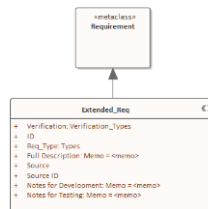


# Extending the Requirement Type

An MDG Technology for adding attributes to manage requirements for systems and software engineering projects



## Who Am I



- Systems Engineer
- Modeling Consultant
- Sparx representative to the OMG
- Member of the Requirements Working Group at INCOSE
  - A special thank you to Mark Harris and Lou Wheatcraft of the RWG for their reviews of this presentation

## Abstract



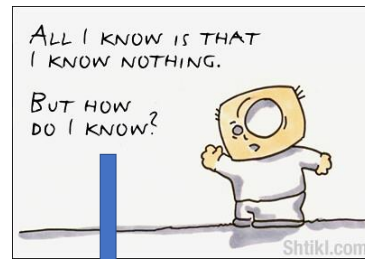
- Because UML® does not define a requirement element type, Enterprise Architect includes one in the Core Extensions set of elements. SysML® includes a requirement element type that expands the EA type with two attributes. In the Guide for Writing Requirements, the INCOSE Requirements Working Group has defined a further expansion of the numbers and types of attributes that need to be considered in requirements engineering. In the software arena, Karl Wiegiers has written extensively about software requirements and what you need to know about them. This presentation describes an MDG Technology that addresses the INCOSE recommendations and maps them to Wiegiers' recommendations for attributes to capture about your requirements. We will explore both the creation and the use of the extended requirement elements.

# The UML Requirement Element

## UML 1.x




## UML 2.x



EA told me so

Much has been written in the last several years about ways to capture requirements. Things like user stories, use cases, and gold customers have all been discussed. There are many ways in which models can be used to express requirements, such as state machines and business process models. Warren Zevon wrote a song about a 3-pronged approach to problem solving, "Send lawyers, guns, and money." If you work in a situation where informal requirements are the norm, then you probably don't have to worry about lawyers, contracts, and other aspects of the legal system. However, there are those of us who do have to be concerned with lawyers and such. We tend to rely on well-constructed requirement expressions and we need to incorporate those into our models. UML didn't and doesn't offer any support in this area so we have to see what EA has to offer.


# The EA Requirement Element



**Properties**

Element	Tags
Name	MTR < 1 hour
<b>General</b>	
Type	Requirement
Stereotype	
Alias	QOS-01
Keywords	
Status	Proposed
Version	1.0
<b>Requirement</b>	
Abstract	<input type="checkbox"/>
Active	<input type="checkbox"/>
Difficulty	Medium
Final Specialization	<input type="checkbox"/>
Leaf	<input type="checkbox"/>
Priority	Medium
Visibility	Public
<b>Project</b>	
Author	J.D. Baker
Package	Quality of Service
Phase	1.0
Complexity	Easy
Created	5/9/2020 9:55:51 AM
Modified	5/9/2020 9:55:51 AM
Language	<none>
Filename	
GUID	{297BCB47-4F5D-4944-80CC-4DF6...}
WebEA	
Alias Optional alias value	

**Core Extensions**



Location: Internal Technology

**Description**

Requirements, maintenance, analysis, user interface, data modeling, documentation, business modeling and test domain diagrams.

**Web Site**

<http://www.spansystems.com>

**Notes**

In the event of a loss of internet connectivity, the mean time to restore connectivity shall be less than one hour from the initial report.

