Versions/Variants Management –
The Dilemma of Choice between
multiplying Objects and/or multiplying Attributes

Dr. Bernd GRAHLMANN
Bernd@Grahlmann.net
www.grahlmann.net

Agenda

1. Bio highlights;
2. Goals of the Presentation;
3. Versions / Variants and Platforms of (Systems of) Systems, Sub-Systems, Components, …
4. Versions of Specifications, Requirements, V&V Cases, Features, …
5. Challenges;
6. Requirements and Goals wrt. Efficient and Effective Versions/Variants Management and Re-Use;
7. Version/Variant specific Attributes;
8. Version/Variant specific Traceability;
9. Tool Factor & Human Factor;
10. Very brief look at a tool implementation;
11. Questions & Answers (Q&A).

Dr. Bernd GRAHLMANN – Bio Highlights / Background …

1. Computer Science & Medicine background (Software for automatic diagnostic of the human hip based on 3D computed tomography data + tools for operation simulation);
2. Project Director – Tool for modelling, simulation and verification of parallel systems (30 programmers, 500.000 lines of code, distributed worldwide, SUSE Linux …);
3. 3 years Global Manager DOORS & Requirements Management GE Medical (2000+ engineers, process, guidelines, client/server installations, training, support, templates, project setup and migrations, coaching, evangelist, …);
4. **20+ years in various industries** (such as medical devices, railway, automotive, space, aviation, aerospace, defense, energy, banking, pharma, semiconductors, software, elevators, building, gaming, ...) successfully setting up **requirements engineering / management / development (incl. interfaces with verification & validation, change and configuration management, risk, (functional) safety assurance, ...)** and, in particular, **IBM Rational DOORS** (ex QSS DOORS and then Telelogic DOORS) / **IBM Rational DOORS Next Generation (DNG)**, **Siemens Polarion**, **Visure Requirements ALM**, … for a good number of companies worldwide; training and coaching thousands of engineers for hundreds of up-to multi-billion USD/EUR/CHF projects.

Goals of the Presentation

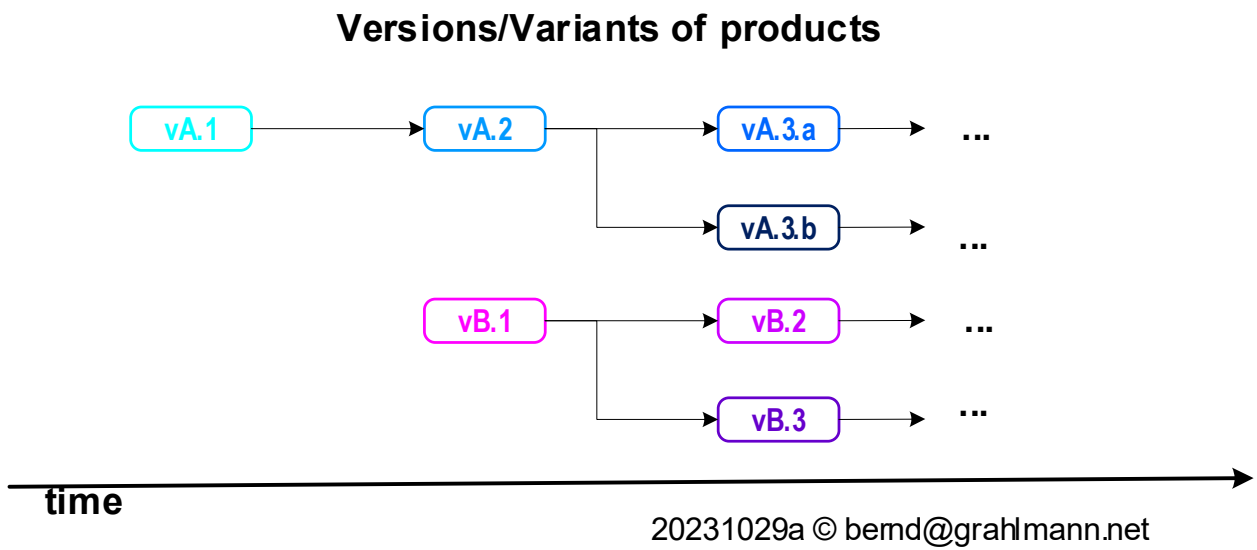
- ✓ Clarify difference between versions / variants and platforms of (systems of) systems, sub-systems, components, … versus versions of specifications, requirements, V&V cases, features, …
- ✓ Gather requirements and goals wrt. efficient and effective versions/variants management and re-use;
- ✓ Go into some detail of version/variant specific attributes;
- ✓ Look at version/variant specific traceability;
- ✓ Give some tool specific ideas;

