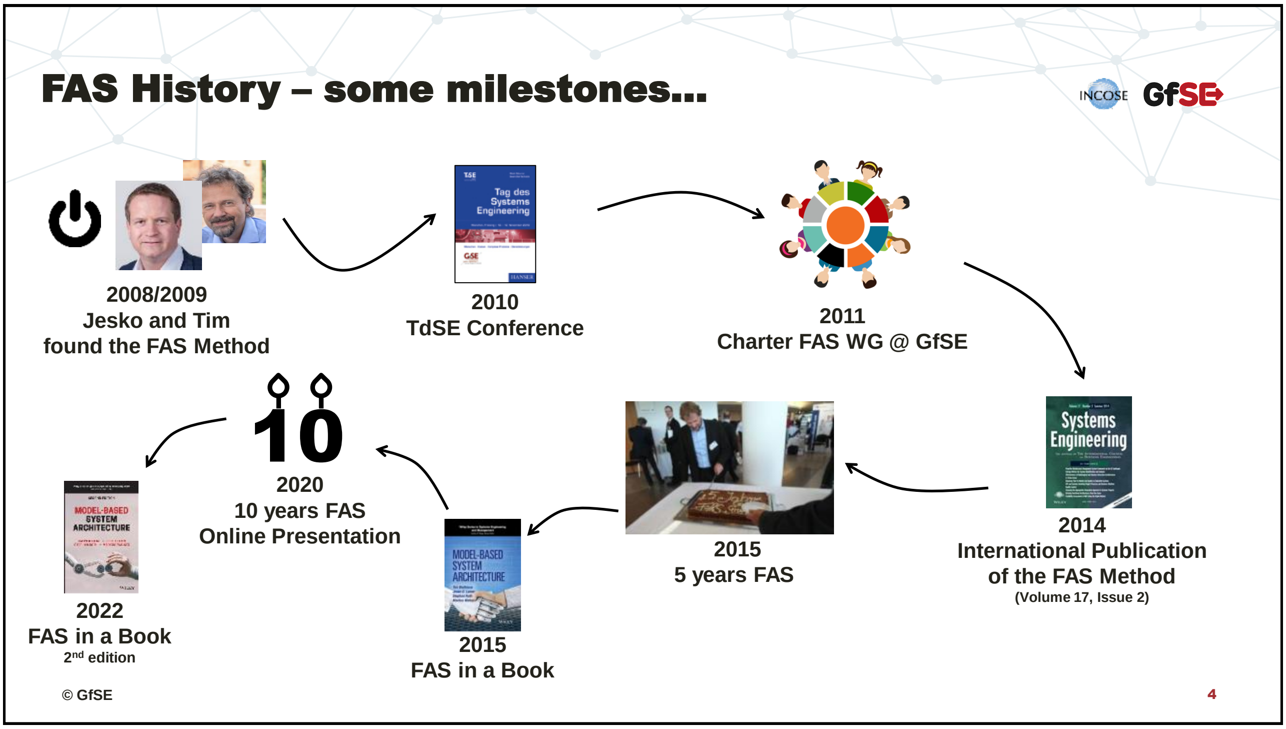
Jesko Lamm

- Systems Engineer in the field of hearing healthcare technology
- Active INCOSE member with focus on the Swiss and German chapter
- Co-Chair FAS @ GfSE

Tim Weilkiens - https://www.linkedin.com/in/timweilkiens/

- MBSE Consultant & Trainer & Executive Board Member @ oose
- Co-Chair FAS @ GfSE
- Co-Chair SysML v2 FTF @ OMG
- Founder @ MBSE4U
- Co-Host @ MBSE-Podcast.Rocks

3



FAS History – some milestones…

2008/2009
Jesko and Tim found the FAS Method

2010
TdSE Conference

2011
Charter FAS WG @ GfSE

2014
International Publication of the FAS Method
(Volume 17, Issue 2)

2015
5 years FAS

2015
FAS in a Book

2020
10 years FAS
Online Presentation

2022
FAS in a Book
2nd edition

© GfSE

4

4

FAS Working Group of GfSE (=German chapter of INCOSE)

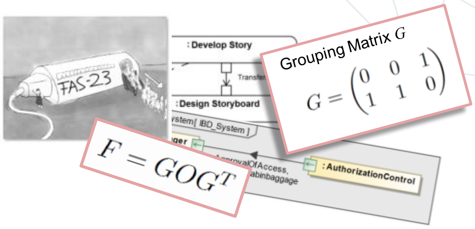
- Chartered as GfSE working group in 2011
- The working group shares experiences in working with functional architectures and aims to improve the available methods, especially FAS.
- Online meetings around 7x per year
- Webinars (like today)

© GfSE

Content of the Webinars

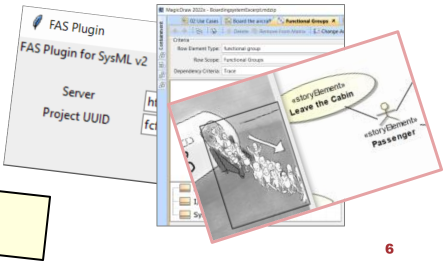
- PART I: Concepts / theory
  - Overview Functional Architectures
  - The FAS Method
  - Introducing an example system
  - Different representations content when working with the FAS Method
  - The so-called „SAMS Method“ as add-on

DONE (Oct 11th)



- PART II: Application in practice / demos
  - Recap
  - Working with the above on paper and in tools
  - Recommendations for daily work in workshops and in the back-office

TODAY



Patient Information

Prescription
for Happy Engineering team

3 Use Case Analyses
by Workshops of 1h

Apply before solution design.

2.2.1.2020
Th.Dr.

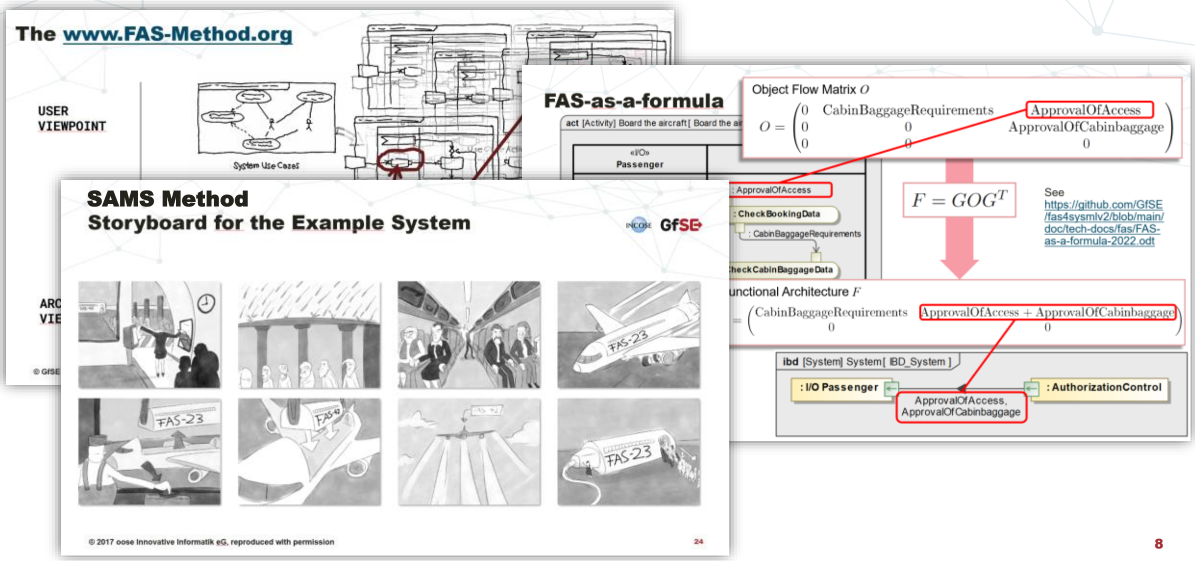
Risks and side effects

As a Systems Engineer, you should first clarify how you will contribute to the success of the company.

Accordingly, the methods in systems engineering should be chosen. These can then be supported by tools if required.

If you use the methods and tools mentioned here without evidence of their contribution to your company's success, you run the risk of an ulcer.