Quality Attributes

Design Constraints

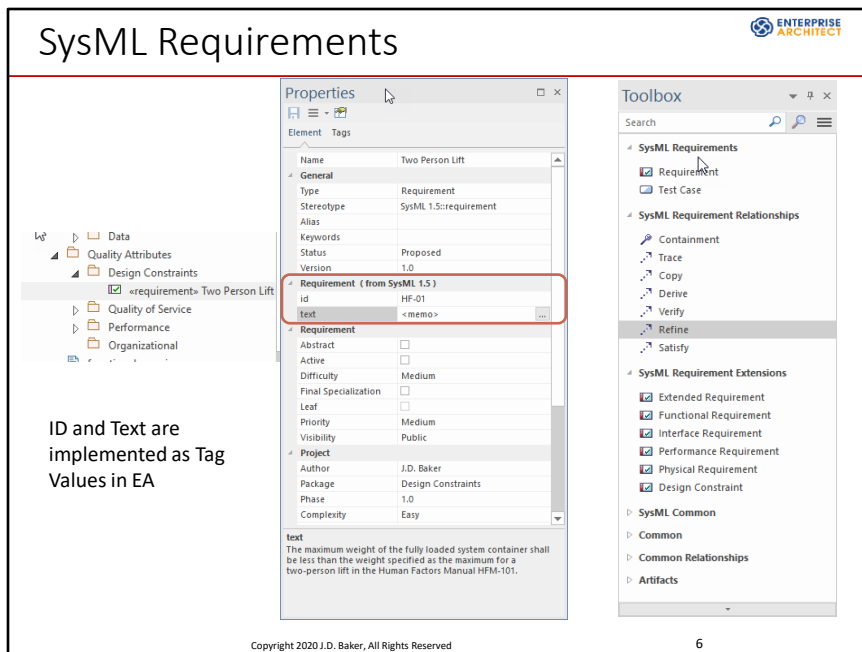
Quality of Service

Performance

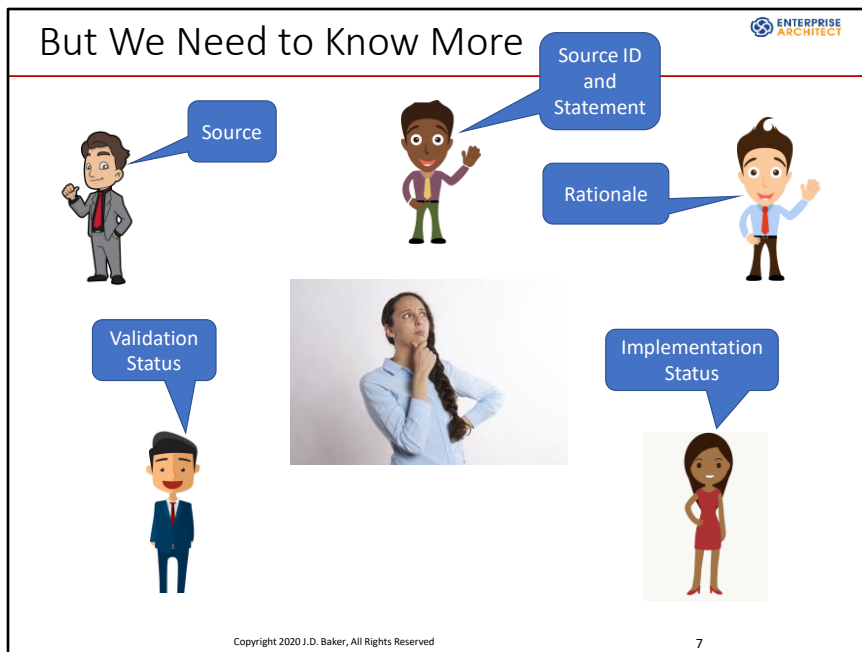
Organizational

Copyright 2020 J.D. Baker, All Rights Reserved 5

Sparx has long recognized the need for a requirement element type. The Core Extensions are a number of elements and their supporting diagrams that are a part of every EA distribution and may optionally be included in every EA model. Every model element in EA has an associated set of properties that includes Alias and Notes. A common way to use the Requirement element is to capture a requirement ID in the Alias property and the requirement text in the notes property.



SysML requirements in EA are based on the Core Extension requirement element. SysML specifies ID and text properties for the requirement element. These are implemented in EA as tag values. The text tag is where the requirement statement is captured, thus freeing the Notes property for other purposes. The ID tag frees the Alias property for other uses as well. The SysML specification also includes a set of non-normative extensions of the requirement stereotype. The «extendedRequirement» stereotype adds source: String risk: RiskKind. and verifyMethod: VerifyMethodKind properties. RiskKind is actually risk level (High, Medium, and Low) while VerifyMethodKind is one of (Test, Demonstration, Analysis, or Inspection). The other stereotypes do not include additional properties but have specified uses and constraints. For example a «functionalRequirement» is constrained to be “satisfied by an operation or behavior”.



Is there anything else we need to know about our requirements?

I had one customer tell me they needed to know the source of all requirements, not just one type

. I had another tell me they needed two pieces of information regarding the source.

A common question is what is the rationale for a requirement?

Others might need to have information about validation or maybe some information about implementation.

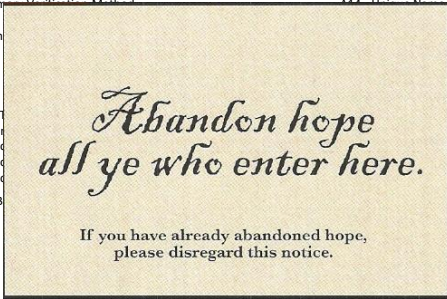
The good folks in the INCOSE Requirements Working Group have put together the Guide for Writing Requirements (we'll call it the GfWR after this) and given us their thoughts on what might be important to know about a requirement.

ATTRIBUTES TO HELP DEFINE THE REQUIREMENT AND ITS INTENT

- A1 - Rationale
- A2 - System of Interest (SOI) Primary Functions, Methods
- A3 - SOI Verification Approach
- A4 - Trace to Parent Requirement
- A5 - Trace to Source
- A6 - Condition of Use
- A7 - States and Modes
- A8 - Allocation
- ATTRIBUTES ASSOCIATED WITH THE REQUIREMENT
- A9 - SOI Verification or Validation
- A10 - SOI Verification or Validation
- A11 - SOI Verification or Validation
- A12 - SOI Verification or Validation
- ATTRIBUTES TO SHOW APPLICABILITY
- A39 - Applicability
- A40 - Region
- A41 - Country
- A42 - State/Province
- A43 - Application
- A44 - Market Segment
- A45 - Business Unit
- A46 - Business (Product) Line

ATTRIBUTES TO HELP MAINTAIN THE REQUIREMENTS

- A13 - Unique Identifier
- A33 - Criticality or Essentiality
- A34 - Risk (of implementation)
- A35 - Risk (Mitigation)
- A36 - Key Driving Requirement (KDR)
- A37 - Additional Comments
- A38 - Type/Category



THERE'S MORE!! *"This list is not exhaustive"*

Section 5 of The Guide for Writing Requirements defines 46 different requirement attributes in 4 different groups. The current version of the GfWR *Notes that Many of the attributes listed are useful for both managing needs as well as requirements. Others may be more useful as applied to only requirements.*

CLICK

And then they hit you with the notion that there may be more to consider.

