



Modeling For Requirements Development

J.D. Baker
James.baker@incose.org

1

copyright © 2011 K-LASSE, Inc.

1/31/2012

“one has to graduate from software engineering into systems/product engineering” - Eric Lasley, BAE Systems

- J.D. Baker's path to graduation

- Aviation Electronics, USN – applying systems engineering principles
 - Not the way I thought about it at the time
- Software Project Engineer – Basic Electronics Maintenance Trainer, satellite ground stations
- Software Technologist – methods and tools
 - OMT, Booch, UML, OOAD
- Process Architect
 - Model-based systems engineering (SysML)
 - Requirements driven software engineering
 - Bridging the gap between systems and software
- System Architect
 - Applying model-based systems engineering practices
- Sr. Product Manager at Seilevel doing requirements modeling/visualization
 - Mapping Seilevel models to OMG technologies, including SysML

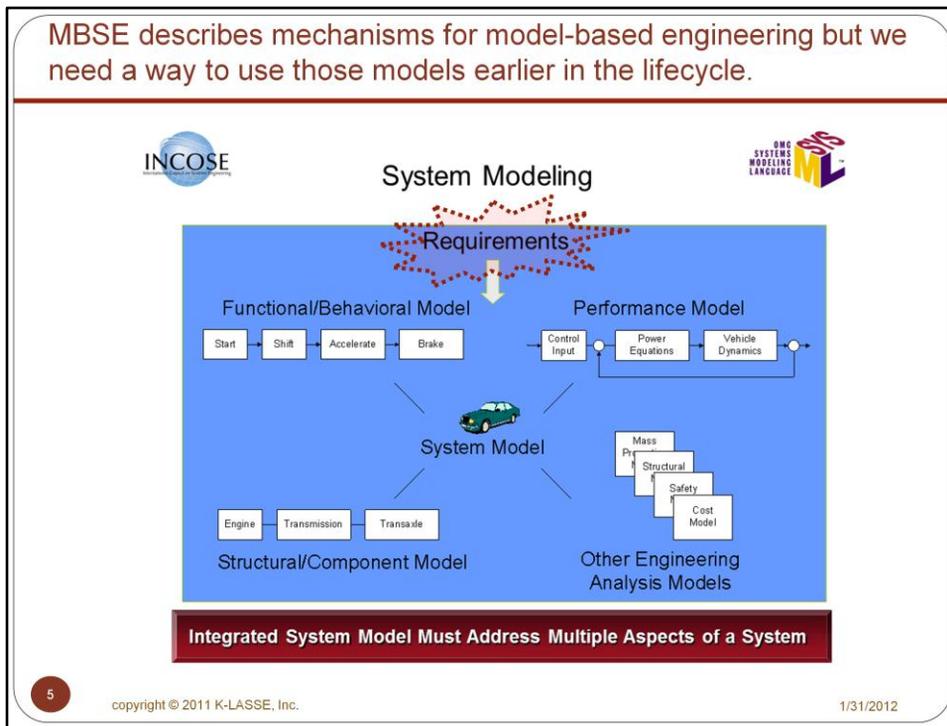


Agenda

- Setting the stage and RML® Overview
- People Models
- System Models
- Data Models
- Adapting the RML® for use with SysML

Setting the Stage

Figuring out what we need to know and why we need to know it.



Despite the fact that many of the people working on the UML Profile for Systems Engineering talked about using models to discover system requirements, most of the methodologies and tutorials talk about interfacing with textual requirement management tools like DOORS and importing and exporting requirements to and from models captured in modeling tools.

After attending a SysML tutorial at the INCOSE International Symposium in 2011 Jeff Grady, notable systems engineer and author of the Universal Architecture Framework came up to me and said – “I thought SysML would enable us to derive requirements from models but all they talked about in the tutorial was an interface to DOORS.” The language is not the problem. The way the language is being used – i.e. the methodology is the problem. You could no more craft good textual requirements if I just offered you an English language dictionary than you could create good model-based requirements if I handed you a SysML specification.

In this tutorial we will explore not only a modeling language but a method for using it to document requirements.

Long lists of natural language system shall statements create too many opportunities for ambiguity and misunderstanding.

Clarus Weather System Design
Detailed System Requirements Specification

ID	Requirement	Source	Allocation - CAS	Comment	Criticality
F-175	The Clarus system shall be able to use multiple algorithms for its quality checking process.	Inferred from ConOps §4.3	QCChS, CAS	Multiple methods or comparisons may be needed for a given observation.	M
F-175B1	The CAS shall be able to configure multiple algorithms to be used in the quality checking process.				
F-213	The Clarus system shall allow access to new surface transportation-related environmental data.	ConOps §1, 2.4, 3.1	CS, QCChS, QEDC, QEDS, CAS, EMC, EMS, DOG	Access could only be provided when new data sources are established and available.	L
F-213B1	The CAS shall be configurable to allow new observation types to be implemented as they become available.				
F-806	The Clarus system shall enable administrators to manage security groups.	MHI	CAS, CAUI	Manage means create, read, update, and delete.	H
F-806B1	The CAS system shall enable administrators to manage security group members.				
F-806B2	The CAS shall have an administrator security group.				
F-806B3	The CAS shall have a quality manager security group.				

6

copyright © 2011 K-LASSE, Inc.

1/31/2012

In my experience with distributed development teams, requirements that consist of only written statements can be confusing. Visualizations that can be referenced by text greatly improve understanding.

A picture is worth a thousand words – more or less

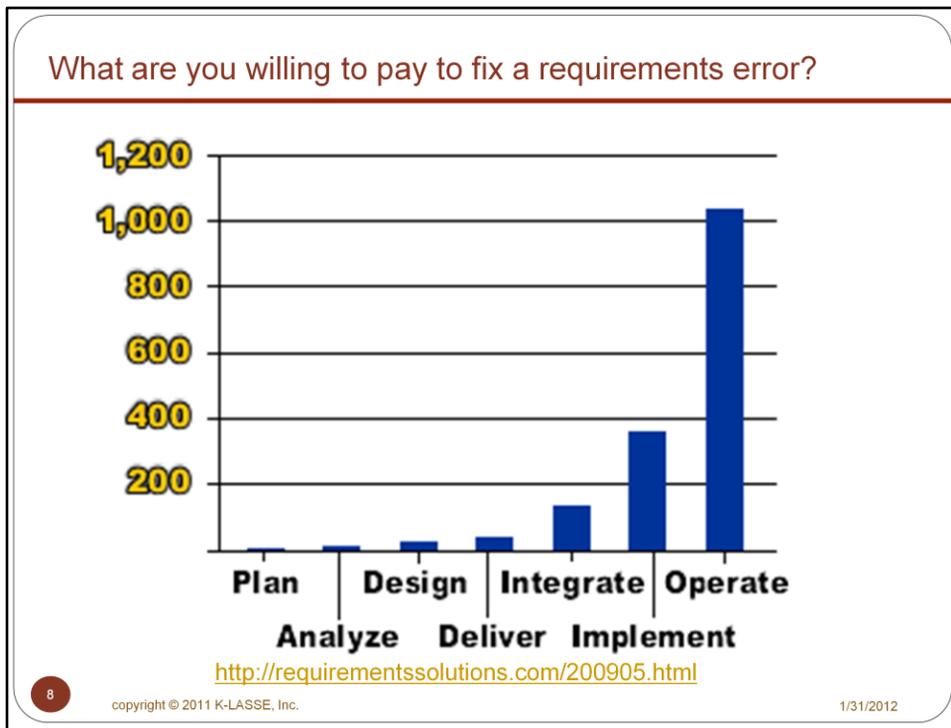
Visualization helps to solve communications problems with stakeholders

This system requirement table for the Clarus Weather System spans 20 pages in a Word document.

I worked with a medical device company that wound up with a 1000 requirements for a single device in a document that spanned several levels of system development (system, sub-system and embedded code). We worked to extract the system requirements into their own document. These were supported by various kinds of visualizations that revealed misunderstandings about requirements for a relatively mature product.



Errors are everywhere – how would you interpret that sentence?



This graph appears in every systems and software requirements book. Although the numbers are different there is general agreement that fixing a requirements error during requirements analysis is significantly cheaper than letting it creep into later development activities.

Visualization will go a long way toward early identification of errors.

Can you find the
the mistake?

1 2 3 4 5 6 7 8 9

reblog if you found it! :)

LIVELOVEREBLOG.TUMBLR.COM

9 copyright © 2011 K-LASSE, Inc. 1/31/2012

Most people focus on the numbers and miss the the

Introducing 

Seilevel is a professional services company that creates software requirements for Fortune 1000 companies.



<http://www.seilevel.com>

10 copyright © 2011 K-LASSE, Inc. 1/31/2012

Seilevel has developed an approach, a set of models and a methodology over the course of several years that they have applied to requirements development for many different types of software projects across many different industries. In addition to the resources available on their web site the company is in the process of writing a book that describes the types of diagrams and how to use them.