Recap of Part I


The www.FAS-Method.org

SAMS Method
Storyboard for the Example System

FAS-as-a-formula

Object Flow Matrix O

$$O = \begin{pmatrix} 0 & \text{CabinBaggageRequirements} & \boxed{\text{ApprovalOfAccess}} \\ 0 & 0 & \text{ApprovalOfCabinbaggage} \\ 0 & 0 & 0 \end{pmatrix}$$

$F = GOG^T$

See https://github.com/GfSE/fas4sysmlv2/blob/main/doc/tech-docs/fas/FAS-as-a-formula-2022.odt

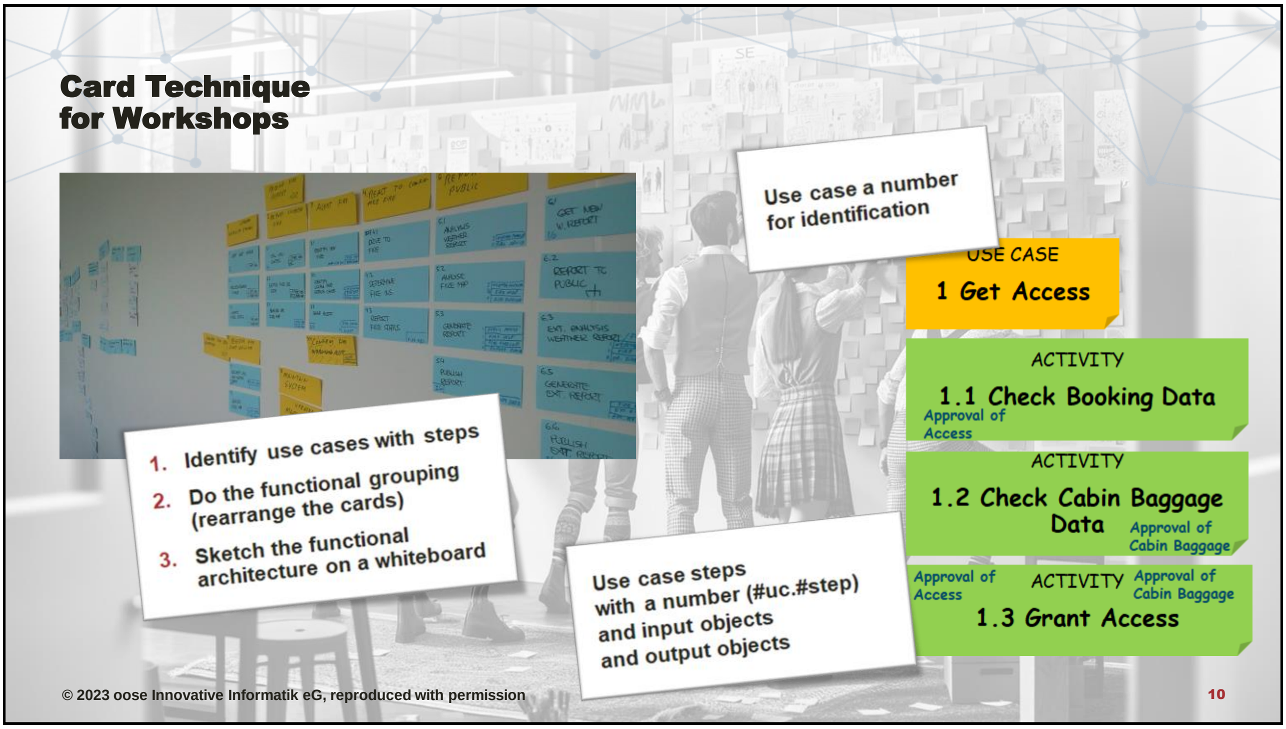
unctional Architecture F

$$= \begin{pmatrix} \text{CabinBaggageRequirements} & \boxed{\text{ApprovalOfAccess} + \text{ApprovalOfCabinbaggage}} \\ 0 & 0 \end{pmatrix}$$

© 2017 oose Innovative Informatik eG, reproduced with permission

Card Technique for Workshops

9



Card Technique for Workshops

1. Identify use cases with steps
2. Do the functional grouping (rearrange the cards)
3. Sketch the functional architecture on a whiteboard

Use case a number for identification

USE CASE
1 Get Access

ACTIVITY
1.1 Check Booking Data
Approval of Access

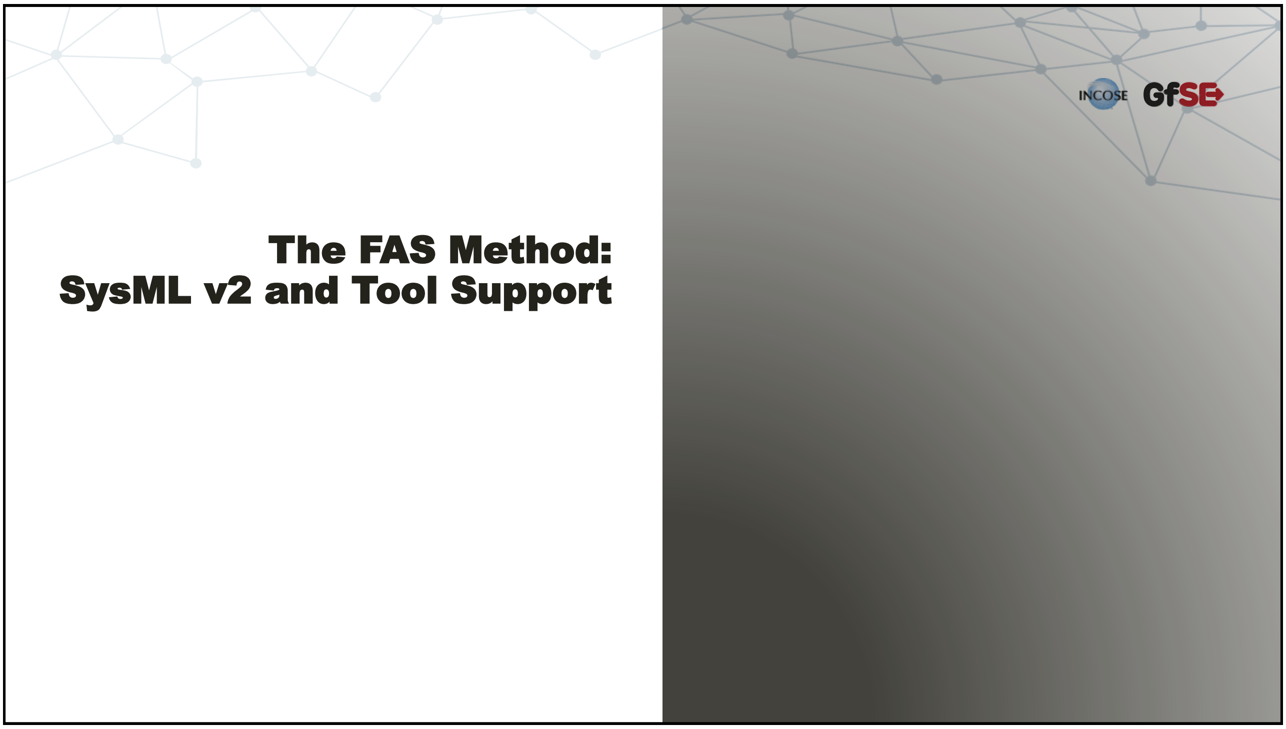
ACTIVITY
1.2 Check Cabin Baggage Data
Approval of Cabin Baggage