CLICK – What’s a poor requirements engineer to do? Where do we go to capture all of this stuff??!???



## What Else Does GfWR Say?



*"It is not the intention that an organization should include all of these attributes when defining needs or requirement expressions."*

*"As with the use of all information, a "lean" approach should be taken when deciding which attributes will be used - don't include a specific attribute unless you, your team, or your management has asked for that attribute and will be using that attribute in some manner to manage the project and set of requirements."*

Ok – turns out the Guide authors knew what they were doing. They bracketed the list with statements that said all 46 attributes are not required in all situations. The notion that you should only include what is useful is an important concept not only in defining requirement attributes but in all things in modeling systems. Before you commit resources to creating and managing an attribute or element, you should know what it's going to be used for and who is going to use it.

## First Group

- ATTRIBUTES TO HELP DEFINE THE REQUIREMENT AND ITS INTENT
  - A1 – **Rationale**
    - A separate note in SysML
  - A2 - **System of Interest (SOI) Primary Verification Method**
  - A3 - **SOI Verification Approach**
  - A4 - Trace to Parent Requirements
    - An attribute implemented as a relationship in EA
  - A5 - Trace to Source
    - An attribute that could be implemented as a relationship in EA
  - A6 - Condition of Use
    - Still trying to figure out how this is an attribute
  - A7 - States and Modes
  - A8 - Allocation

Ok – so let’s take a look at the list again and this time see which ones are our favorites. I personally believe that every requirement should have a stated rationale. Verification is also important to consider as the requirement is being developed. Tracing to a source is an interesting approach. If this means literally modeling a connector between a requirement and an element representing a source document then it seems to me to be very difficult to identify any specific location in that source.

## Second Group

- ATTRIBUTES ASSOCIATED WITH THE SOI AND ITS VERIFICATION
  - A9 - SOI Verification Level
  - A10 - SOI Verification Phase
  - A11 - SOI Verification Results
  - A12 - SOI Verification Status



**Baker sez – this information is better captured as a verification case, not isolated with the requirement element.**

The first two of this group are part of the requirement development and planning. There may be a way to incorporate them into the requirement itself. The second two are about an event that happens later in the development cycle. They are attributes of the verification case, not the requirement. I like the notion of relating a requirement to a verification case where all 4 can be managed together.

## Third Group



- ATTRIBUTES TO HELP MAINTAIN THE REQUIREMENTS
- **A13 - Unique Identifier**
- **A14 - Unique Name**
- **A15 - Originator/Author**
- **A16 - Date Requirement Entered**
- A17 – Owner
  - *As long as it's not a person's name*
- A18 – Stakeholders
  - *RACI Matrix*
- A19 - Change Board
- A20 - Change Status
  - *Use a separate Change element*
- A21 - Version Number
- A22 - Approval Date
- A23 - Date of Last Change
- A24 - Stability
- A25 - Responsible Person
  - *No person's names*
- A26 – Need or Requirement Verification Status
- A27 – Need or Requirement Validation Status
- A28 - Status (of the need or requirement)
- A29 - Status (of implementation)
- A30 - Trace to Interface Definition
- A31 - Trace to Peer Requirements
- **A32 - Priority**
- **A33 – Criticality or Essentiality**
- **A34 – Risk (of implementation)**
  - *Not embedded in the requirement*
- A35 – Risk (Mitigation)
- A34 - Key Driving Need or Requirement (KDN/KDR)
- **A35 - Additional Comments**
  - *Notes in EA*
- **A36 - Type/Category**

Copyright 2020 J.D. Baker, All Rights Reserved

12

The third group of attributes is rich with things to consider. A user controlled unique identifier, as opposed to a tool controlled identifier, is important in ensuring that requirements can be disambiguated. Many of the highlighted attributes in this list can be captured as built-in properties of an EA element and no customization is required, such as name, author, Date created, Comments, and Priority. Others are best captured in different elements or visualizations, such as a RACI matrix for stakeholders (to include owners and Responsible Person without a name). Beyond the author, names belong in organization charts and the role names from those charts are what is associated with the requirement. If the author leaves the project or the organization having their name associated with the requirement might still be useful. There is no advantage to knowing the name of someone who used to be the Responsible Person if you need approval for a change in the requirement. I continue to believe that verification status, validation status, and implementation status are not best managed as a requirement attribute.

## Fourth (and Last) Group



- ATTRIBUTES TO SHOW APPLICABILITY AND ALLOW REUSE
  - A39 - Applicability
  - A40 - Region
  - A41 - Country
  - A42 - State/Province
  - A43 - Application
  - A44 - Market Segment
  - A45 - Business Unit
  - A46 - Business (Product) Line

This set of requirement attributes have limited usefulness. If you are dealing with requirements in a multi-national setting or in a product line then some or all of these may be of interest.