- ✓ Help to get you going in the right direction;
- ✓ Show ways to reduce the tooling complexity;

Versions/variants & platforms of products, systems, components, … (I)

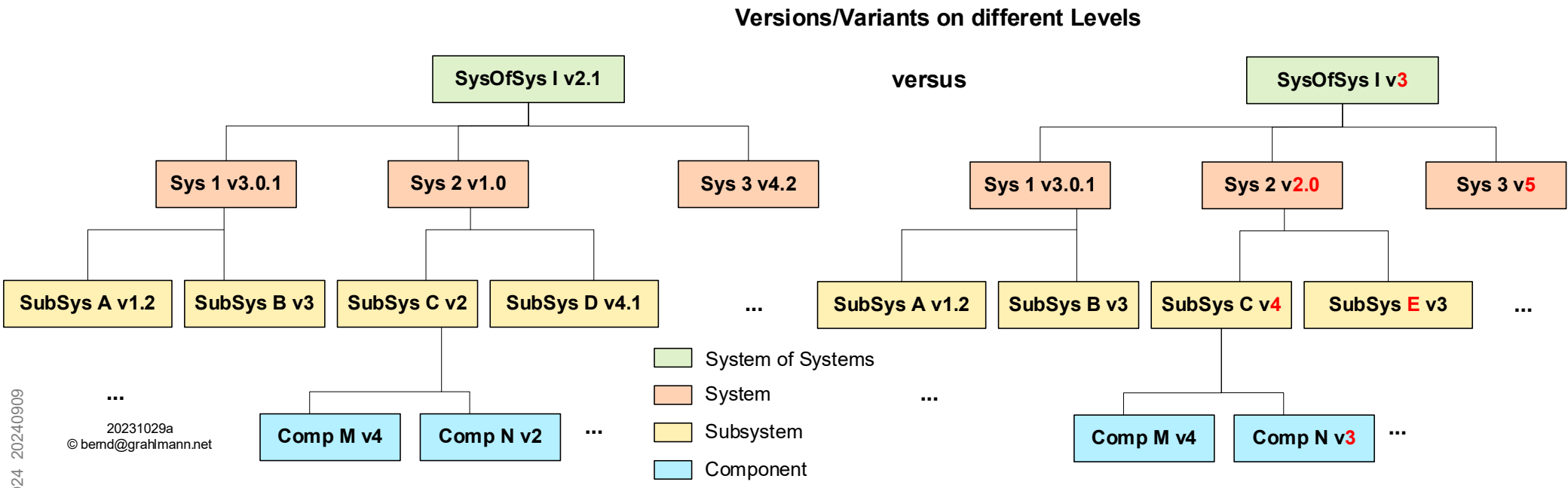
- Companies try to develop and market products and/or services (e.g., medical device, train, locomotive, aircraft, satellite, elevator, vehicle, power plant, ship, submarine, nuclear fusion reactor, ground control, flight inspection system, airborne surveillance system, defense system, cellular product package, GNSS receiver IC, TV, broadcast reception device customer support package, car entertainment system, smart card controller platform, eID, connected truck system, automotive supply, 3D video game, slot-machine, banking software, laboratory management system, fire protection system, Train Control System, …);

- Typically companies try to develop and market different versions/variants of their products:
  a) the product may evolve over time; and/or
  b) different flavors of a product are wanted;
  c) even different products may be considered versions/variants of one more general product;
  d) Organization into platforms / families … often helps.