Approval of Access
ACTIVITY
Approval of Cabin Baggage
1.3 Grant Access

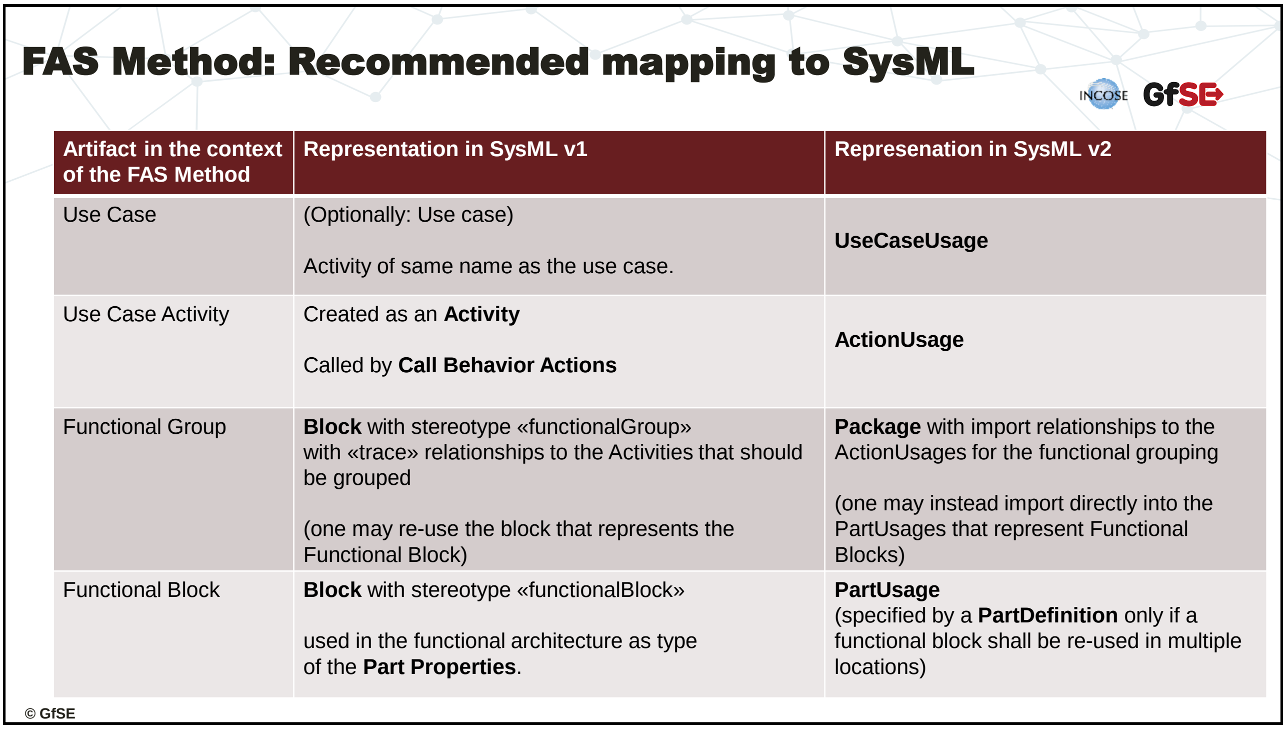
Use case steps with a number (#uc.#step) and input objects and output objects

© 2023 oose Innovative Informatik eG, reproduced with permission

10



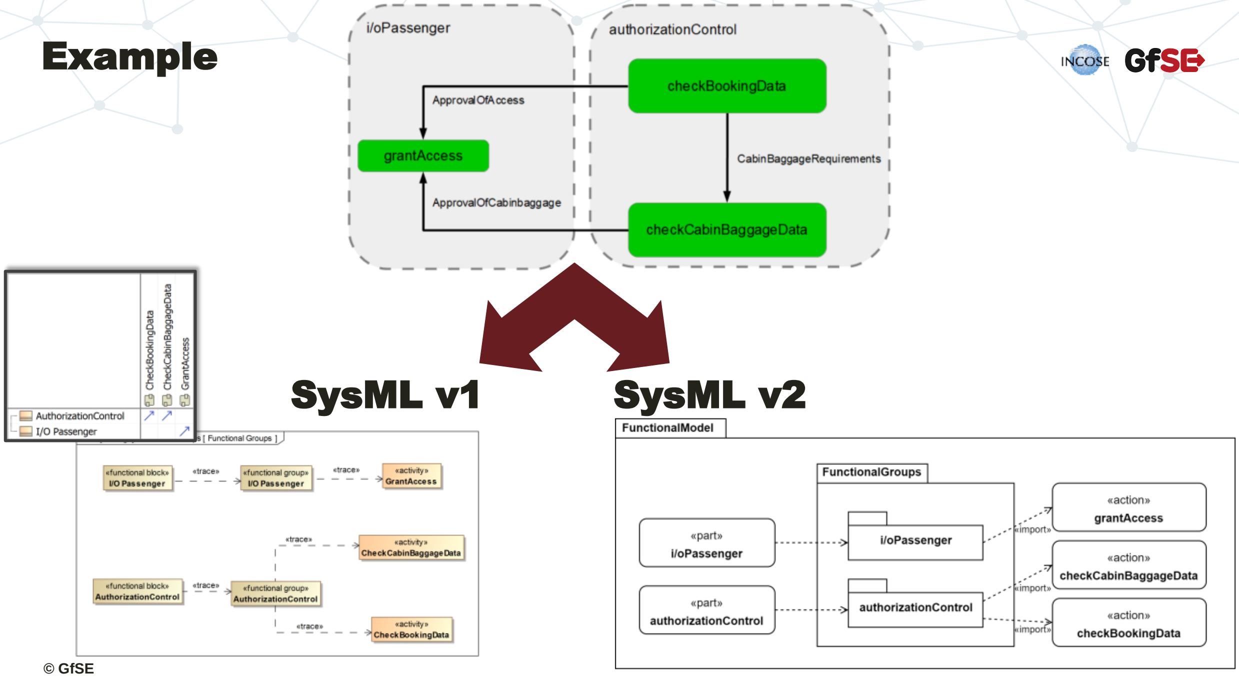
The FAS Method: SysML v2 and Tool Support



FAS Method: Recommended mapping to SysML

Artifact in the context of the FAS Method	Representation in SysML v1	Represenation in SysML v2
Use Case	(Optionally: Use case) Activity of same name as the use case.	**UseCaseUsage**
Use Case Activity	Created as an **Activity** Called by **Call Behavior Actions**	**ActionUsage**
Functional Group	**Block** with stereotype «functionalGroup» with «trace» relationships to the Activities that should be grouped (one may re-use the block that represents the Functional Block)	**Package** with import relationships to the ActionUsages for the functional grouping (one may instead import directly into the PartUsages that represent Functional Blocks)
Functional Block	**Block** with stereotype «functionalBlock» used in the functional architecture as type of the **Part Properties**.	**PartUsage** (specified by a **PartDefinition** only if a functional block shall be re-used in multiple locations)

Example

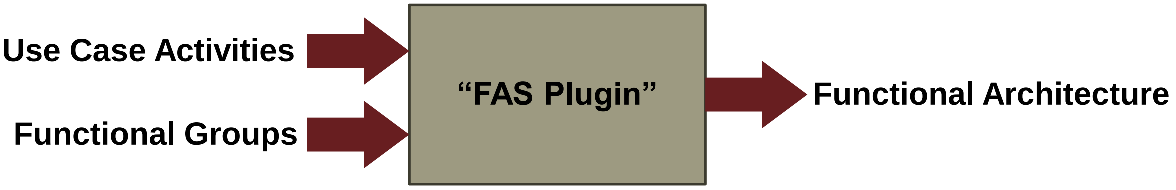


	CheckBookingData	CheckCabinBaggageData	GrantAccess
AuthorizationControl	↗	↗	
I/O Passenger			↗