You're here to learn how to use visualization in requirements development with RML®.

Seilevel
Requirements defined

RML® Quick Reference

RML® is a language for modeling software requirements to organize and communicate large quantities of information, help identify missing requirements, give context to individual details within the overall collection of requirements, and represent different views of requirements' details.

People Models

Organizational Chart
A chart that identifies every person and/or every role within the organization and how they relate. Org Charts are used to identify all users and stakeholders who might be using the system.

When to use:

- Internal systems with internal users
- Identify users you must interview for requirements
- Identify actors for use cases

When not to use:

- Highly matrixed organizations
- Users are outside the organization

System Models

Context Diagram
A diagram that shows the relationships between the systems in the project.

When to use:

- Provides context on the project
- Understand which integration points are in scope
- Highlight which interfaces requirements activities must cover

When not to use:

- Detailed interactions between systems
- Decision-making process flows

Use Case
Formalized text that describes the interactions between a user and a system, from the user's perspective. Use Cases are used to discover the functional requirements for each step of the interaction.

When to use:

- User and system interactions
- Business review of requirements
- Many alternate scenarios to discover
- Starting point for UAT scripts

When not to use:

- System to system interactions
- Complex branching scenarios

Decision Tree
A diagram that represents complex logic, typically a more detailed representation of a part of a process flow or use case.

When to use:

- Layered "if" statements
- Identify missing decision points or options
- Supplement use cases or process flows

When not to use:

- Sequential process steps
- To imply an order that does not exist

Display-Action Response
A combination of screen representation and tables to describe user interface display and behavior requirements.

When to use:

- User interface representation events
- User interfaces is important to project success

When not to use:

- No user interface

Data Models

Business Data Diagram
A diagram to represent the relationships between business objects from the user's experience.

When to use:

- Logical data entities that relate
- Ensure all data objects are identified for other models

When not to use:

- Data relationships are not relevant to requirements
- Database objects

Data Flow Diagram
A diagram to show the flow of data within systems.

When to use:

- Transaction processing systems
- Many data objects and processing

When not to use:

- Showing decision branches

Data Dictionary
A table used to describe business object attributes from the user's perspective.

When to use:

- Standard data model
- Brand new systems
- Distributed design

When not to use:

- Database design

State Table
A table to show the transition of states of any object in the system.

When to use:

- Ensure you have all states transitions covered
- Audience prefers tabular representations

When not to use:

- Very few states to represent
- To represent the states of the overall system

State Diagram
A diagram to show the transition of states of any object in the system.

When to use:

- Objects with complex status changes
- Audience prefers visual representations

When not to use:

- Very few states to represent

Report Table
A table to capture requirements for reports in a structured format.

When to use:

- Pre-defined reports
- New reports

Do not use:

- Identify a lot of all reports

Structure Models

Topic Matrix
A matrix to track a set of relationships between objects. Topic matrices can represent forward or backward traces.

When to use:

- Find missing requirements (forward)
- Find necessary requirements (backward)

When not to use:

- There is no obtained use for the matrix

Activity Diagram
An organizing tool used to group large amounts of information according to their similarity. Used to create categories and move entities around between the categories, using "1-2-3" as a guideline for the appropriate number of entities per category.

When to use:

- Brainstorming
- Organization is obvious

When not to use:

- Organization is obvious

Resources and Information
Selected Message Boards: seilevel.com/interchangeboards
Requirements Blog: seilevel.com/blog

Seilevel
Requirements defined

11 copyright © 2011 K-LASSE, Inc. 2/4/2012

RML is a relatively mature approach to requirements visualization that has been applied to many software projects for many different organizations. I believe it is an approach that can be applied to MBSE using SysML

Why Seilevel doesn't currently use UML/SysML – many of their customers are very new to modeling and visualization. Visio is an easy entry point.

Their engagements are typically at the beginning of a project. As projects transition to implementation, they transition off of the project. Leaving behind models in Visio provides deliverables that can easily be changed as the requirements change.

UML and SysML tool adopters can utilize the mappings between RML and these languages to maximize their investment in tools and training.

copyright

11

Modeling Principles – choosing what models to create



It is important to choose well



12

copyright © 2011 K-LASSE, Inc.

1/31/2012

The term model is overloaded. Some people talk about a system and the many models of the system. Others talk about the model of the system and the many views of that model.

¹ Booch, G, J.Rumbaugh and I. Jacobson; *The Unified Modeling Language User Guide*; Addison-Wesley; 1999.

The choice of what models to create has a profound influence on how a problem is attacked and how a solution is shaped

Every model may be expressed at different levels of precision

You need to keep it real and simple



13 copyright © 2011 K-LASSE, Inc. 1/31/2012

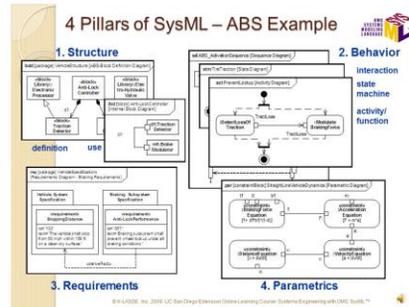
The best models are connected to reality

No single model is sufficient. Every nontrivial system is best approached through a small set of nearly independent models

These principles can be applied no matter what process or method you are using

An adaptation of the KISS principle also applies - every artifact should be created to help solve the problem, not just because it's in the process

You need to keep it real and simple



14

copyright © 2011 K-LASSE, Inc.

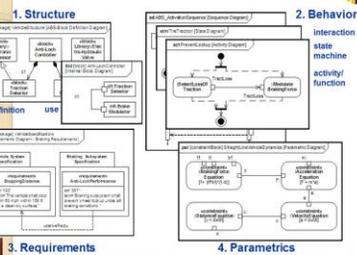
2/4/2012

No single model is sufficient. Every nontrivial system is best approached through a small set of nearly independent models

You need to keep it real and simple



4 Pillars of SysML – ABS Example



"Bureaucrats write memoranda both because they appear to be busy when they are writing and because the memos, once written, immediately become proof that they were busy." Charles Peters

15

copyright © 2011 K-LASSE, Inc.

2/4/2012

These principles can be applied no matter what process or method you are using
An adaptation of the KISS principle also applies - every artifact should be created to help solve the problem, not just because it's in the process

Characteristics of Useful Models 1

- Abstract
 - Emphasize important aspects while removing irrelevant ones
- Understandable
 - Expressed in a form that is readily understood by observers
- Accurate
 - Faithfully represents the modeled system
- Predictive
 - Can be used to derive correct conclusions about the modeled system
- Inexpensive
 - Much cheaper to construct and study than the modeled system

¹ Bran Selic, IBM

To be useful, engineering models must satisfy all of these characteristics

16

copyright © 2011 K-LASSE, Inc.

1/31/2012

Follow these directions

Driving directions to Austin, TX

625 Amigos Rd

Ramona, CA 92065

1. Head south on Amigos Rd toward Amigos Ln
157 ft

2. Turn left to stay on Amigos Rd

0.2 mi

3. Turn right onto W Old Julian Hwy

0.4 mi

4. Continue onto 3rd St

0.7 mi

5. Turn left onto Main St

4.2 mi

6. Continue onto CA-67 S

14.9 mi

7. Turn left onto Maplevue St

0.8 mi

8. Continue onto Lake Jennings Park Rd

2.3 mi

9. Merge onto I-8 E via the ramp to El Centro

Entering Arizona

326 mi

10. Take exit 178A to merge onto I-10 E toward

Tucson

Passing through New Mexico

Entering Texas

11. Take exit 477 to merge onto US-290 E
toward Fredericksburg/Johnson City/Austin

71.7 mi

12. Slight right onto US-281 S/US-290 E

5.6 mi

13. Slight left onto US-290 E

34.9 mi

14. Take the exit onto TX-1 Loop N

4.7 mi

15. Take the Cesar Chavez St exit

0.1 mi

16. Keep left at the fork, follow signs for W 5th
St and merge onto W 5th St

1.8 mi

17

copyright © 2011 K-LASSE, Inc.

1/31/2012

Follow the map



Can you answer any questions from looking at the map that you couldn't answer from the written directions?