Versions/variants & platforms of products, systems, components, … (II)

- These typically imply decomposition into (partially) different versions/variants of sub-systems, components, sub-components, …
- Potentially even different sub-systems, components, sub-components, …
- For sake of reducing complexity it may be pragmatic to use only the set of product versions/variants on all levels (and to map in the background to the 'real' versions of the lower levels – e.g. via DOORS DXL attributes).



Versions of Specifications, Requirements, V&V Cases, Features, …

- Companies have various specifications (e.g., Customer Requirements Specifications, Standard and Regulations, Feature Lists, List of (Goals of) Use Cases, Technical System Requirements Specifications, Technical Sub-System Requirements Specifications, V&V Specifications, Architecture, …);
- Those contain various objects/items/artefacts (e.g., Requirements, Features, (Goals of) Use Cases, V&V Cases, V&V Steps, Architecture Elements, Information, …) having their text/description as THE relevant property;
- Specifications as well as objects/items/artefacts have additional properties managed via attributes/fields (e.g., Specification Name, Specification Type, Responsible, Object Type, Applicability, Risk Level, Satisfaction Level, Priority, V&V Result, …);
- Managing some of those properties respectively attributes/fields in a version/variant specific way (e.g., Applicability per Version/Variant, Risk Level per Version/Variant, Satisfaction Level per Version/Variant, Priority per Version/Variant, V&V Result per Version/Variant, …) avoids duplication of specifications as well as of objects/items/artefacts – later more on that;
- Specifications and their objects/items/artefacts evolve over time – i.e., their properties respectively attributes/fields evolve over time – resulting in versions of specifications, requirements, V&V cases, features, …
- Most tools manage these versions of specifications, requirements, V&V cases, features, … internally;

Challenges

No matter what (systems of) systems you develop, most often your challenge to **do the systems engineering in an efficient, effective, … way** becomes extremely **more complex, complicated, …** because you actually want to **develop multiple versions/variants** (of your *Scopes* – i.e. systems, sub-systems, components, …).

In addition to the usual challenge to properly manage changes and configurations of all your specifications (with all their requirements, architecture elements, V&V cases, V&V steps, …) one or even more dimensions are added if versions/variants of your systems of systems, systems, sub-systems, components, … require proper distinct management.

Different versions/variants imply that you **need to manage** (at least) certain **properties / characteristics / … of all your requirements, architecture elements, V&V cases, V&V steps, … per version/variant**.

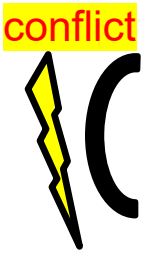
Starting with Requirements and Goals (I)

Taking requirements and goals on what a systems engineering tool including setup shall provide … as an example and abbreviating this combination with '*SE Environment*' and abbreviating product, system of systems, system, sub-system, component, … as '*Scope*':

The *SE Environment* shall allow to:

1. specify requirements, features, V&V cases, … for different versions/variants of a *Scope*;
2. specify version/variant **independent** properties of requirements, features, V&V cases, …;
3. specify version/variant **specific** properties of requirements, features, V&V cases, …;
4. get attributes which automatically summarize version/variant specific properties (like Applicability or V&V Results) for a set of versions/variants of your choice;
5. filter specifications on requirements, features, V&V cases, … by their applicability to versions/variants of a *Scope;*
   Note, that this includes comparison …
6. show traceability filtered on what is applicable to a version/variant of a *Scope;*

Starting with Requirements and Goals (II)



Goals - The *SE Environment* should:

1. Minimize the number of objects/items for requirements, features, V&V cases, architecture elements, … which are required to manage multiple versions/variants of a *Scope*;
2. Minimize the number of attributes/fields of requirements, features, V&V cases, architecture elements, … which are required to manage multiple versions/variants of a *Scope*;
3. Minimize **human errors** when managing requirements, features, V&V cases, architecture elements, … for multiple versions/variants of a *Scope*;
4. Minimize the effort when managing requirements, features, V&V cases, architecture elements, … for multiple versions/variants of a *Scope*;
5. Minimize the setup effort for re-use and multiple versions/variants of a *Scope*;
6. Optimize the speed when managing requirements, features, V&V cases, architecture elements, … for multiple versions/variants of a *Scope*;

Version/Variant specific Attributes [I]

If you want to avoid / reduce duplication per version/variant (of those) of your requirements, architecture elements, V&V cases, V&V steps, … (which share the same text/description), you **need to manage** (at least) certain **properties / characteristics / … of (all) your requirements, architecture elements, V&V cases, V&V steps, … per version/variant**.

Typical examples of such version/variant specific properties / attributes are:

1. Applicability
2. Customer Priority
3. Risk Level
4. Qualification Level
5. V&V Measures
6. Realization Level

Req Text	v1 Applicability	v2 Applicability	v1 Prio	v2 Prio	v1 Risk	v2 Risk
Req1	Approved	Draft	High	Medium	High	High
Req2	Ready for Review	Not Applicable	Medium	Low	Medium	High

(On a first look – for like 10 versions/variants) There is a trade off:

1. multiplying the requirements, architecture elements, V&V cases, V&V steps, … by almost 10 
2. multiplying the attributes per requirement, architecture element, V&V case, V&V step, … by 3-5 

Version/Variant specific Traceability

If you avoided / reduced duplication per version/variant (of those) of your requirements, architecture elements, V&V cases, V&V steps, … (which share the same text/description), you can **avoid / reduce duplication of their traceability BIG times** ☺ !

✓No need to have different traceability between, e.g., one system requirement and one system level V&V / test case ☺

✓A system requirement (i.e. sharing the same text/description for different versions/variants) typically does NOT require:

- ✓ different system level V&V cases ☺,
- ✓ different sub-system requirements, … ☺

✓Efforts for satisfaction arguments, qualification arguments, … go down ☺

✓Review efforts go down ☺

So, after considering traceability, on a second look the trade off looks much more

1. **in favor of multiplying the attributes** per requirement, architecture element, V&V case, V&V step, …
2. versus multiplying the requirements, architecture elements, V&V cases, V&V steps, …

Tool Factor & Human Factor

Of course, tooling is a big factor:

- How well does your tooling (with setup, tool extensions, …) 'support' multiplication of objects versus multiplication of attributes – wrt. setup, ease of use, … but also wrt. performance, … ???
  E.g., IBM Rational DOORS with iDARM tools and appropriate templates, … automate and ease the multiplication of attributes ☺ – while this still requires quite some manual work in other tools ☹
  Multiplication of objects and subsequent 'explosion' of traceability often result in performance issues ☹

➔ **Tool extensions, good templates, correct setup**, … are (still) key to success !!!

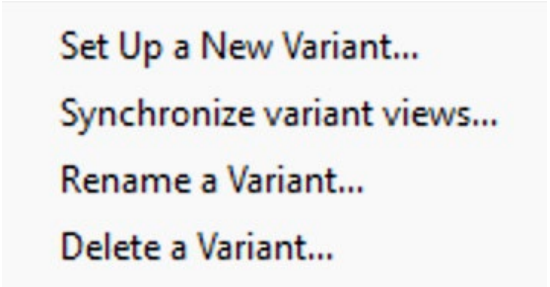
Underestimating or even neglecting the 'Human Factor' (too) often brought companies in trouble:

1. **Users got confused** having to handle versions of requirements, architecture elements, V&V / test cases, risks, … evolving over time and in parallel (versions of) requirements, architecture elements, V&V / test cases, risks, … mapping to versions/variants of products, systems, sub-systems, … - in particular, when traceability was involved !!!
2. Users had difficulties with (too) complex parameterization approaches ☹

➔ **KISS** (Keep it Simple, Stupid) combined with **training, coaching**, … are (still) key to success !!!