© GfSE

$F = GOG^T$

Automatic Creation of the Functional Architecture

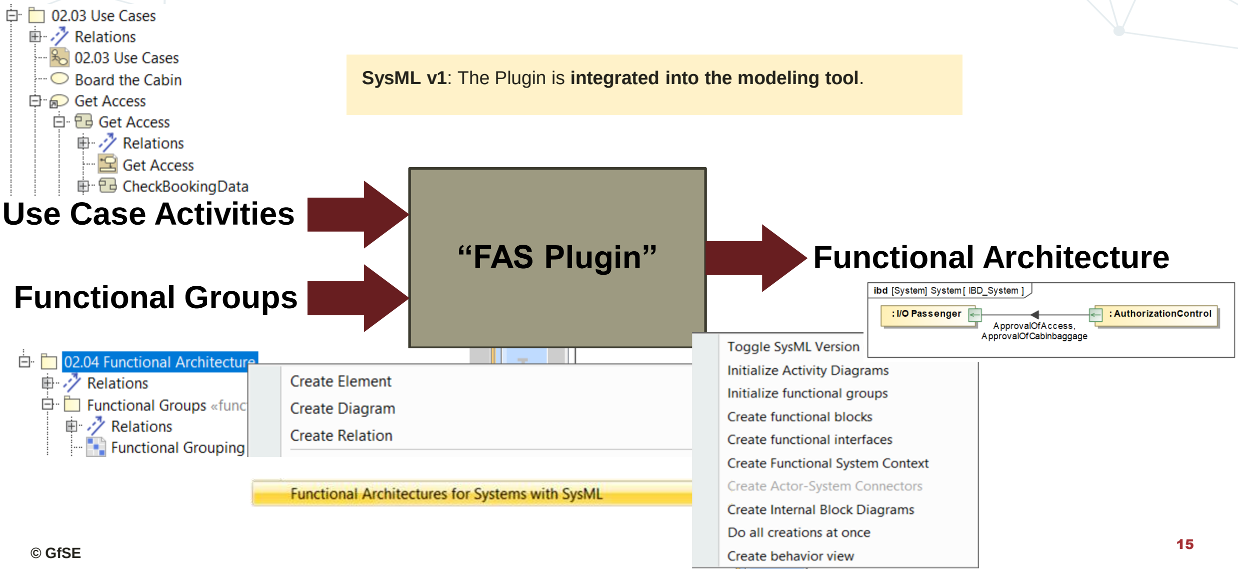


© GfSE

Automatic Creation of the Functional Architecture

$F = GOG^T$

SysML v1: The Plugin is **integrated into the modeling tool**.



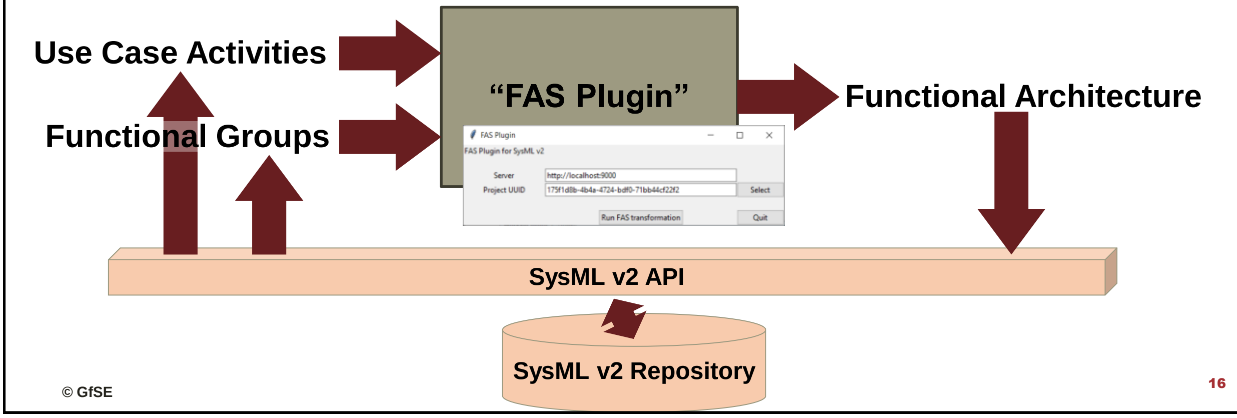
© GfSE

15

Automatic Creation of the Functional Architecture

$F = GOG^T$

SysML v2: We have not worked with an off-the-shelf modeling tool yet.
The plugin works with the **reference implementation** of the standardized API.



FAS Plugin for SysML v2

Server: http://localhost:9000
Project UUID: 175f1d8b-4b4a-4724-bdf0-71bb44cf22f2

© GfSE

16

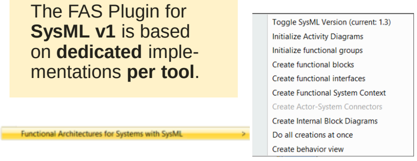
FAS Plugin Implementations

- **SysML v1**

  FAS Plugin for MagicDraw: https://sourceforge.net/projects/fas4md/

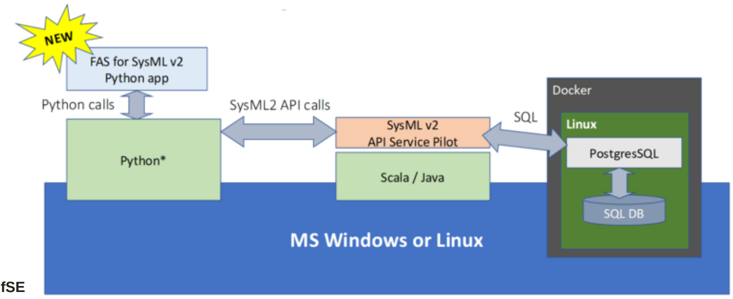
  FAS Plugin for Enterprise Architect: https://sourceforge.net/projects/fas4ea/

The FAS Plugin for **SysML v1** is based on **dedicated** implementations **per tool**.

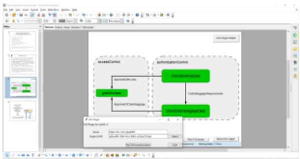

Functional Architectures for Systems with SysML >
Toggle SysML Version (current: 1.3)
Initialize Activity Diagrams
Initialize functional groups
Create functional blocks
Create functional interfaces
Create Functional System Context
Create Actor-System Connectors
Create Internal Block Diagrams
Do all creations at once
Create behavior view

- **SysML v2**

  First prototype of the FAS Plugin for SysML v2: https://github.com/GfSE/fas4sysmlv2 or https://sourceforge.net/projects/fas4sysmlv2/



The FAS Plugin for **SysML v2** is intended to be **tool-independent**. It supports visualization of its operation in the **FAS-as-a-formula** notation. It is accompanied by some OpenOffice macros + some scripts for filling the repository with input data and for visualizing output data.



© GfSE

17

17

Demo

You can reproduce all demos yourself, with freely available resources.

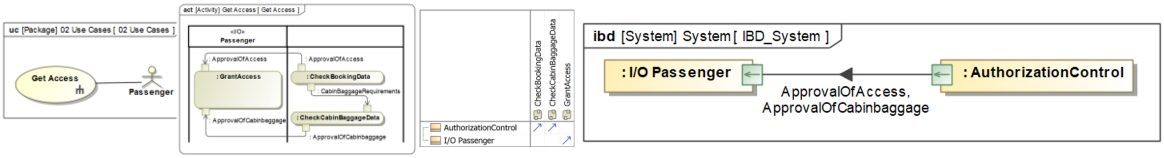
Here is a link to the used model files and to presenter notes, describing how to execute the demos:

https://fas-method.org/content/wp-content/uploads/2023/10/2023-11-08_Webinar_Demo.zip