18 copyright © 2011 K-LASSE, Inc. 1/31/2012

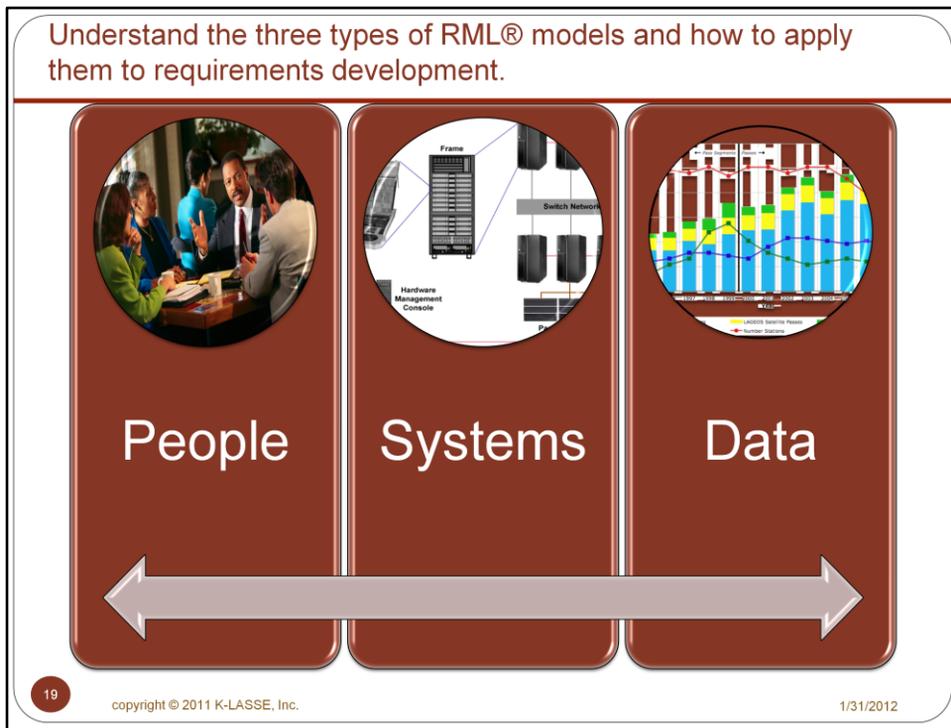
See if they notice things like:

How close to Mexico?

What cities are on the route?

<click>

Ciudad Juarez is a city that has been in the news. Do you want to adjust your route?



RML is a registered mark of Seilevel, Inc.

Seilevel has intentionally chosen to use Visio and other MS Office products as their modeling tools. This is in part due to their business model in which they often engage at the beginning of a project to assist with requirements development and then leave behind the modeling artifacts to be maintained by the customer. In addition, RML includes diagrams that are not supported in typical UML/SysML modeling tools. I have mapped the RML to OMG technologies and this presentation will illustrate the adaptation of RML to the systems engineering domain using UML/SysML modeling tools.

<click>

RML consists of models that are categorized as People,
People models are those associated with users and other stakeholders

<click>

Systems

Systems models are those related to both system interconnections and business process execution

<click>

and Data

Data models are those that show relationships among data elements, data transformations and how that data is used

RML® is a related set of models used for software requirements engineering.

20 copyright © 2011 K-LASSE, Inc. 1/31/2012

RML is a registered mark of Seilevel, Inc.

Seilevel has intentionally chosen to use Visio and other MS Office products as their modeling tools. This is in part due to their business model in which they often engage at the beginning of a project to assist with requirements development and then leave behind the modeling artifacts to be maintained by the customer. In addition, RML includes diagrams that are not supported in typical UML/SysML modeling tools. I have mapped the RML to OMG technologies and this presentation will illustrate the adaptation of RML to the systems engineering domain using UML/SysML modeling tools.

<click>

RML consists of models that are categorized as People,
People models are those associated with users and other stakeholders

<click>

Systems

Systems models are those related to both system interconnections and business process execution

<click>

and Data

Data models are those that show relationships among data elements, data transformations and how that data is used

People

```

graph TD
    President[President] --> Manufacturing[Manufacturing]
    President --> Sales[Sales]
    President --> RnD[RnD]
    Manufacturing --> Manufacturing_Mgr[Manufacturing Manager]
    Manufacturing --> Plant_Plant_Mgr[Plant Plant Manager]
    Manufacturing --> Sales_Sales_Mgr[Sales Sales Manager]
    Manufacturing --> Chemical_Chemical_Mgr[Chemical Chemical Manager]
    Manufacturing --> Ops_Ops_Mgr[Ops Ops Manager]
    Manufacturing --> QA_QA_Mgr[QA QA Manager]
    
```

State Key	System	Related System
U-1	U-1	1. Sales Req. system type in query
U-2	U-2	2. System request state in query for related work item
U-3	U-3	3. Sales Req. and change state information, including requests and approvals
U-4	U-4	4. System request state information in response
U-5	U-5	5. Sales Req. request in query with Request case Manager (Case M.)
U-6	U-6	6. Sales Req. state query in detail
U-7	U-7	7. System request state and associated Case Manager (Case M.)

```

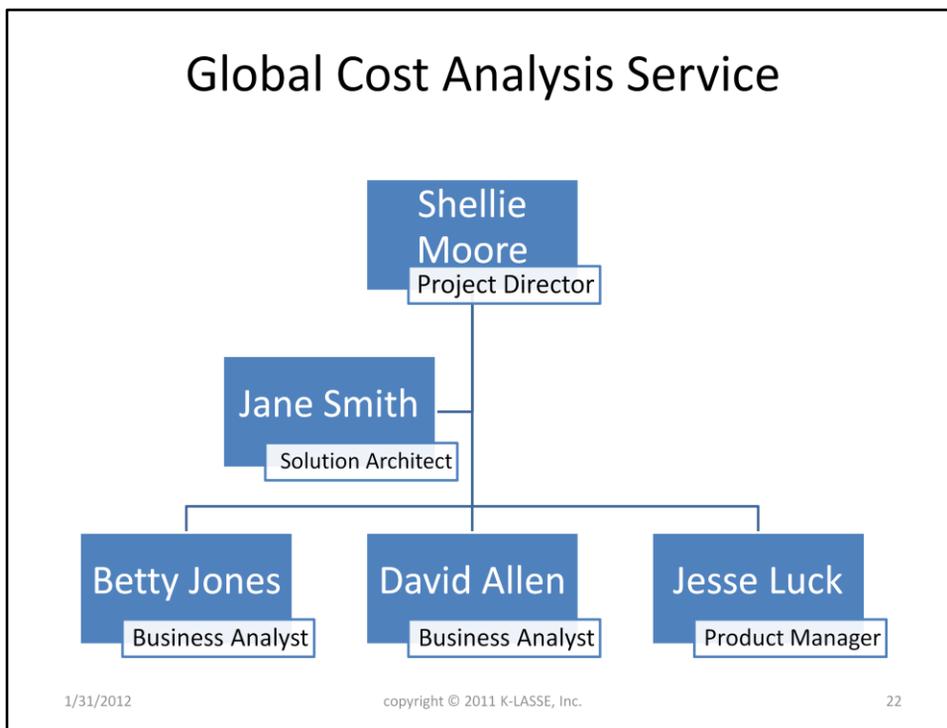
graph TD
    Start(( )) --> D1{Decision 1}
    D1 -- Yes --> R1[Request Configuration]
    D1 -- No --> D2{Decision 2}
    D2 -- Yes --> R2[Request Configuration]
    D2 -- No --> D3{Decision 3}
    D3 -- Yes --> R3[Request Configuration]
    D3 -- No --> R4[Request Configuration]
    
```

21

copyright © 2011 K-LASSE, Inc.

1/31/2012

Org charts
 Use cases
 Decision trees



The org chart can be built with many different tools. Although the appearance is not the primary importance, this can become a deliverable so it needs to be professionally prepared.

In large organizations there are often times when the architects and business analysts don't know who the stakeholders are. I worked on a project where we were creating the requirements for a shipping service. The initial deployment was to Latin America. As the Product Manager I had to identify people responsible for costing freight, setting the price customers would pay, controlling extended logistics services, etc. An org chart was one of the first things we put together, and it was a living document that was revisited once or twice a month.