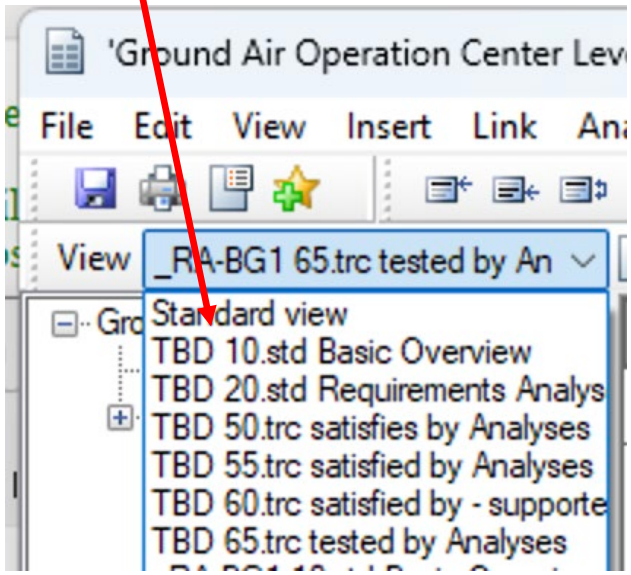
IBM Rational DOORS with iDARM Tools and Templates (I)

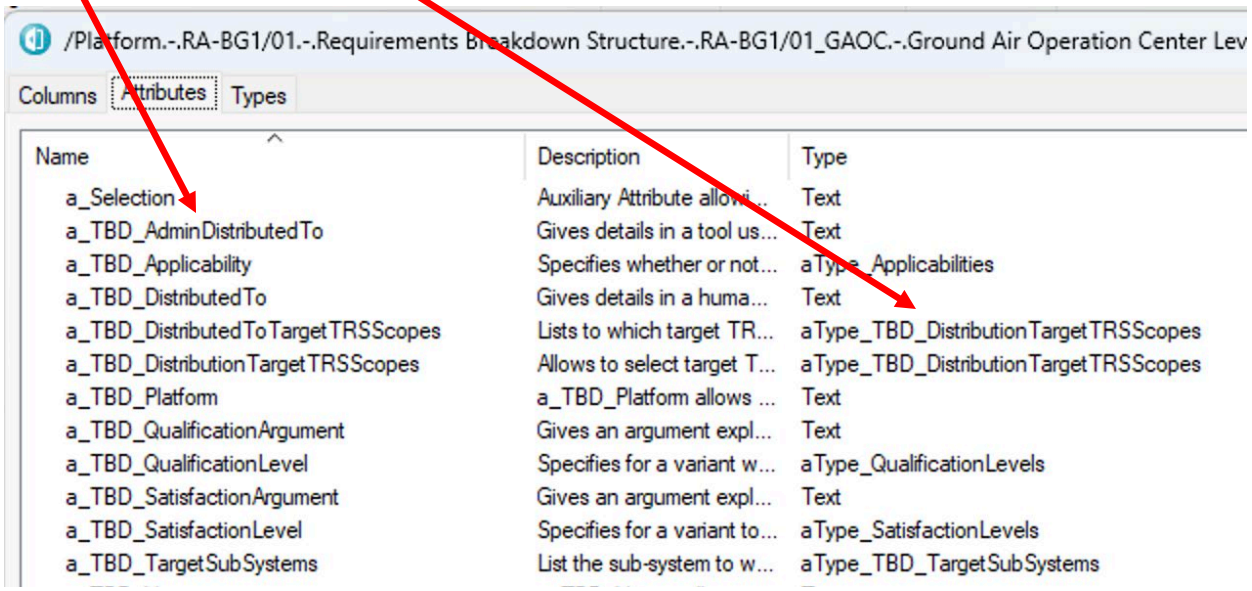
For IBM Rational DOORS, the
iDARM Tools offer (among others):



With the appropriate templates containing for a template ‘dummy’ version/variant ‘TBD’:

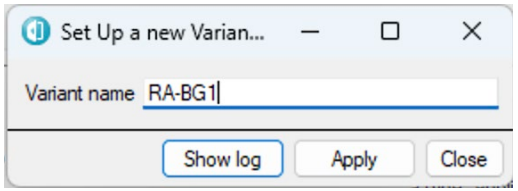
1. (potentially DXL = scripted) Attributes and Types;
2. Views (potentially with DXL layout columns = scripted).