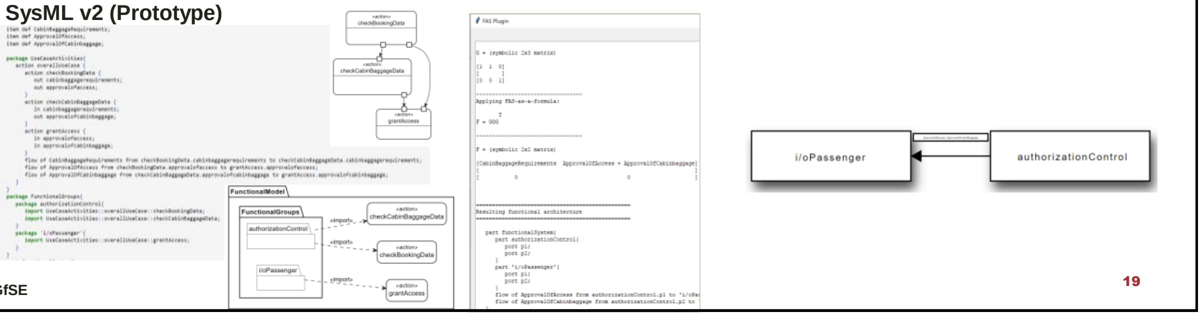
18

Demo: Automatic Creation of the Functional Architecture

- SysML v1



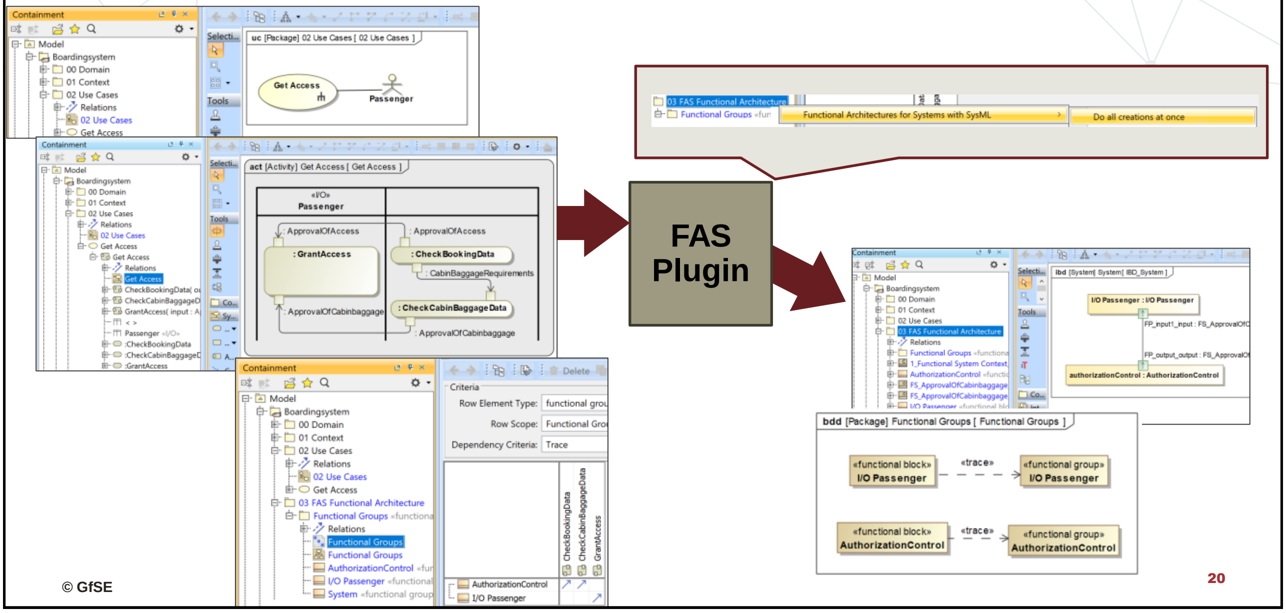
- SysML v2 (Prototype)



```
item def CabinBaggageRequirements;
item def ApprovalOfAccess;
item def ApprovalOfCabinbaggage;

package UseCaseActivities{
   action overallUseCase {
      action checkBookingData {
         out cabinbaggagerequirements;
         out approvalofaccess;
      }
      action checkCabinBaggageData {
         in cabinbaggagerequirements;
         out approvalofcabinbaggage;
      }
      action grantAccess {
         in approvalofaccess;
         in approvalofcabinbaggage;
      }
      flow of CabinBaggageRequirements from checkBookingData.cabinbaggagerequirements to checkCabinBaggageData.cabinbaggagerequirements;
      flow of ApprovalOfAccess from checkBookingData.approvalofaccess to grantAccess.approvalofaccess;
      flow of ApprovalOfCabinbaggage from checkCabinBaggageData.approvalofcabinbaggage to grantAccess.approvalofcabinbaggage;
   }
}
package FunctionalGroups{
   package authorizationControl{
      import UseCaseActivities::overallUseCase::checkBookingData;
      import UseCaseActivities::overallUseCase::checkCabinBaggageData;
   }
   package 'i/oPassenger'{
      import UseCaseActivities::overallUseCase::grantAccess;
   }
}
```

```
G = (symbolic 2x3 matrix)

[1  1  0]
[       ]
[0  0  1]

--------------------------------
Applying FAS-as-a-formula:

      T
F = GOG

--------------------------------

F = (symbolic 2x2 matrix)

[CabinBaggageRequirements  ApprovalOfAccess + ApprovalOfCabinbaggage]
[                                                                    ]
[           0                               0                        ]


================================================
Resulting functional architecture
================================================

  part functionalSystem{
     part authorizationControl{
        port p1;
        port p2;
     }
     part 'i/oPassenger'{
        port p1;
        port p2;
     }
     flow of ApprovalOfAccess from authorizationControl.p1 to 'i/oPa
     flow of ApprovalOfCabinbaggage from authorizationControl.p2 to
```

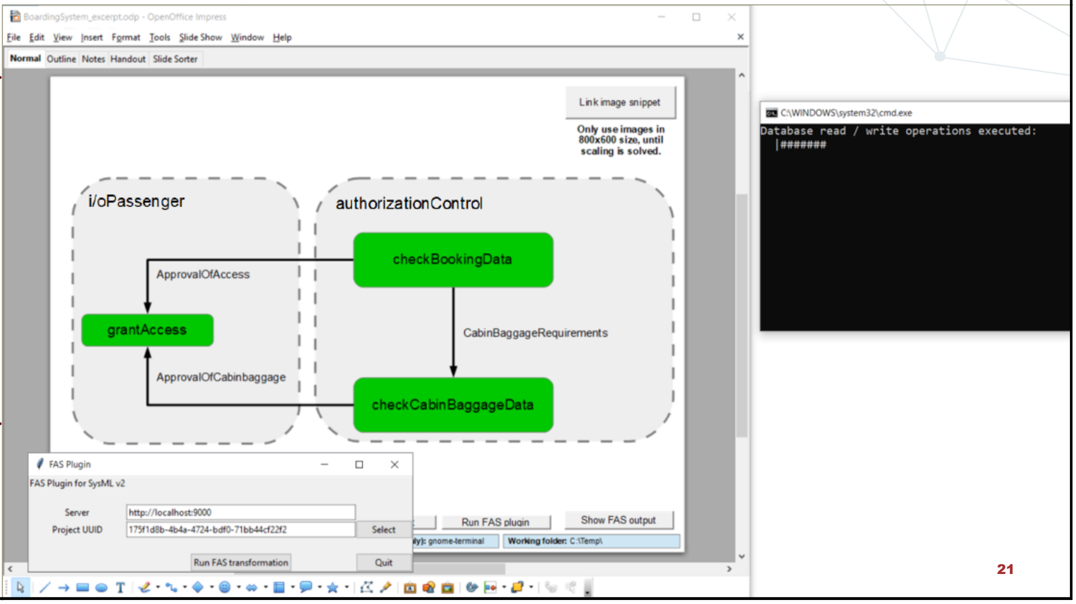
Demo: FAS Plugin for SysML v1


Containment
Model
Boardingsystem
00 Domain
01 Context
02 Use Cases
Relations
02 Use Cases
Get Access
uc [Package] 02 Use Cases [02 Use Cases]
Get Access
Passenger
CheckBookingData
CheckCabinBaggageData
GrantAccess
Passenger «I/O»
:CheckBookingData
:CheckCabinBaggageData
:GrantAccess
act [Activity] Get Access [Get Access]
«I/O» Passenger
: ApprovalOfAccess
: GrantAccess
: CheckBookingData
: CabinBaggageRequirements
: ApprovalOfCabinbaggage
: CheckCabinBaggage Data
03 FAS Functional Architecture
Functional Groups «functional
Functional Groups
AuthorizationControl «functional
I/O Passenger «functional
System «functional group
Delete
AuthorizationControl
I/O Passenger
FAS Plugin
Functional Groups «fun
Functional Architectures for Systems with SysML
Do all creations at once
1_Functional System Context
FS_ApprovalOfCabinbaggage
ibd [System] System [IBD_System]
I/O Passenger : I/O Passenger
FP_input1_input : FS_ApprovalOf
FP_output_output : FS_ApprovalOf
authorizationControl : AuthorizationControl
bdd [Package] Functional Groups [Functional Groups]
«functional block» I/O Passenger
«functional group» I/O Passenger
«functional block» AuthorizationControl
«functional group» AuthorizationControl
«trace»

Criteria
Row Element Type: functional grou
Row Scope: Functional Gro
Dependency Criteria: Trace

Demo: FAS Plugin for SysML v2

The graphical parts of our demo implementation will hopefully be replaced by off-the-shelf SysML v2 tools.