Org Chart Basics

- When to use:
 - Internal systems with internal users
 - Identify users you must interview for requirements
 - Identify actors for use cases
- When not to use:
 - Highly matrixed organizations
 - Users are outside the organization

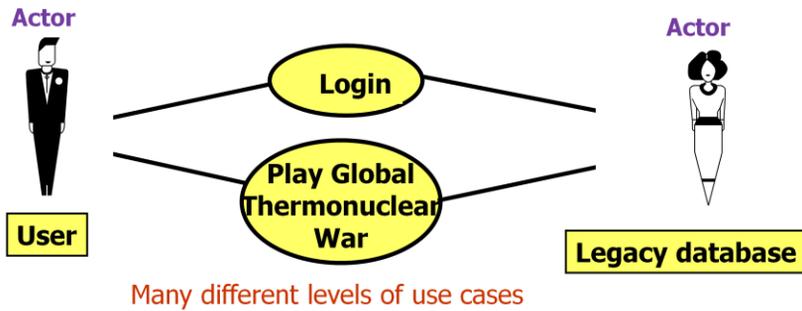
23 copyright © 2011 K-LASSE, Inc. 1/31/2012

Even when users are outside the organization an org chart can be very useful to identify system stakeholders but it will need to be a generic org chart

Emergency Room Dr. vice Dr. John Jones

Use Case Diagram

- The detail is in the written document that is “connected” to the diagram
 - Usually a separate document maintained in a word processor or on-line editor



24

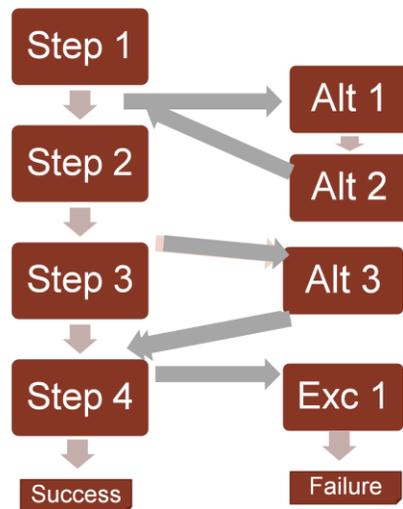
copyright © 2011 K-LASSE, Inc.

1/31/2012

Other icons can be substituted for the Actors and should be used when they add understanding to the diagram.

Beware of getting fancy at the expense of understanding such as happens when we substitute a person for an inanimate object.

Use Case Descriptions / Scenarios



One use case with 2 alternate paths and one exception path.

How many scenarios?

25

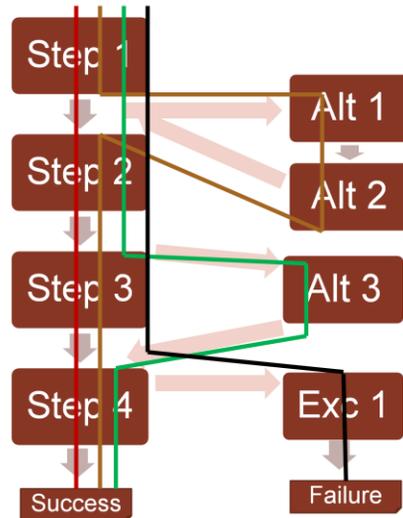
copyright © 2011 K-LASSE, Inc.

1/31/2012

25

A diagram of a use case with two alternative paths and one exception path. How many scenarios?

Use Case / Scenarios



One use case.

How many scenarios?

1

2

3

4

2 + 3

2 + 3 + 4

2 + 4

3 + 4

Here are 8 scenarios that can result from that one use case. Use cases can be built from the bottom up, by identifying a collection of scenarios and abstracting them into a use case. You can also build them from the top down, by identifying the basic flow and then asking at each step, what are the alternatives or exceptions that can occur here?

Use Cases Do ...

- Document behavioral requirements
- Represents the goal of an interaction between an actor and the system.
- Record a set of success scenarios
- Record a set of failure scenarios
- Provide multi-level organization

- Use Cases Do Not ...
 - Specify user interface design
 - Specify implementation detail (unless it is of particular importance to the actor to be assured that the goal is properly met)

27

copyright © 2011 K-LASSE, Inc.

1/31/2012



Use Cases Do ...

It is important to recognize what use cases are intended to accomplish.

As use cases become more familiar to the industry, the use case text is being used as a replacement for a portion of the requirements documentation. It clearly specifies the behavior of the system but does not address the non-behavioral requirements as listed in the “Requirements Document” section of this course.

Use Cases Do Not ...

Specification of implementation within the use case text makes the use case text much harder to maintain. As implementation decisions change, the actual behavioral requirements of the system are unaffected. As long as the behavioral requirements are the same, the use case text should not require change.

Application of Use Cases

- To capture the requirements of the system
- To act as a springboard for the system design
- To validate the system design
- For system testing and QA
- Initial draft of the user manual

28

copyright © 2011 K-LASSE, Inc.

1/31/2012



Application of Use Cases

A correctly written use case document should provide this support in a use case-driven process:

- It can be handed to the developers as a description of behavioral requirements
- It can be handed to the testers as a draft test plan
- It can be handed to the tech writers as a draft user guide

Use Case

- Scope
 - Enterprise
 - Of value to the organization
 - Shows how the system is used to benefit the organization
 - System
 - Bounded by system under design
 - Goals represent specific functionality
 - Often steps in the strategic use cases
- Levels
 - Summary Level Use Case
 - Collection of User Level goals
 - User Level Use Case
 - A user task or elementary business function
 - Sub-function Level Use Case
 - Below the main level of interest



29

copyright © 2011 K-LASSE, Inc.

1/31/2012

Use Case Scope

It is critically important when writing use cases to identify the scope of the use case that you are describing. Otherwise, you will be in danger of writing a document that varies in its scope and is therefore less precise and focused.

Use Case Levels

These levels are defined by graphic symbols in Cockburn's process

Cloud (white) – Highest level summary

Kite (white) – Lower level summary

Waves (blue)- user goal

Fish (indigo) – sub-function

These levels help to define a hierarchy of use cases

The Role of the Actor

- Four potential roles that an actor may play in a use case
 - Initiator - role is to set the use case in motion
 - Server - provides one or more services necessary to achieving the goal of the use case
 - Receiver - role is to receive information from the system
 - E.g data warehouse
 - Facilitator - performs an action on behalf of another actor in a system
- Actors may play multiple roles simultaneously
- An actor may be human or machine
- May be anonymous or known
 - Anonymous – all users can read some information
 - Known – only one user can access an instance of account information

30

copyright © 2011 K-LASSE, Inc.

1/31/2012



From *Writing Effective Use Cases*

If an actor is anonymous, its identity has no impact on the system. An end user may play many roles in a system without ever being identified; likewise, regardless of which end user plays that role -- Tom, Mike, or Judy -- he or she will experience the exact same functionality. Machines may also act as anonymous actors, particularly in the Web services domain.

In contrast, a system may require identifying information in order to handle matters such as security or quality of service. In these cases, the actor must be known. Anytime that a system requires information about an actor -- whether that information is a precise identification or just some particular information -- that actor is considered to be a known entity.

Actor analysis from "*The Handbook of Program Management*", by James T. Brown

- *Follow the money!*
*Whoever is **paying** is definitely a stakeholder.*
- *If the program produces savings or additional costs for an organization then the organization is also a stakeholder*



Actor Analysis

- *Follow the resources.*
 - *Every entity that provides resources, whether internal or external, labor or facilities, and equipment, is a stakeholder.*
 - *Line managers and **functional** managers providing resources are stakeholders*



Actor analysis from "*The Handbook of Program Management*", by James T. Brown

Actor Analysis

- *Follow the deliverables. whoever is the recipient of the product or service the program is providing is a stakeholder.*



Actor analysis from "*The Handbook of Program Management*", by James T. Brown

Actor Analysis

- *Follow the signatures. The individual who signs off on completion of the final product or service (or phases thereof) is a stakeholder.*
- *Note: this may or may not be the recipient referred to in the previous bullet. Often there may be more recipients than signatories.*

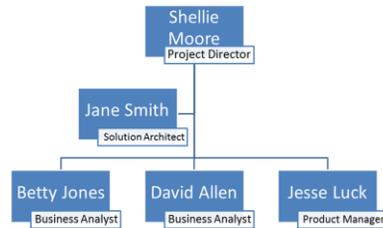


Actor analysis from "*The Handbook of Program Management*", by James T. Brown

Actor Analysis

- Review the organizational chart to assess which parts of the organization may be stakeholders.

Global Cost Analysis Service



Actor analysis from "*The Handbook of Program Management*", by James T. Brown

Actor Analysis

- *Ask team members, customers, and any other confirmed stakeholder to help you identify additional stakeholders.*



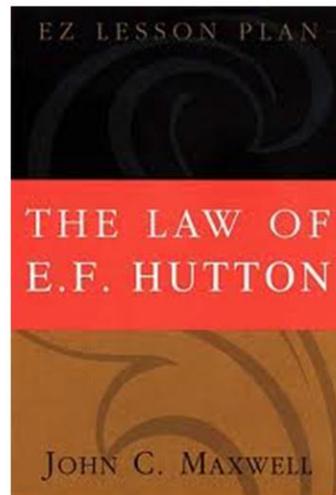
36 copyright © 2011 K-LASSE, Inc. 2/3/2012

This works well in large organizations. There can be key personnel and parts of the organization who are not readily apparent that need to be included. I always ask if there is anyone else who might be interested and pay close attention to peo

Actor analysis from "*The Handbook of Program Management*", by James T. Brownple who are added in email replies.

Actor Analysis

- Look for the "Unofficial People of Influence".
 - These may be people who are trusted by high-level leaders
 - People who wield a lot of power through influence and not position.