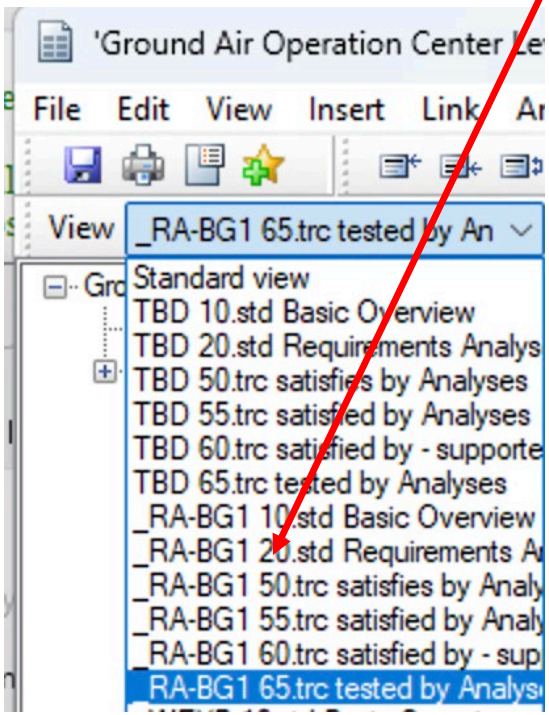


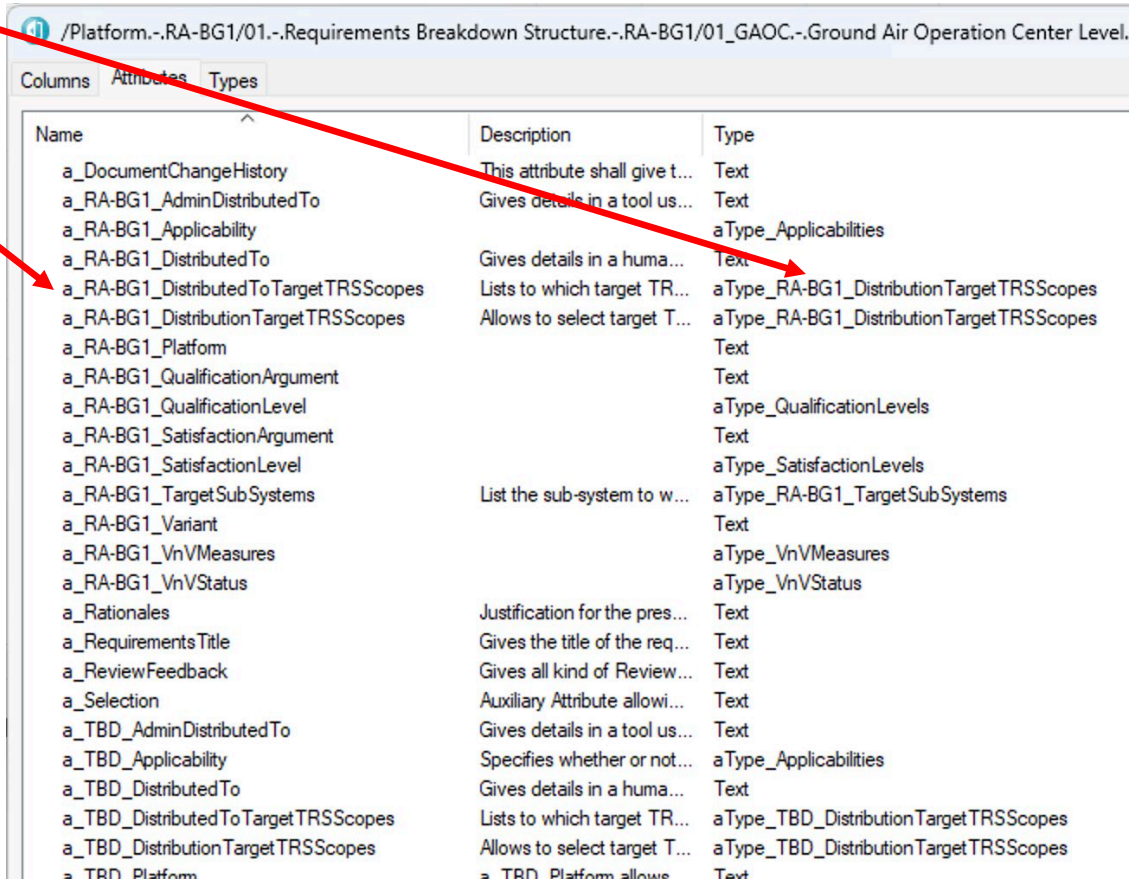
IBM Rational DOORS with iDARM Tools and Templates (II)