Only use images in 800x600 size, until scaling is solved.

```
Database read / write operations executed:
  |#######
```

FAS Plugin

FAS Plugin for SysML v2

Server: http://localhost:9000

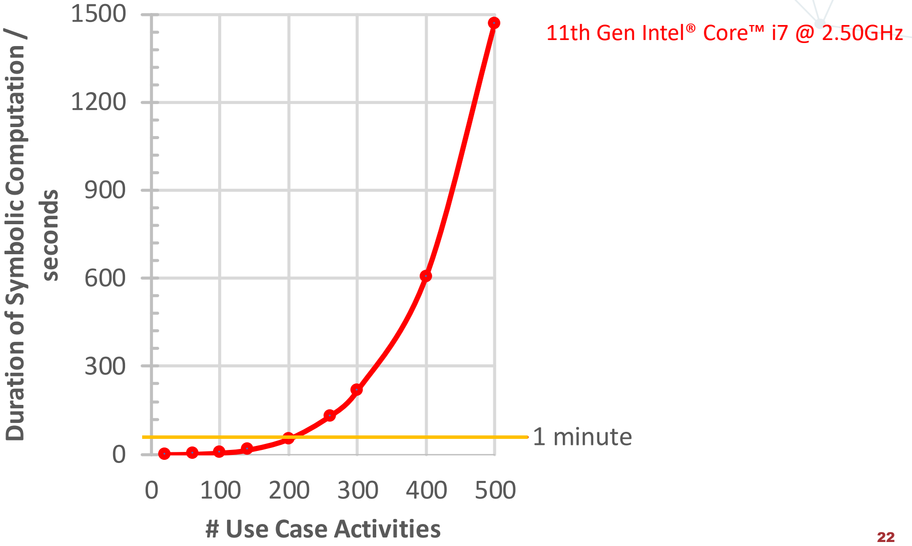
Project UUID: 175f1d8b-4b4a-4724-bdf0-71bb44cf22f2

...y): gnome-terminal Working folder: C:\Temp\

© GfSE

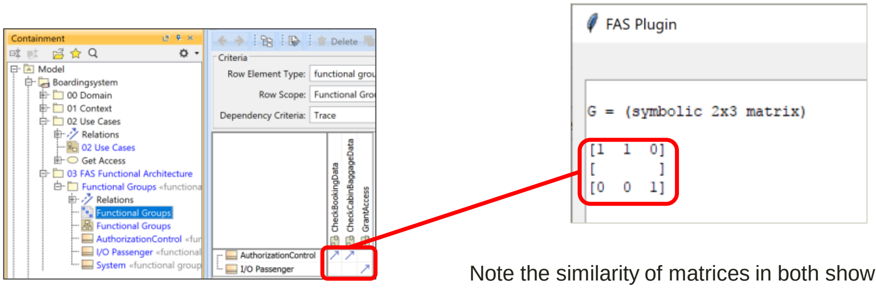
Symbolic Computation: Easy Formulation of the Job, but ...

$$F = GOG^T$$



© GfSE

Final Remark About the FAS Method



Containment

- Model
  - Boardingsystem
    - 00 Domain
    - 01 Context
    - 02 Use Cases
      - Relations
      - 02 Use Cases
      - Get Access
    - 03 FAS Functional Architecture
      - Functional Groups «functiona
        - Relations
        - Functional Groups
        - Functional Groups
        - AuthorizationControl «fur
        - I/O Passenger «functional
        - System «functional group

Criteria

Row Element Type: functional grou

Row Scope: Functional Gro

Dependency Criteria: Trace

	CheckBookingData	CheckCabinBaggageData	GrantAccess
AuthorizationControl	↗	↗	
I/O Passenger			↗

FAS Plugin

```
G = (symbolic 2x3 matrix)

[1  1  0]
[       ]
[0  0  1]
```

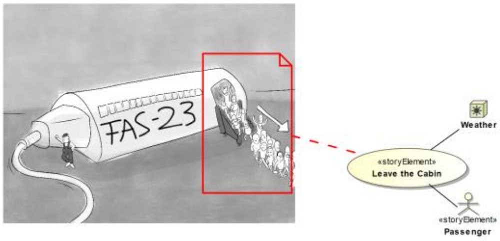
Note the similarity of matrices in both shown implementations (which is there, because row and column elements are sorted alphabetically in both implementations).

The SAMS Method

Tool Support for the SAMS Method

SysML v1

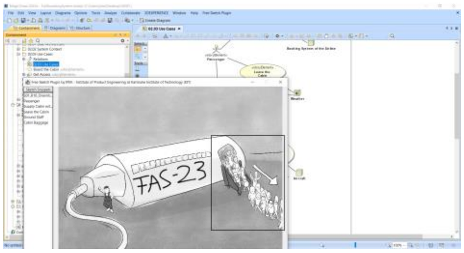
Option #1: Transparent comments on the diagram



+: Persisted in the repository

-: Only works on diagrams and not in direct connection with the described model element

Option #2: Dedicated plugin
http://sourceforge.net/projects/freesketches-for-magicdraw/



+: Direct linking with model elements

+: Full compliance with the data model of the SAMS method

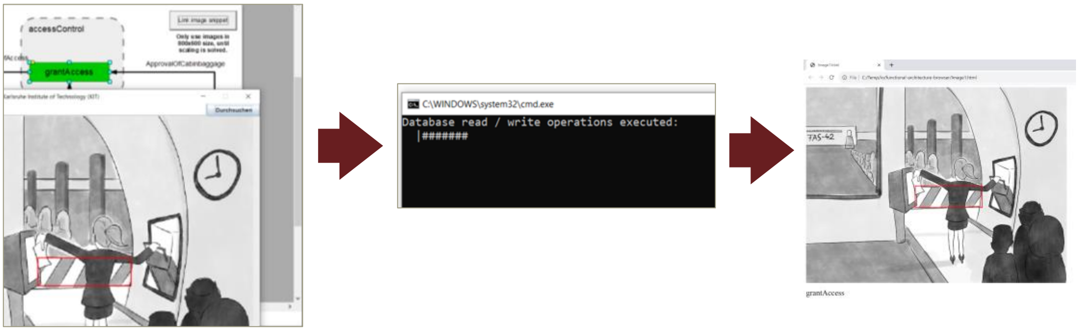
-: No implementation yet for persisting images in the repository

Tool Support for the SAMS Method

SysML v2

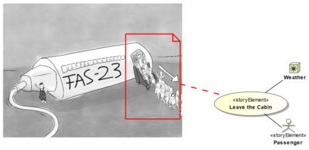
A first solution has been prototyped, including persistence in the repository.

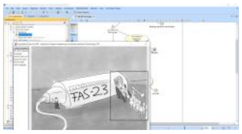
Since use cases are not yet implemented in the SysML v2 prototype, we temporarily link image snippets directly to use case activities.

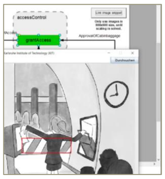

accessControl
grantAccess
Link image snippet
Only use images in 800x600 size, until scaling is solved.
ApprovalOfCabinbaggage
Karlsruhe Institute of Technology (KIT)
Durchsuchen
C:\WINDOWS\system32\cmd.exe
FAS-42
grantAccess

```
Database read / write operations executed:
 |#######
```

Demo of Tool Support for the SAMS Method







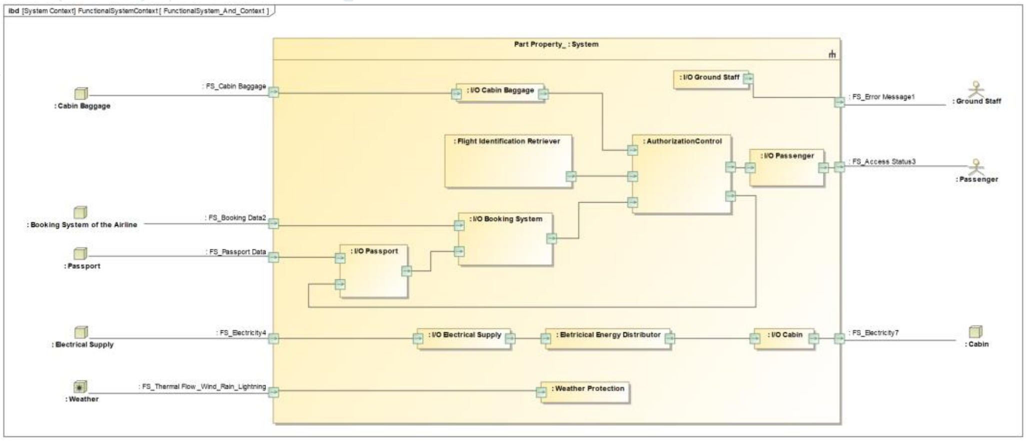
Storyboard images © 2017 oose Innovative Informatik eG. Rest of the slide: © GfSE