Actor analysis from "The Handbook of Program Management", by James T. Brown

Decision Tree

```
graph TD; D1{ } -- Yes --> T1[ ]; D1 -- No --> D2{ }; D2 -- Yes --> T2[ ]; D2 -- No --> D3{ }; D3 -- Yes --> T3[ ];
```

When to use:

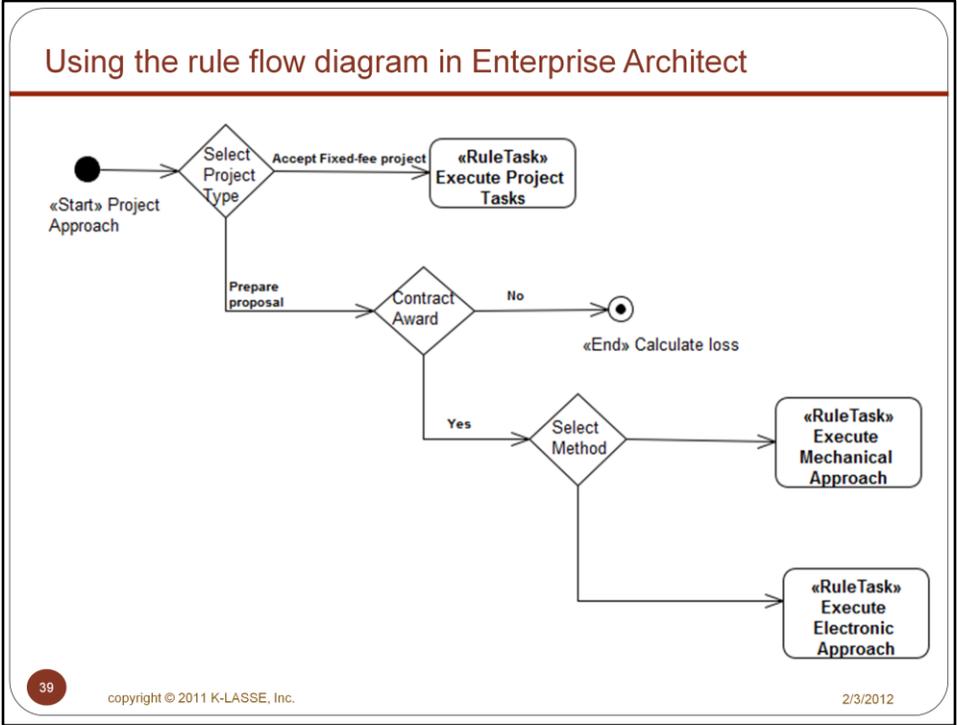
- Layered “if” statements
- Identify missing decision points or options
- Supplement use cases or process flows

When not to use:

- Sequential process steps
- To imply an order that does not exist

38 copyright © 2011 K-LASSE, Inc. 1/31/2012

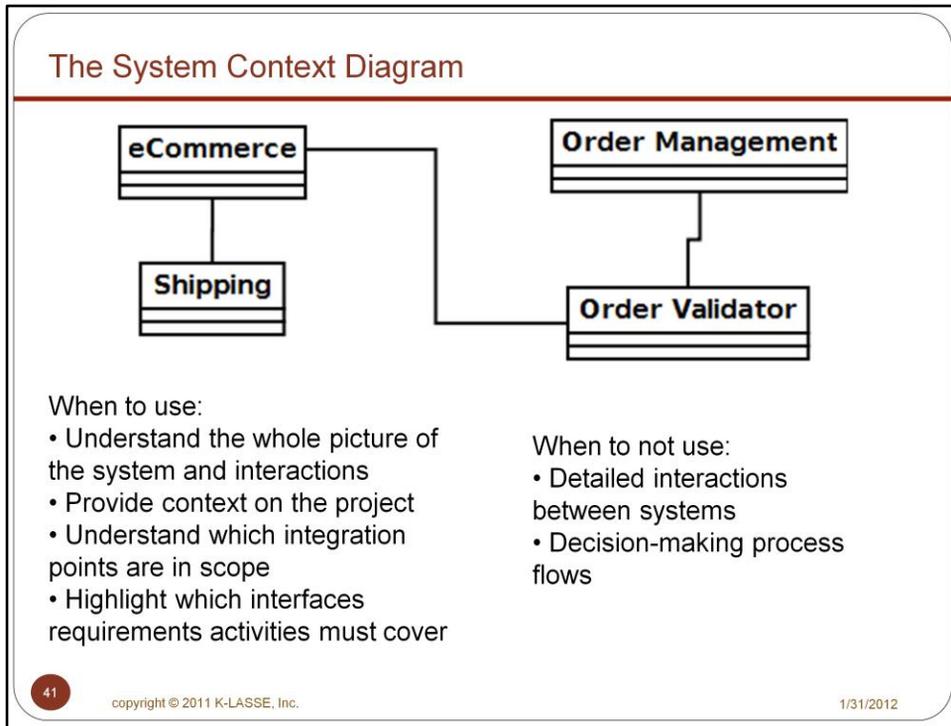
This is a simple tree – more complex trees are possible



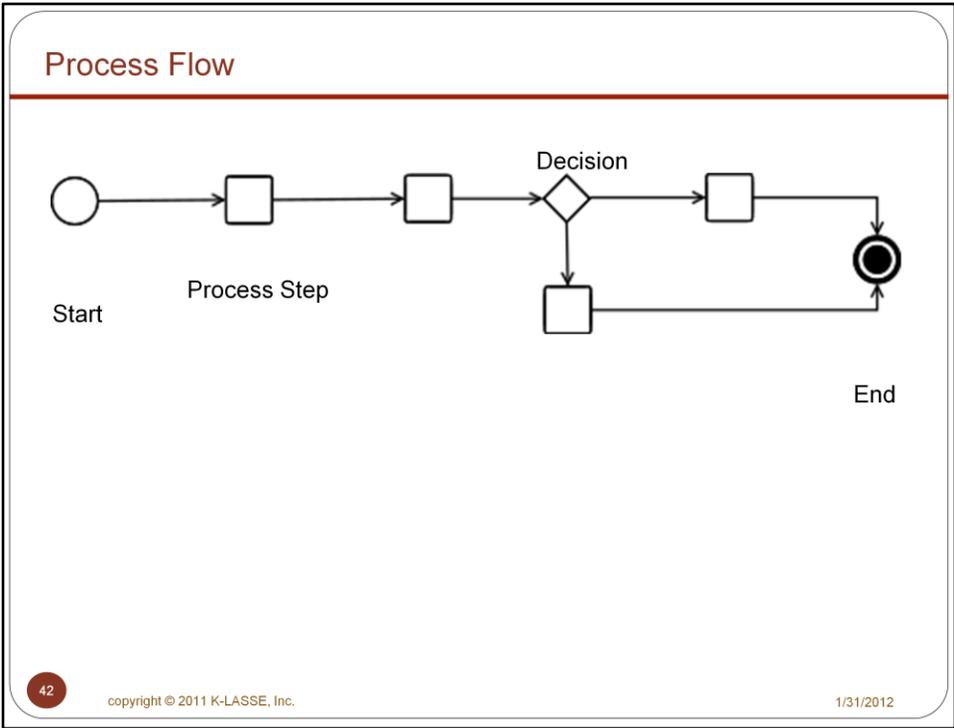
The rule flow diagram is not a standard UML or SysML diagram.

Systems

Modeling the systems and their interactions



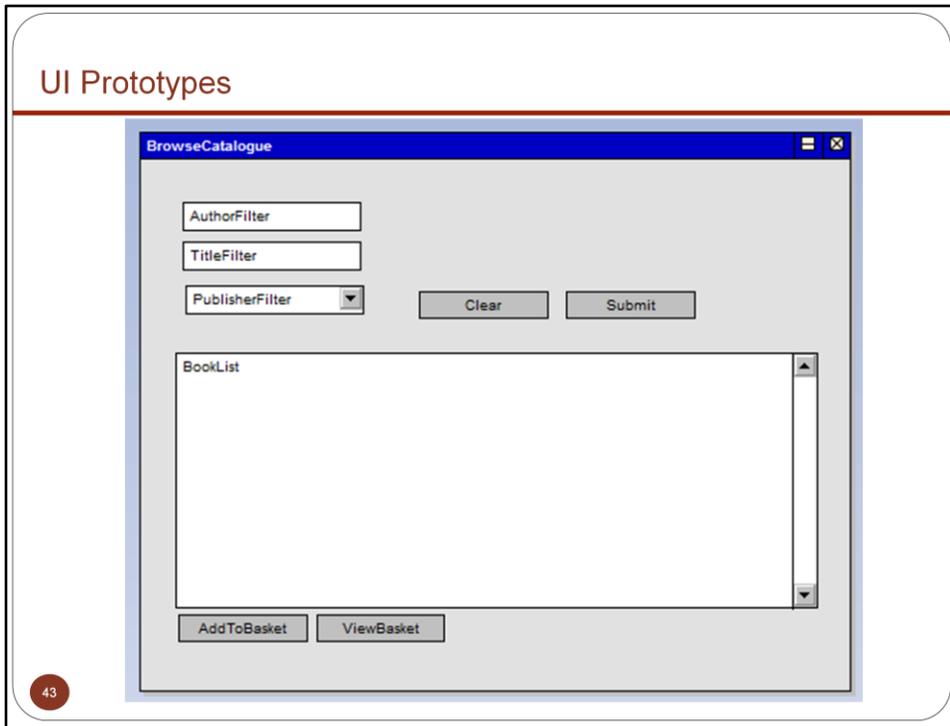
Jump out to domain model in RML demo2



Work on the whiteboard a L1 process to purchase a product

Jump out to the RML-Demo in EA

UI Prototypes



Discuss Low, Medium and High Fidelity Prototypes & that we are only going to cover Wireframes