## Wiegiers Suggestions

- Date the requirement was created (A16)
- Current version number of the requirement (A21)
- Author who wrote the requirement (A15)
- Priority (A32)
- Status (A28)
- Origin or source of the requirement (A5)
- Rationale behind the requirement (A1)
- Release number or iteration to which the requirement is allocated (A29?)
- Stakeholder to contact with questions or to make decisions about proposed changes (A17 and 18)
- Validation method to be used or acceptance criteria (A3)
- “Selecting too many requirements attributes can overwhelm a team. They won’t supply all attribute values for all requirements and won’t use the attribute information effectively. Start with perhaps three or four key attributes. Add others only when you know how they will add value.”
- Chapter 27, Software Requirements 3rd ed. by Karl E Wiegiers and Joy Beatty Published by Microsoft Press, 2013

Karl Wiegiers had a much shorter list of attributes for software requirements. It should not be surprising that all of them map to one of the GfWR attributes. I did find it curious that he and Joy Beatty list validation method, but not verification method. I checked the book and in chapter 18 they very clearly state that they understand the difference between verification and validation and they truly mean validation here. They also say that their approach conflates the two. Let me just throw this out there – I believe all requirements need to be verified but only customer facing requirements need to be validated.

They echo the GfWR statements about only including the attributes that add value and have a purpose.

## Our Requirement Metadata

- Date the requirement was created (A16)
- Current version number of the requirement (A21)
- Author who wrote the requirement (A15)
- Priority (A32)
- Status (A28)
- Origin or source of the requirement (A5)
- Rationale behind the requirement (A1)
- Release number or iteration to which the requirement is allocated (A29?)
- Stakeholder to contact with questions or to make decisions about proposed changes (A17 and 18)
- Validation method to be used or acceptance criteria (A2)
- Created property
- Version property
- Author property
- Priority property
- Status property
- extension
- extension
- Phase property
- extension
- extension



Where do we capture the attributes in EA?

With those two sources as our foundation, let's select the attributes for our requirement metadata. Since all of the items on the Wiegers' list are included in the INCOSE list, we will start with that. A number of these attributes can be captured as built-in properties of all EA elements.

Not Sufficient!



I don't think the Wieggers/Beatty list is sufficient. There is more that needs to be added.



## Our Requirement Metadata



- Requirement ID
  - Requirement Name/Short Text
  - Requirement Statement
  - Verification
    - Analysis
    - Inspection
    - Demonstration
    - Test
  - Type (e.g. FURPS+)
    - Functional
    - Usability
    - Reliability
    - Performance
    - Supportability
- Alias property is often used
  - Name property
  - Extension
  - Extension
  - Extension

Once we add the items here to our list, these 15 are enough for us to get started on the UML profile

A word about the Types. I have had customers who asked me for help simplifying the requirements choices. It turns out that the SysML approach of creating separate stereotypes for each requirement type both limits the choices an organization can make and sometimes causes engineers to select the wrong stereotype. Changing those requirements is challenging, especially when you are changing from a type that has additional attributes to a type that does not. Our approach to Type information will solve this problem.

## 7 Attributes Noted as “Extension”



- Rationale
  - A stereotyped comment in SysML
    - Not sufficient
- Source
- Stakeholders (RACI)
- Validation Method
- Verification Method
  - A Tag Value
  - A Relationship matrix
  - ?
  - Tag Value
    - The profile also includes the need for a verification requirement
- Requirement Statement
  - Tag Value
- Requirement Type
  - Tag Value

Rationale is a stereotyped comment in SysML. A comment in UML is not a NamedElement and as a consequence does not appear in the Project Browser. It only appears on a diagram. It is very common to have a set of requirements in a model where all or even most of the requirements do not appear on a diagram and therefore could not have the rationale documented for them. I believe that every requirement should have an associated rationale and therefore the UML profile will include a tag value for that.

Multiple stakeholders can have a relationship to a requirements. A common way to describe that is a RACI matrix. Capturing that information should be external to the requirement attributes. Responsible: Accountable: Consulted: Informed: Every element in EA has a Note item. We have reserved that for ad hoc notes about our requirement

## Requirement traceability

- The Requirements Traceability Matrix (RTM) is used to control & track system level, allocated and derived requirements.
- A Requirements Analysis Checklist and the Requirements Management Planning Template may describe the RTM and the RVTM.
- The RTM and RVTM are not distinct files. They are created as needed by EA
  
- Why use requirements traceability?
  - Ensure that the system does what it is supposed to do
  - Ensure that the system does only what it is supposed to do
  - Assess impact of change
    - Find related requirements
    - Inspect related requirements


Requirements Traceability is very important as the program moves into the design phase since lower level derived requirements are developed to support the physical design.

These lower level requirements need to be linked to their parent requirement to establish and maintain traceability.

At some point, a parent requirement can be changed or deleted. This will affect all linked lower level requirements.

To re-iterate – why is traceability so important because it makes sure the system does what it is supposed to do and it makes sure the system “ONLY” does what it is supposed to do.

# Relationship Matrix



---

Start Page | Relationship Matrix

Source: Functional | Type: Requirement | Link Type: Trace | Profile: req\_demo3  
 Target: Subsystems | Type: Component | Direction: Target -> Source | Overlays: <None>

Source	Alert Management	Data Acquisition and Processing	Data Exchange	Data Management	Maintenance	Quality Control
accept atmospheric condition data derived from atm	↕					
accept atmospheric condition data derived from atm	↕					
accept environmental data derived from images.			↕			
accept environmental data derived from images.			↕			
accept surface condition data derived from surface						
accept weather hazard reports containing the hazar	↕					
acquire and disseminate National Weather Service (						
allow access to new surface transportation related						


Colors on the matrix highlight the elements which have no relationships of the designated Link Type and Direction. Other relationships could exist.