Functional Models with Higher Complexity

An Example of Automation Benefits with MBSE

Boarding System’s Full Functional Architecture

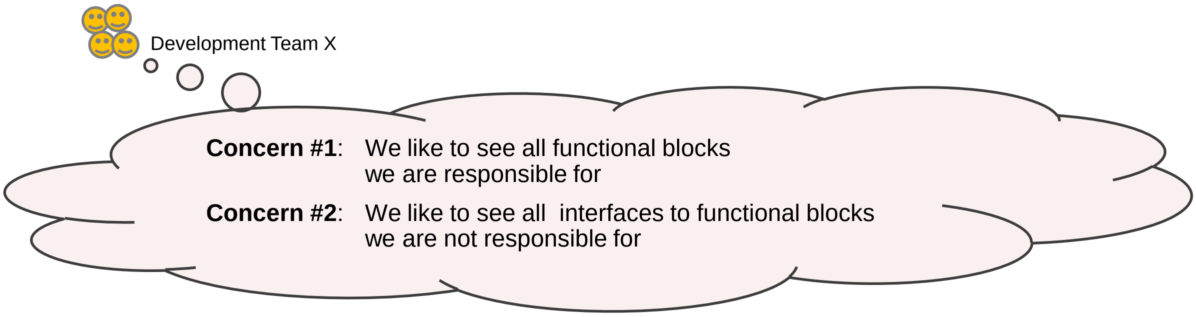
(as far as modeled in our little working group study)



- This simple example is already quite complex to overlook.
- Real-world systems are usually considerably more complex.
- Overlooking everything at once may not be needed. Usually, the model will be viewed regarding one concern at a time.

  => It is the strength of MBSE to support the generation of the corresponding views

Example Concerns



Concern #1: We like to see all functional blocks we are responsible for

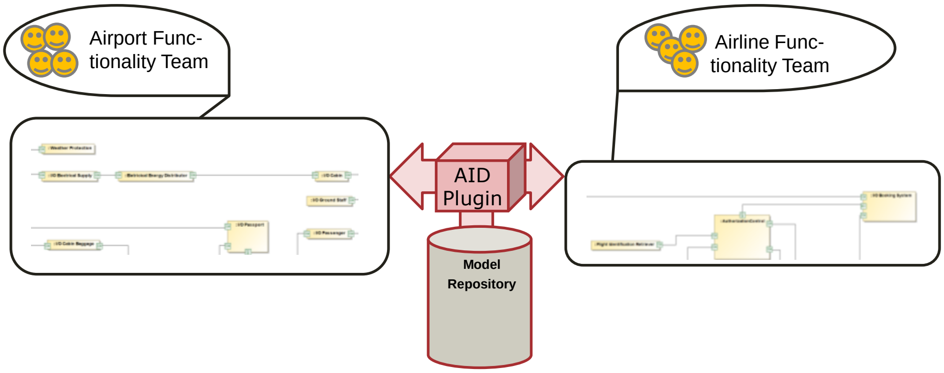
Concern #2: We like to see all interfaces to functional blocks we are not responsible for

Adapted from: Gerritsen W., Lamm J.G., Neitzel K.E., Scheithauer A., Webel C., Weilkiens T.: "Zielgerichtete Modellierung und stets aktuelle Views durch ein präzises Viewpoint-Konzept im MBSE", in Schulze, S.-O.; Tschirner, C.; Kaffenberger, R.; Ackva, S. (Eds.): Tag des Systems Engineering 2019 München 6.-8. November 2019, Gesellschaft für Systems Engineering e.V., Bremen, Germany, 2019, pp. 23-32

AID Plugin

https://sourceforge.net/projects/aid4md/

AID = Automated Instrument for Diagrams



🟨 = Functionality within the team's responsibility **[Concern #1]**

© GfSE

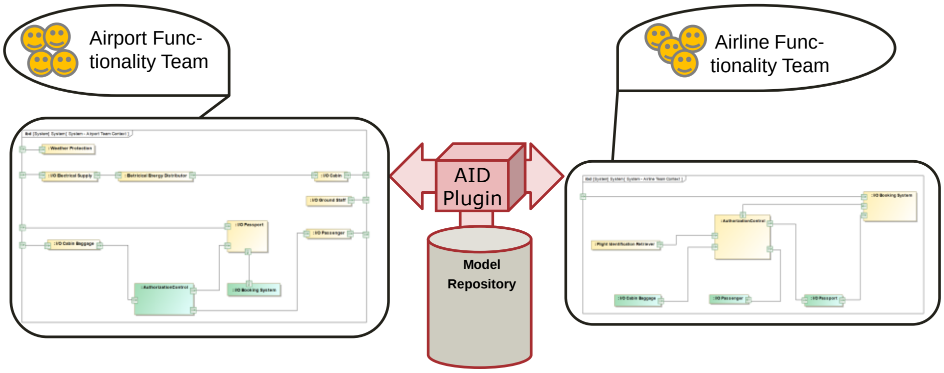
31

31

AID Plugin

https://sourceforge.net/projects/aid4md/

AID = Automated Instrument for Diagrams



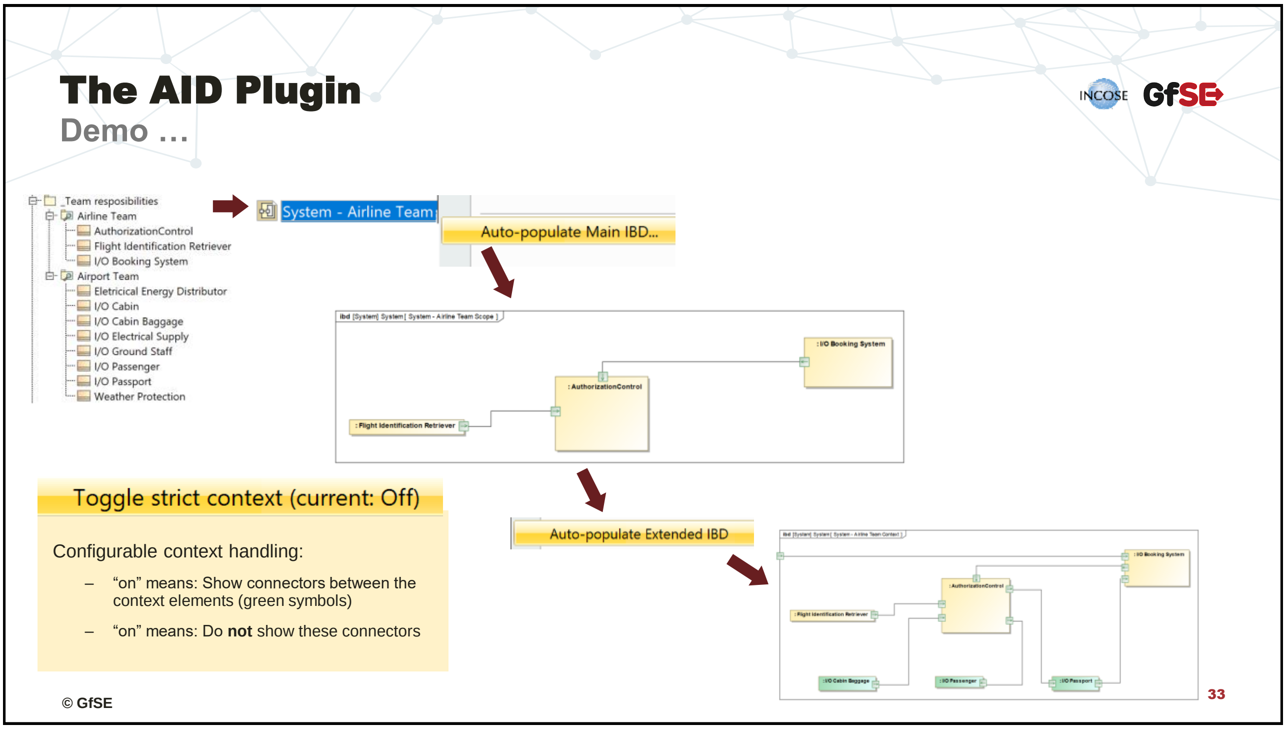
🟨 = Functionality within the team's responsibility **[Concern #1]**