When to Use

As an adjunct to validating or creating use cases and/or an BDD

As an adjunct to eliciting functional requirements and business rules

When Not to Use

Before the business objectives and use cases have been identified

With systems that have no UI

How to Derive Requirements From

Creation of the screen represented by the wireframe

Possible navigation options

Possible actions a user could perform and the system responses

Layout decisions – if area X must always appear to the right of area Y, etc.

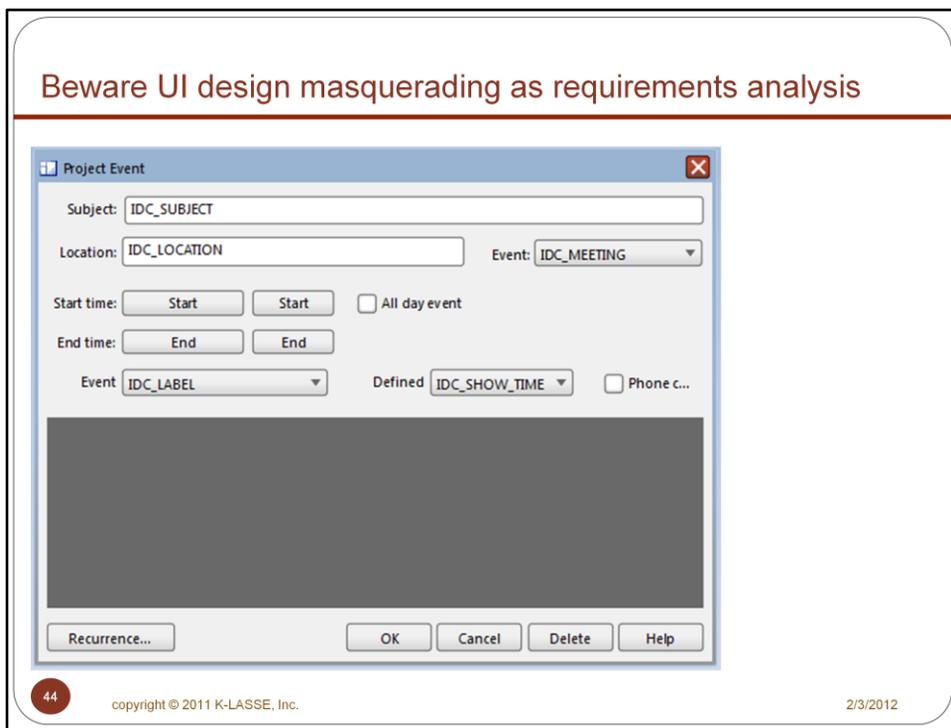
Non-functional requirements covering access/display modifications for the disabled, response times

The data that needs to be displayed

The business rules that apply to display and entry of the various data elements

Tools: Visio, Powerpoint

Beware UI design masquerading as requirements analysis



Some modern wireframe tools can produce very real looking prototypes. Unless these are created by UI designers you run the risk of implementing a sub-optimal user interface.

Data

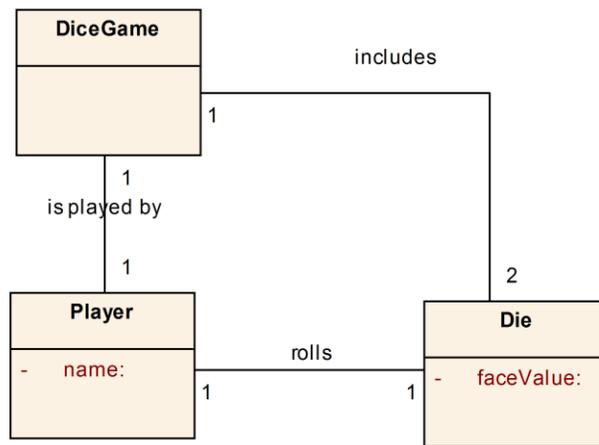
Modeling the business data

45

copyright © 2011 K-LASSE, Inc.

1/31/2012

Business Data Diagrams



46

© K-LASSE, Inc. 2009 UC San Diego Extension Online Learning
Course: Systems Engineering with OMG SysML™

Example: Dice Game

In the domain of the dice game, the conceptual model shows the concepts Player, Die, and DiceGame, their associations and attributes.

When to use:

- Logical data entities that relate
- Ensure all data objects are identified for other models

When to not use:

- Data relationships are not relevant to requirements
- Database objects

Data Dictionary for business use, not database design



<http://www.flickr.com/photos/crdot/5510506796/> Caleb Roenigk

What are they

- Detailed description of business objects
- Not intended to be a database representation
- Requirements may exist at the data object level or on individual fields

When to use

Undefined data model

Brand new systems

Can be used for database design but that happens after requirements elicitation and analysis and usually means much more detail. What we want to do during requirements modeling is to capture the characteristics of the business objects.

Data Dictionary example

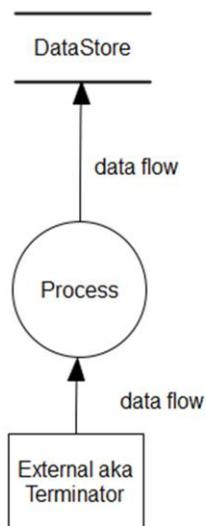
	A	B	C
1	Adverse Reaction Assessment Data		
2	Field Name	DataType	Field Description
3	AdverseReactionAssessmentDateTime	date time	Non-blank. This field contains the date/time of the last reaction assessment for this patient.
4	AdverseReactionAssessmentID	numeric	Protocol ID from handbook
5	AdverseReactionAssessmentPerformed	text	Description of protocol. This field indicates if the patient has a reaction on file or not. 1:YES;0:NO;
6	AssessingStaffID	numeric	Unique ID assigned by HR. This field contains the name of the user who made the last reaction assessment for this patient
7	DateExtracted	date time	Value assigned by Data Warehouse indicating datetime the record was extracted from the source database (e.g. VistA)
8	PatientID	numeric	Unique ID from EHR. This is the patient who has been asked about allergies/adverse reactions
9	Site	numeric	Value assigned by Data Warehouse indicating three-digit VA facility number (e.g. 663)

48 copyright © 2011 K-LASSE, Inc. 2/4/2012

An example of data necessary for a specific type of patient report

This type of information collection is supported in UML as a table diagram which by it's very nature drives toward design. If a modeling tool is used, it is best to collect this information in a tool like Excel and link the file to the model in order to avoid premature design effort.

Data Flow Diagrams (DFD)



- 4 intuitive elements
 - Store – shows a collection of data packets at rest
 - Flow – shows the movement of data, i.e. it shows data in motion
 - Process – transforms inputs into outputs
 - Terminator – external entities, typically a person or group

49

DFDs have existed as a modeling tool since the late 1960's where they were introduced as part of structured analysis. They are a simple tool that highlights the movement and transformation of data. They don't show process flow but there should be process flow diagrams for the processes. They don't define data relationships – there should be data object diagrams that include each of the objects on a DFD.