Via the iDARM Tools the setup for variants/versions can easily be automated:



Instantiating attributes, types, and views from the template 'dummy' version/variant 'TBD'

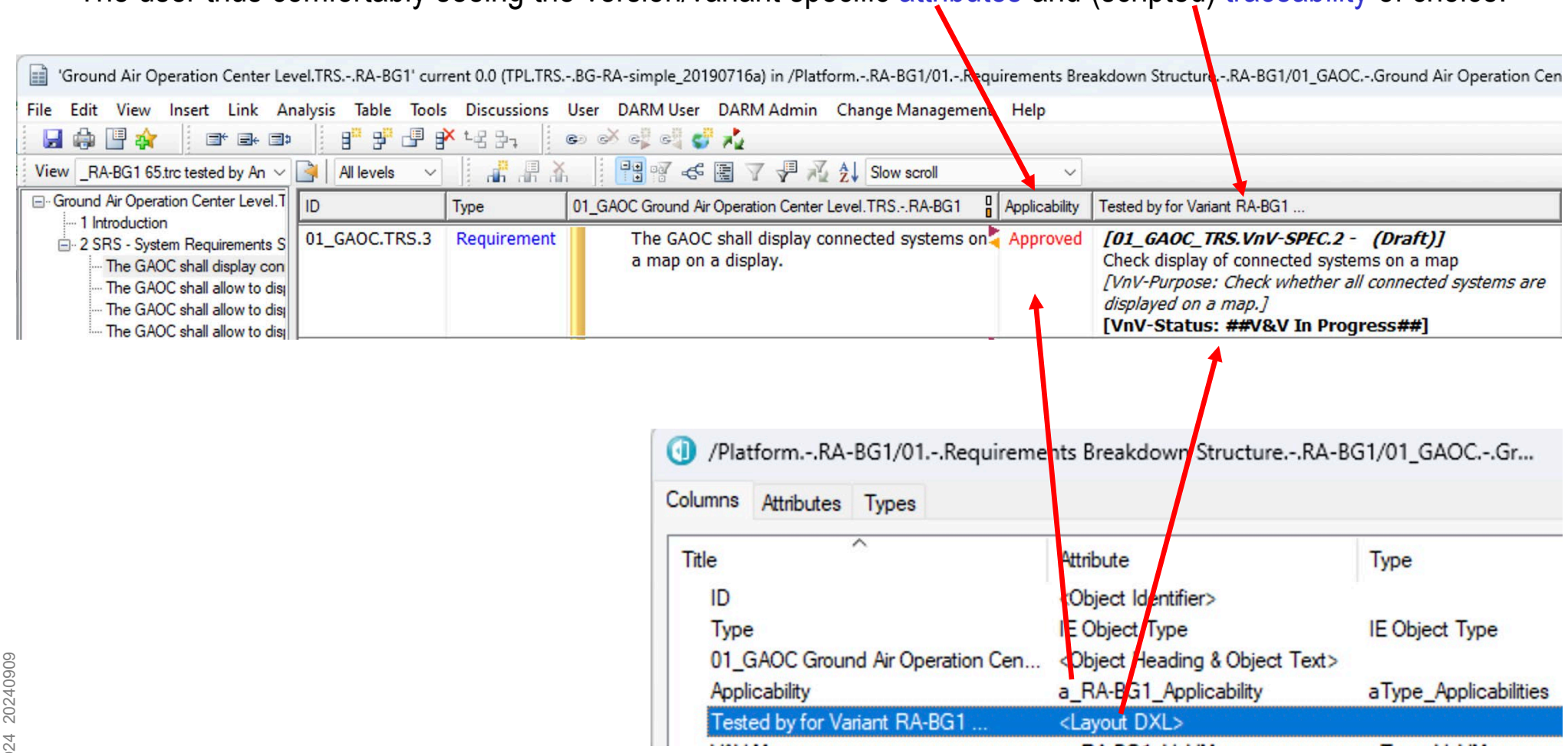




Name	Description	Type
a_DocumentChangeHistory	This attribute shall give t...	Text
a_RA-BG1_AdminDistributedTo	Gives details in a tool us...	Text
a_RA-BG1_Applicability		aType_Applicabilities
a_RA-BG1_DistributedTo	Gives details in a huma...	Text
a_RA-BG1_DistributedToTargetTRSScopes	Lists to which target TR...	aType_RA-BG1_DistributionTargetTRSScopes
a_RA-BG1_DistributionTargetTRSScopes	Allows to select target T...	aType_RA-BG1_DistributionTargetTRSScopes
a_RA-BG1_Platform		Text
a_RA-BG1_QualificationArgument		Text
a_RA-BG1_QualificationLevel		aType_QualificationLevels
a_RA-BG1_SatisfactionArgument		Text
a_RA-BG1_SatisfactionLevel		aType_SatisfactionLevels
a_RA-BG1_TargetSubSystems	List the sub-system to w...	aType_RA-BG1_TargetSubSystems
a_RA-BG1_Variant		Text
a_RA-BG1_VnVMeasures		aType_VnVMeasures
a_RA-BG1_VnVStatus		aType_VnVStatus
a_Rationales	Justification for the pres...	Text
a_RequirementsTitle	Gives the title of the req...	Text
a_ReviewFeedback	Gives all kind of Review...	Text