🟩 = Functionality outside the team's responsibility with an interface to the above functionality **[Concern #2]**

© GfSE

32

32



The AID Plugin

Demo …

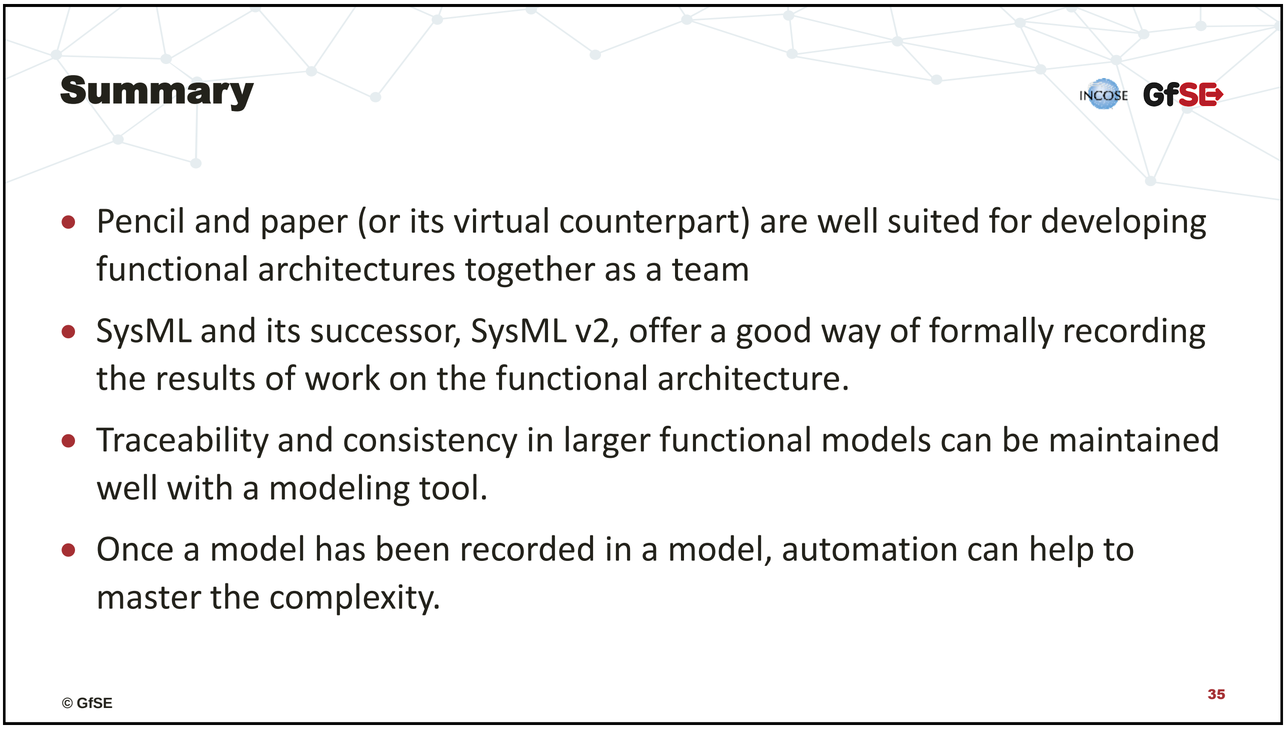
Toggle strict context (current: Off)

Configurable context handling:

- "on" means: Show connectors between the context elements (green symbols)
- "on" means: Do **not** show these connectors

© GfSE

Summary



Summary

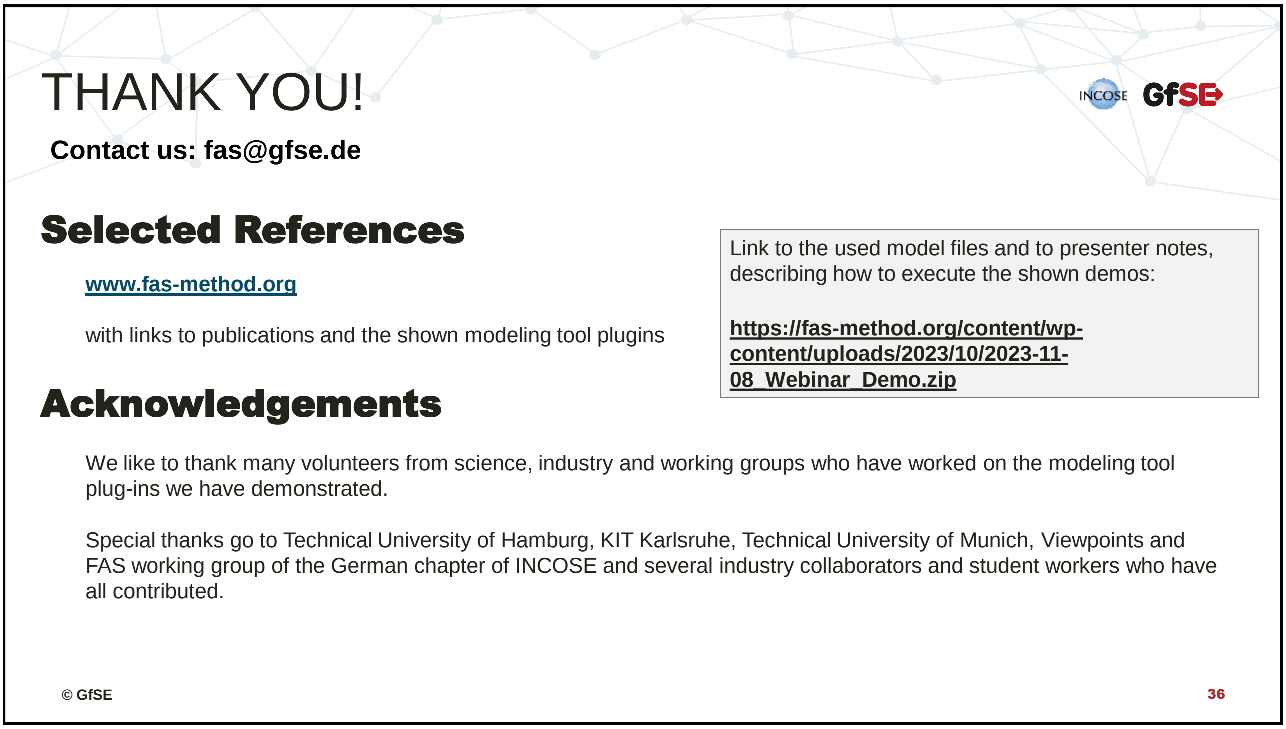
INCOSE GfSE

- Pencil and paper (or its virtual counterpart) are well suited for developing functional architectures together as a team
- SysML and its successor, SysML v2, offer a good way of formally recording the results of work on the functional architecture.
- Traceability and consistency in larger functional models can be maintained well with a modeling tool.
- Once a model has been recorded in a model, automation can help to master the complexity.

© GfSE

35

35



THANK YOU!

INCOSE GfSE

Contact us: fas@gfse.de

Selected References

www.fas-method.org

with links to publications and the shown modeling tool plugins

Link to the used model files and to presenter notes, describing how to execute the shown demos:

https://fas-method.org/content/wp-content/uploads/2023/10/2023-11-08_Webinar_Demo.zip

Acknowledgements

We like to thank many volunteers from science, industry and working groups who have worked on the modeling tool plug-ins we have demonstrated.

Special thanks go to Technical University of Hamburg, KIT Karlsruhe, Technical University of Munich, Viewpoints and FAS working group of the German chapter of INCOSE and several industry collaborators and student workers who have all contributed.

© GfSE

36

36