An alternate visualization of the requirements and one of their trace relationships

Copyright 2020 J.D. Baker, All Rights Reserved 20

What this matrix shows is a trace to implementation. Any number of matrices can be created enabling all of the trace visualizations – requirement to requirements, requirement to interface, etc. If a relationship is created in the matrix, then it exists in the model. If the two elements appear on a diagram together at a later date, the relationship will automatically be represented.

# Stakeholder Matrix



---

Source: Functional ... Type: Requirement ... Link Type: Trace ... Profile: raci ...  
 Target: Actors ... Type: Actor ... Direction: Target -> Source ... Overlays: RACI ...

Source	Data Collector	Environmental Sensing Station	Fleet Operations	Information Service Provider	Invehicle Sensing Station	Management Operations	NOAA ISOS	NWS Operator	Operations Supervisor	Operator	Weather Service Provider
accept atmospheric condition data derived from atm	R	A	C	I							
accept atmospheric condition data derived from atm	I	R	A	C							
accept environmental data derived from images.	C	I	R	A							
accept environmental data derived from images.	A	C	I	R							
accept surface condition data derived from surface											

Real UML relationships exist behind the matrix overlay.

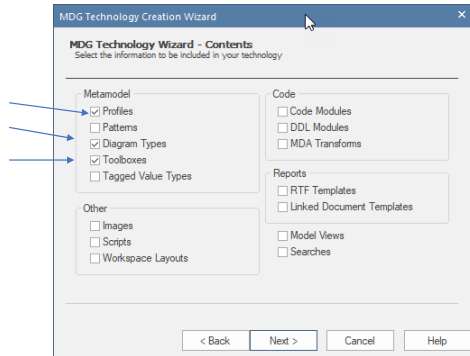
Copyright 2020 J.D. Baker, All Rights Reserved 21

If the stakeholder list/organization chart is maintained in the model, the matrix is easy to develop. If it exists outside the model then a mechanism for importing the information into the model needs to be developed. All of the stakeholder information does not have to be in the model, just enough to uniquely identify each role so it can be accurately mapped to the requirements.

- Risks
  - IN EA these can be captured as elements
  - More flexibility in reporting
  - The same risk can be associated with multiple requirements
- Verification level
  - Adding test cases to the model can provide more than just a level identifier
- Tracking “Deleted” requirements could be a challenge since they need to be present in the model but flagged as deleted

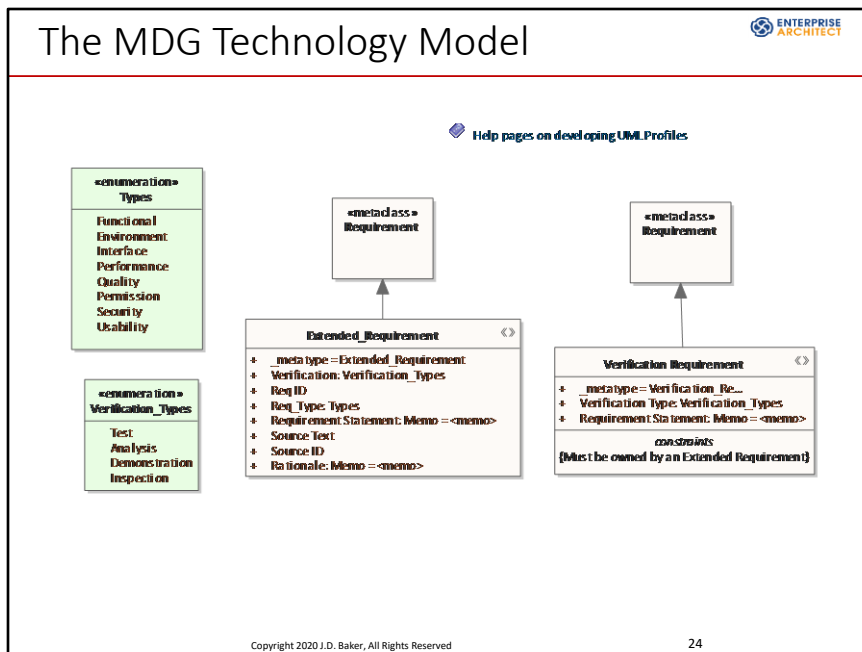
# Model Driven Generation

An EA extension mechanism that is based on models and metamodels



The MDG Technology that we are going to create will have a UML profile that extends the Requirement metatype. That profile will have associated diagrams and toolboxes that enable modelers to select the extended items for creation in the model. These three will be bundled together for use in design models. There are many other options that we can include.