What are they

Show flow of data within and between systems

Consider each process to discover requirements

When to use

Transaction processing systems

Systems with lots of data objects and processing

Do not use for

Showing decision branches

DFD purpose and audience



: "Mr. User, I'd like to show you a top-down, partitioned, graph-theoretic model" ¹

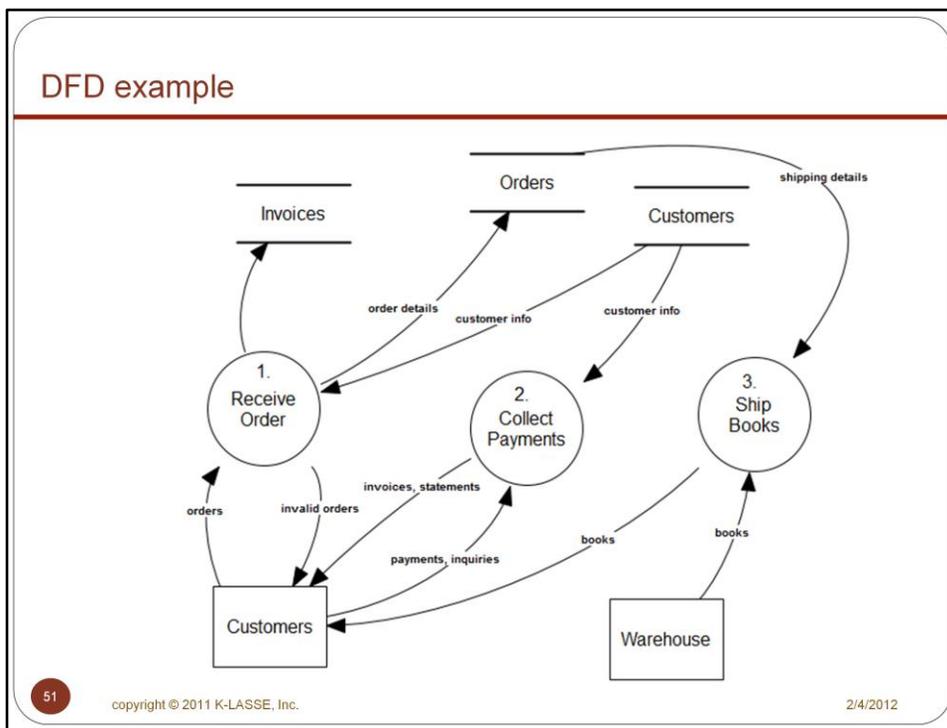
¹Modern Structured Analysis, page 140

50 copyright © 2011 K-LASSE, Inc. 2/4/2012

How not to start from Yourdon's book

Focus on the users. The intention is to communicate with them to elicit requirements, not other analysts or developers.

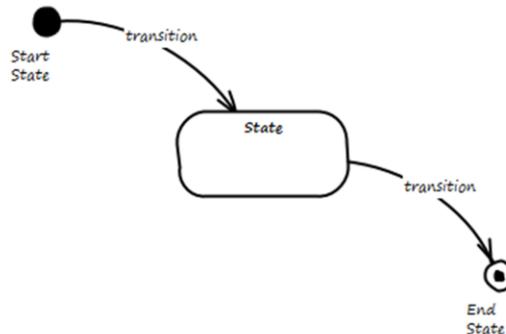
Don't use DFDs where BDDs or state diagrams would be more appropriate



From Modern Structured Analysis, ED Yourdon

Best practice is to number the processes. This does not provide sequencing information but it is a way to disambiguate processes. It also provides a way to link when decomposing processes.

State Diagram Elements



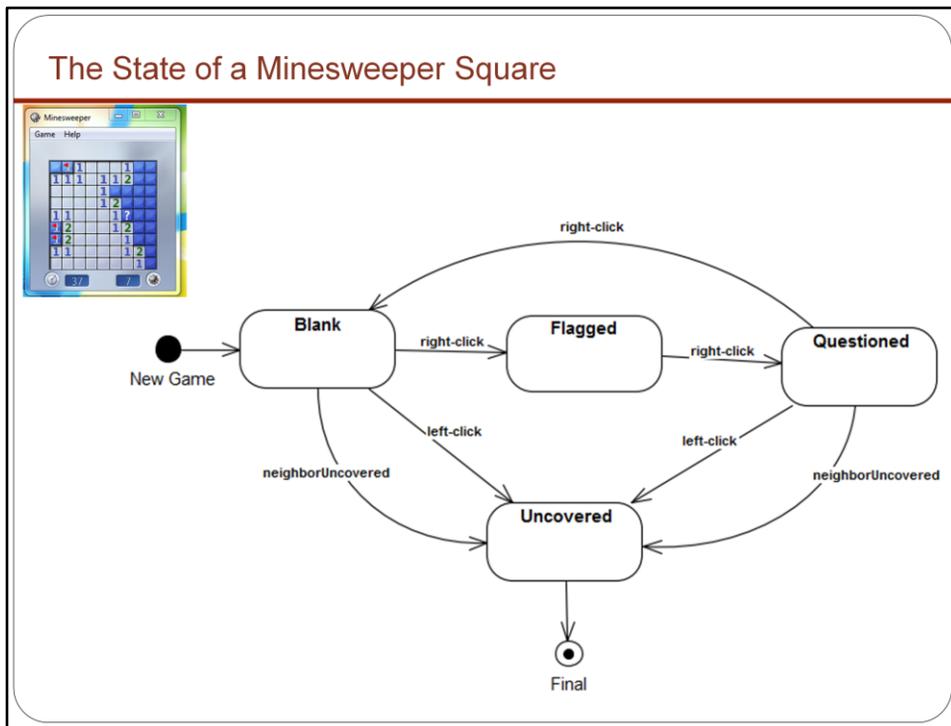
What are they

- State Machines illustrate how an element can move between states
- Used to capture the lifecycle of an object
- Events trigger state changes along transitions

When to use

- Not all objects need a state diagram
- Objects with complex or interesting state changes

Often it's not obvious that understanding our requirements would benefit from creating a state diagram. If you find yourself having difficulty how an object changes during a process or which events trigger a change in what configuration, then a state diagram is a good choice for visualization. Sometimes, however, it is obvious. The bottom line is that like any artifact, you should know why you need to create a state diagram, that is what questions it will help answer, before you create it.



Minesweeper: rules and basics

The object

Find the empty squares while avoiding the mines. The faster you clear the board, the better your score.

The board

Minesweeper has three standard boards to choose from, each progressively more difficult.

Beginner: 81 tiles, 10 mines

Intermediate: 256 tiles, 40 mines

Expert: 480 tiles, 99 mines

You can also create a custom board by clicking the Game menu, and then clicking Options.

Minesweeper supports boards of up to 720 squares and 668 mines.

How to play

The rules in Minesweeper are simple:

Uncover a mine, and the game ends.

Uncover an empty square, and you keep playing.

Uncover a number, and it tells you how many mines lay hidden in the eight surrounding squares—information you use to deduce which nearby squares are safe to click.

Hints and tips

Mark the mines. If you suspect a square conceals a mine, right-click it. This puts a flag on the square. (If you're not sure, right-click again to make it a question mark.)

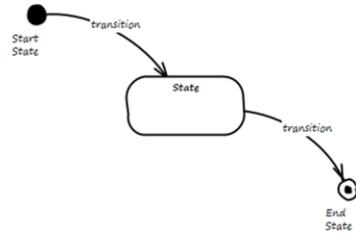
Study the patterns. If three squares in a row display 2-3-2, then you know three mines are probably lined up beside that row. If a square says 8, every surrounding square is mined.

Explore the unexplored. Not sure where to click next? Try clearing some unexplored territory.

You're better off clicking in the middle of unmarked squares than in an area you suspect is mined.

State Tables

Next State		Start State	State	End State
		S0	S1	S2
Start State	S0		<u>event 1</u>	
State	S1			<u>event 2</u>
End State	S2			



State Tables – Short Notes

What are they

Another way to show state transitions

Minesweeper State Table

Next State \ State		New Game	Blank	Flagged	Questioned	Uncovered	Final
		S0	S1	S2	S3	S4	S5
New Game	S0		_____				
Blank	S1			right-click _____		neighborUncovered _____	
						left-click _____	
Flagged	S2				right-click _____		
Questioned	S3		right-click _____			neighborUncovered _____	
						left-click _____	
Uncovered	S4						_____
Final	S5						

A state table shows all possible system states and the events that cause the transitions between them.

Some users are better able to relate to tables than they are to diagrams. Using a tool like EA ensures that the two representations are synchronized.



There are five basic questions that a report developer should address prior to opening Crystal Reports. They are:

What? – The purpose of the report, the desired end result or distribution method (for example, PDF, Excel export, TXT file, CSV file, email, and so on), aggregations, formulas, grouping, and sorting.

Who? – The audience, as in who will be viewing the data, management versus staff level. Reports should be designed for the user and not for the developer.

When? – Determine the timeframe of the report and frequency. This question will determine the need for a date field or a parameter.

Why? – This question sometime draws a negative response. It's important to ask why the user needs this data in order to ensure the report is pulling the correct data. Often people think they know what they need; however, when it is provided it does not meet their needs. It is imperative that the report developer understand the database that they are utilizing for their reports. This will ensure that they clearly understand the needs of the business users.

How? – Where is the information coming from? What data sources will it be coming from? Is it new data or an existing data source? This will require that you have

knowledge of the database, including a data schema or data dictionary that lists and explains the tables and fields.

These answers should be compiled and recorded on a Report Requirements document. There are many sources for a Report Requirements template.