a_Selection	Auxiliary Attribute allowi...	Text
a_TBD_AdminDistributedTo	Gives details in a tool us...	Text
a_TBD_Applicability	Specifies whether or not...	aType_Applicabilities
a_TBD_DistributedTo	Gives details in a huma...	Text
a_TBD_DistributedToTargetTRSScopes	Lists to which target TR...	aType_TBD_DistributionTargetTRSScopes
a_TBD_DistributionTargetTRSScopes	Allows to select target T...	aType_TBD_DistributionTargetTRSScopes

IBM Rational DOORS with iDARM Tools and Templates (III)

The user thus comfortably seeing the version/variant specific attributes and (scripted) traceability of choice:



Questions & Answers

Contact me via email: **Bernd@Grahlmann.net** or phone **+41 792967651**

or check via https://www.grahlmann.net/doors_requirements_management_training_overview.htm
or LinkedIn: https://www.linkedin.com/in/grahlmanndoorstelelogic/
or Xing: https://www.xing.com/profile/Bernd_Grahlmann/

or join ‚my' LinkedIn groups:

- ‚Requirements Engineering Tools' https://www.linkedin.com/groups/12821233/
- ‚IBM Rational DOORS and DOORS Next Generation - DNG (ex Telelogic DOORS) User Group' https://www.linkedin.com/groups/769057/
- ‚Siemens Polarion' https://www.linkedin.com/groups/12004818/

Thanks a lot ☺