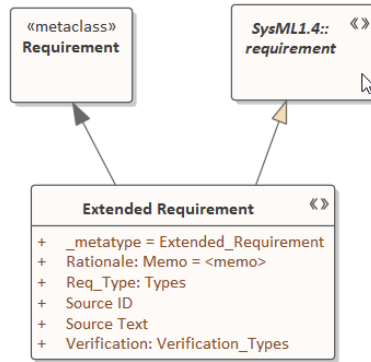
# The MDG Technology Model



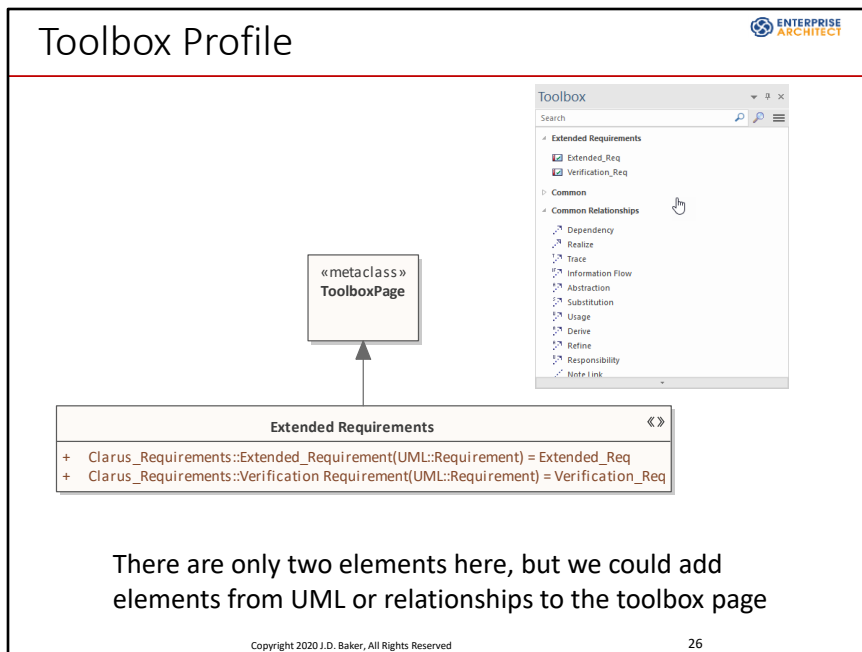
This is the UML profile portion of the MDG Technology. There are two separate extensions of the EA Requirement along with a couple supporting enumerations. Each of the attributes in the two requirement stereotypes will be transformed into tag values when the MDG Technology is created from the profile model. The creation process is essentially model the profile, transform the profile model into an MDG Technology XML, and import the XML into the models where it will be used.



## The SysML Variant

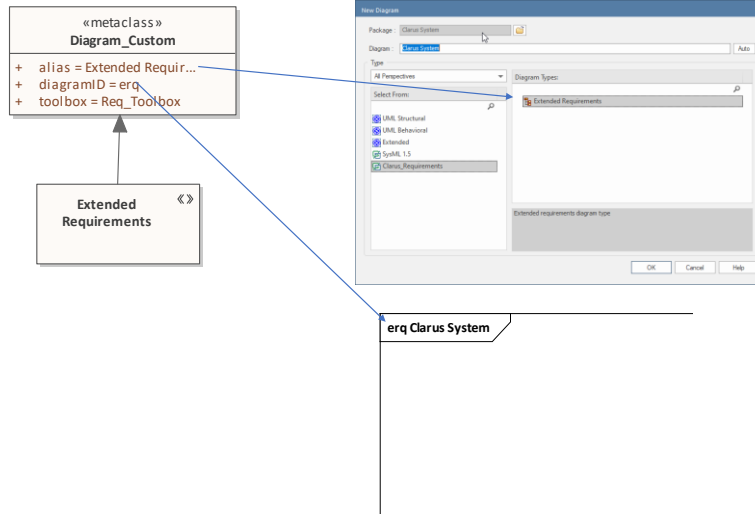


SysML 1.x is itself a UML profile, so the requirement element is already a stereotype. To properly create an Extended Requirement in SysML we need to inherit from the existing stereotype in addition to extending the Requirement meta-element.



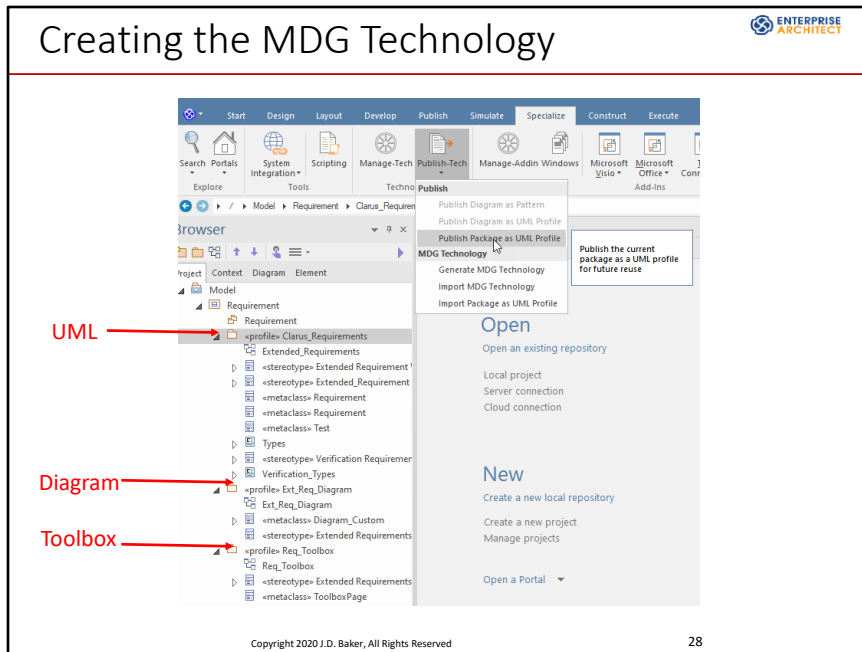
The elements in the toolbox page which will be added to the diagram profile can come from the UML profile we are creating or elements that already exist in the metamodel. Custom relationships or existing relationships are also possible. In his presentation on MDG Technologies yesterday, Phil Chudley gave a very persuasive argument for always creating custom relationships.