Above text from Crystal Reports® 2011 for Developers: Report Design and Integration
By: Cynthia Moore

Report Tables

[University of Melbourne Template](#)

Full Report Name	
Short Report Name	This might be a reference ID or just a common name for a report
Purpose	What questions is this report intended to answer.
Assumptions	Document any assumptions that are important for understanding the context and use of these requirements
Dependencies	List any activities or deliverables that the successful implementation of this report is dependent upon.
Report Audience	Provide a description of the stakeholders who will be generating and/or viewing the report.
Standard Delivery Cycle	Common examples are: Ad-hoc, Daily, Weekly, Monthly.
Standard Distribution Method	Common examples are: Online, Print, Email.
References	A reference to a parent Business Requirements document.
Report Criteria	List inputs to the report. Note that this usually only applies to online reports and is referred to by some as the parameters of the report.
Output Data Requirements	List all of the possible output fields that the report may produce.
Default Sort Order	If there is a sort order
Grouping	How the report's output should be grouped if this is needed
Totals	The derived totals to be included in the report's output.

This table derived from the University of Melbourne template

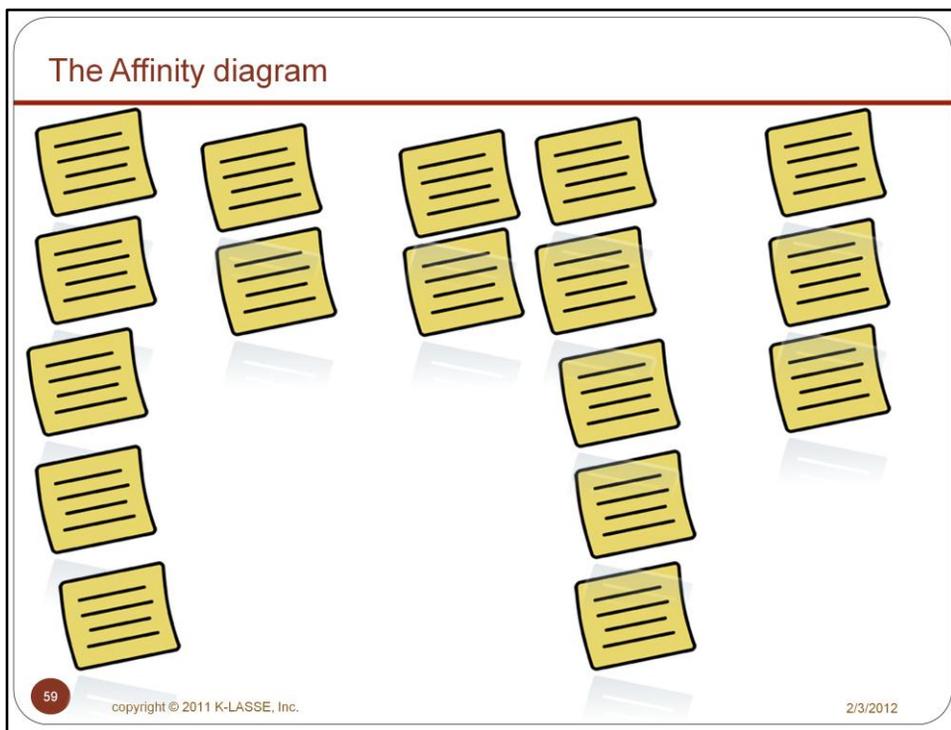
http://www.its.unimelb.edu.au/__data/assets/word_doc/0011/398684/0000-Business_Analysis_Template-Business_Report_Requirements-v100.doc

Other Useful Models

58 copyright © 2011 K-LASSE, Inc. 1/31/2012

Glossary

?



Grouping like items together

When to use:

Brainstorming

When not to use:

Organization is obvious

Often used in brainstorming – simple approaches like put ideas on sticky notes and group/regroup them on a wall or piece of flip chart paper.

Adapting RML® to SysML

60

copyright © 2011 K-LASSE, Inc.

1/31/2012

RML® Mapped to UML/SysML

RML®	UML/SysML
Organization Chart	Class Diagram/UPDM
Process Flow	BPMN Profile
Use Cases	Use Cases
Context Diagram	Class/Block Diagram
Business Data Diagram	Class Diagram
Data Flow Diagram	Activity Diagram
State Diagram	State Diagram
State Table	Matrix
Decision Trees	<i>Decision Model and Notation (WIP)</i>

RML® Mapped to UML/SysML

RML®	UML/SysML
Affinity Diagrams	Packages/Package Diagrams
Traceability Matrix	Matrix
Data Dictionary	
Report Table	
Decision Action Response	<i>Interaction Flow Modeling Language (WIP)</i>
Wireframes	<i>Interaction Flow Modeling Language (WIP)</i>

Use the RML_Demo2 model to illustrate these.

What about the SysML requirements diagram?

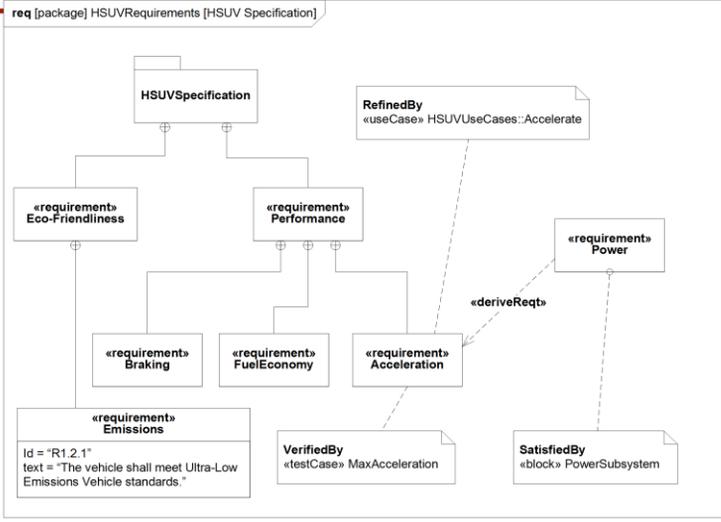
- The «requirement» stereotype represents a text based requirement
 - Includes id and text properties
 - Can add user defined properties such as verification method
 - Can add user defined requirements categories (e.g., functional, interface, performance)
- Requirements hierarchy describes requirements contained in a specification
- Requirements relationships include DeriveReq, Satisfy, Verify, Refine, Trace, Copy

63

© K-LASSE, Inc. 2009 UC San Diego Extension Online Learning
Course: Systems Engineering with OMG SysML™

In other words the SysML requirement models requirements AFTER they have been discovered.

Requirements Breakdown

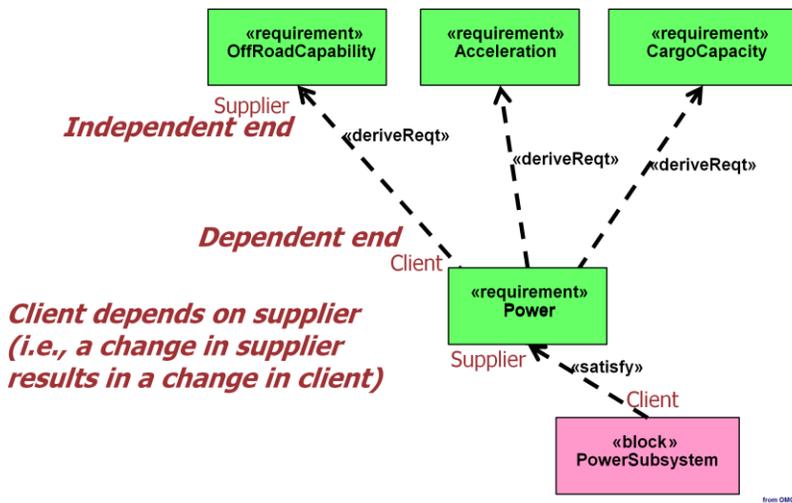


Requirement Relationships Model the Content of a Specification

64

© K-LASSE, Inc. 2009 UC San Diego Extension Online Learning
Course: Systems Engineering with OMG SysML™

Example of Derive/Satisfy Requirement Dependencies



Arrow Direction Opposite Typical Requirements Flow-Down

Select your models based on your audience



An IT Analyst might be most familiar with system context or data diagrams



A business stakeholder might prefer process flows

Elicitation sessions should be designed to minimize the time spent explaining model intent and notation and maximizing the discovery of stakeholder requirements.

Know your audience before the session starts. Find out what they are comfortable with and design your session to use that.

Incorporating models into requirements deliverables

- You can't and shouldn't avoid text
- You are likely to need a requirements management tool for the text
- Start with Org Charts, System Context and Process Flows
- Add models as needed to make sure all of the requirements are understood.

Resources

- These slides will be posted to the INCOSE SD web site
- Seilevel includes RML® Quick Reference
 - <http://www.seilevel.com/requirements-resources/>