# The Diagram Profile



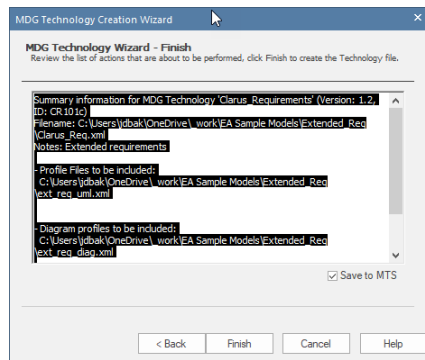
In our example there is only one new diagram type with simple naming and framing information. Each diagram in EA is associated with a selected toolbox so if you are going to create a custom diagram type you should create a custom toolbox page. Note that here we have identified the toolbox page we previously created to be associated with this diagram.

# Creating the MDG Technology



Publish each of the profiles to a separate XML file.

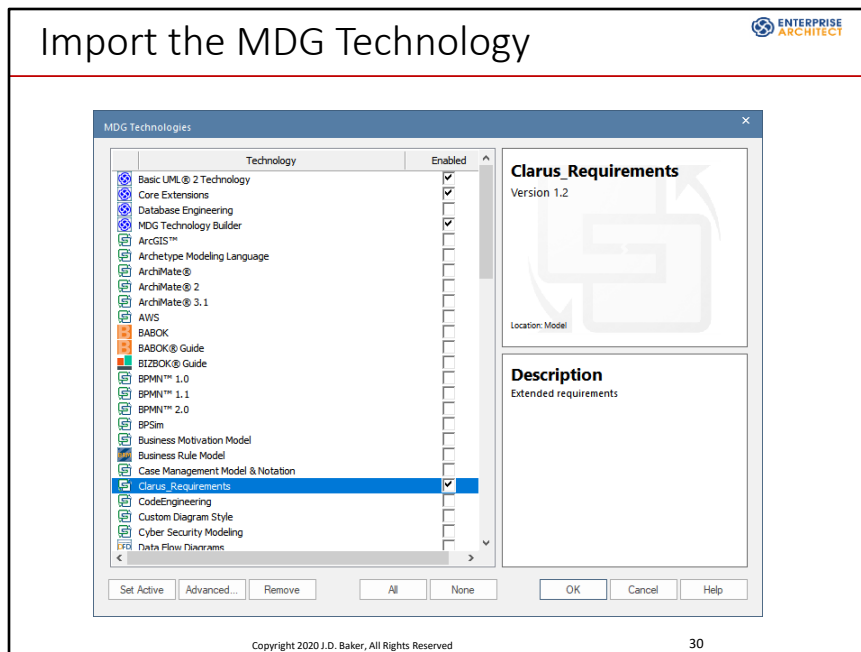
# MDG Technology Creation Wizard



Employ the wizard to generate a combined XML file to be imported wherever it is going to be used. The MDG Technology is named Clarus Requirements because Clarus is the name of the project where the profile was developed.

Not exactly the Wizard of Westwood but it works.

# Import the MDG Technology



EA ships with an ever growing list of MDG Technologies installed. The exact number and type will depend on the version and edition of EA that you have installed. The kinds of technologies include OMG standards, like BPMN and CMMN; standards from other organizations, like the ArchiMate language; and some things that just seem like a good idea, like Data Flow Diagrams.

# An Example

The screenshot displays four windows from the Enterprise Architect software:

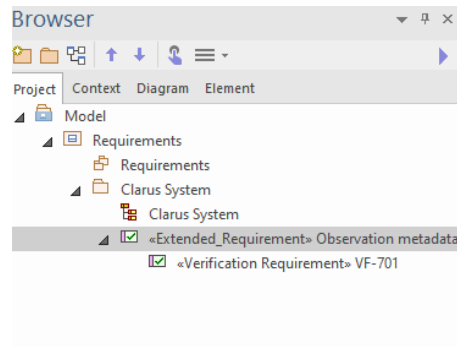
- Properties**: A table showing attributes for an element. The 'Extended\_Requirement' section is expanded, showing 'Req ID = F-701' and 'Req\_Type = Functional'. A blue arrow points from this section to the 'Observation metadata' window.
- Observation metadata**: A window showing tags for the selected element: 'Rationale = <memo>', 'Req ID = F-701', 'Req\_Type = Functional', 'Requirement Statement = <memo>', and 'Verification = Test'.
- VF-701**: A window showing tags for the specific requirement: 'Verification Statement = <memo>' and 'Verification Type = Test'. Below the tags is a note: 'Need to determine the availability of the input simulator and ensure it has the desired capability'.
- Traceability**: A window showing a tree view of relationships. Under 'Observation metadata', there is an 'owns' relationship with 'VF-701'.

Copyright 2020 J.D. Baker, All Rights Reserved

31

This is what it looks like with the MDG Technology in use. The visibility of the tags and notes on a diagram can be controlled per element or per all elements on a single diagram. The Properties window presents the attributes of a selected element. The Traceability window can reveal not only ownership but all of the relations that an element participates in, so one stop shopping for tracing throughout the model.

# Model Organization



The Project Browser reflects the owner relationship through indentation. For the most part I expect there to be a one-to-one relationship between an Extended Requirement and a Verification Requirement.



## Are We There Yet?

- Requirements have exactly the attributes we need
- Verification consideration is an upfront process
- Things are really coming together
- But I don't want to have to do all that clicking around. I miss the way my requirements used to look

There is no need to fear,  
Underdog is here!!



Underdog, Shoeshine Boy's heroic alter ego, appears whenever love interest Sweet Polly Purebred is being victimized by such villains as Simon Bar Sinister or Riff Raff. Underdog nearly always speaks in rhyming couplets,[3] as in "There's no need to fear, Underdog is here!"

# Putting It All Together



Start Page Clarus System Specification Manager

Model Requirements Clarus System Find Package

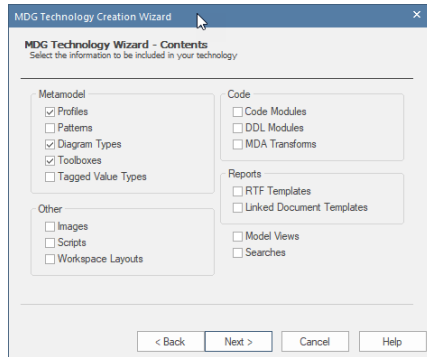
Item	Req ID	Requirement Statement	Rationale	Status	Priority
<input checked="" type="checkbox"/> <b>Observation Metadata</b>	F-701	The Clarus system shall accept only observations that includes the minimum set of metadata. The minimum set of metadata for an observation is location, timestamp, and source information.	Failure to provide the minimum set of metadata means the observations of multiple systems cannot be correlated.	Proposed	Medium
<input checked="" type="checkbox"/> <b>VF-701</b>		The Clarus system shall be connected to a data input simulator and triggered with a sequence of inputs that includes all required metadata, in addition to inputs that do not include one and two of the required metadata.		Proposed	Medium

Need to determine the availability of the input simulator and ensure it has the desired capability

The EA Specification Manager provides the capability to present the requirement expression in a traditional view while maintaining the information as model elements and associated attributes. Like everything else in EA, if you change the value of an attribute here, it is changed everywhere in the model

Copyright 2020 J.D. Baker, All Rights Reserved 34

# Going Forward



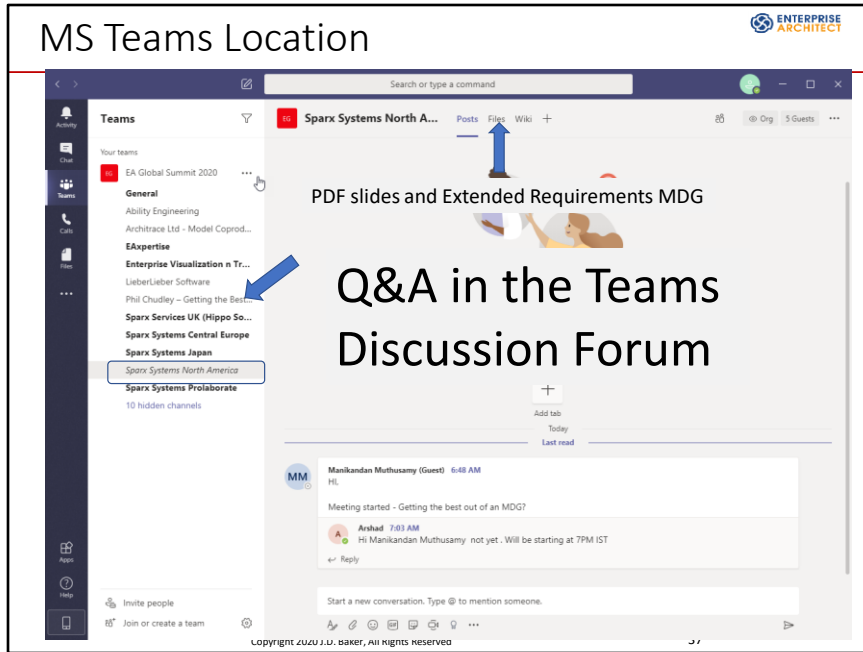
There's more to add to your productivity.

Create your own modeling language with just the right attributes and other modeling support.

Make EA work the way you need it to work.

# Q&A in the Teams Discussion Forum

# MS Teams Location





## Guide for Writing Requirements (Soft Copy)

Digital Download via e-mail link

**Product Code:**  
TechGuideWR2019Soft  
V3, updated 2019

Price: \$25.00  
Member Price: \$0.00

ADD TO CART

DETAILS

Available at the INCOSE.ORG store



## Software Requirements

★★★★★ 6 REVIEWS

by Karl E Wiegers and Joy Beatty

Publisher: Microsoft Press  
Release Date: August 2013  
ISBN: 9780735679658

