

M&S Requirements Engineering

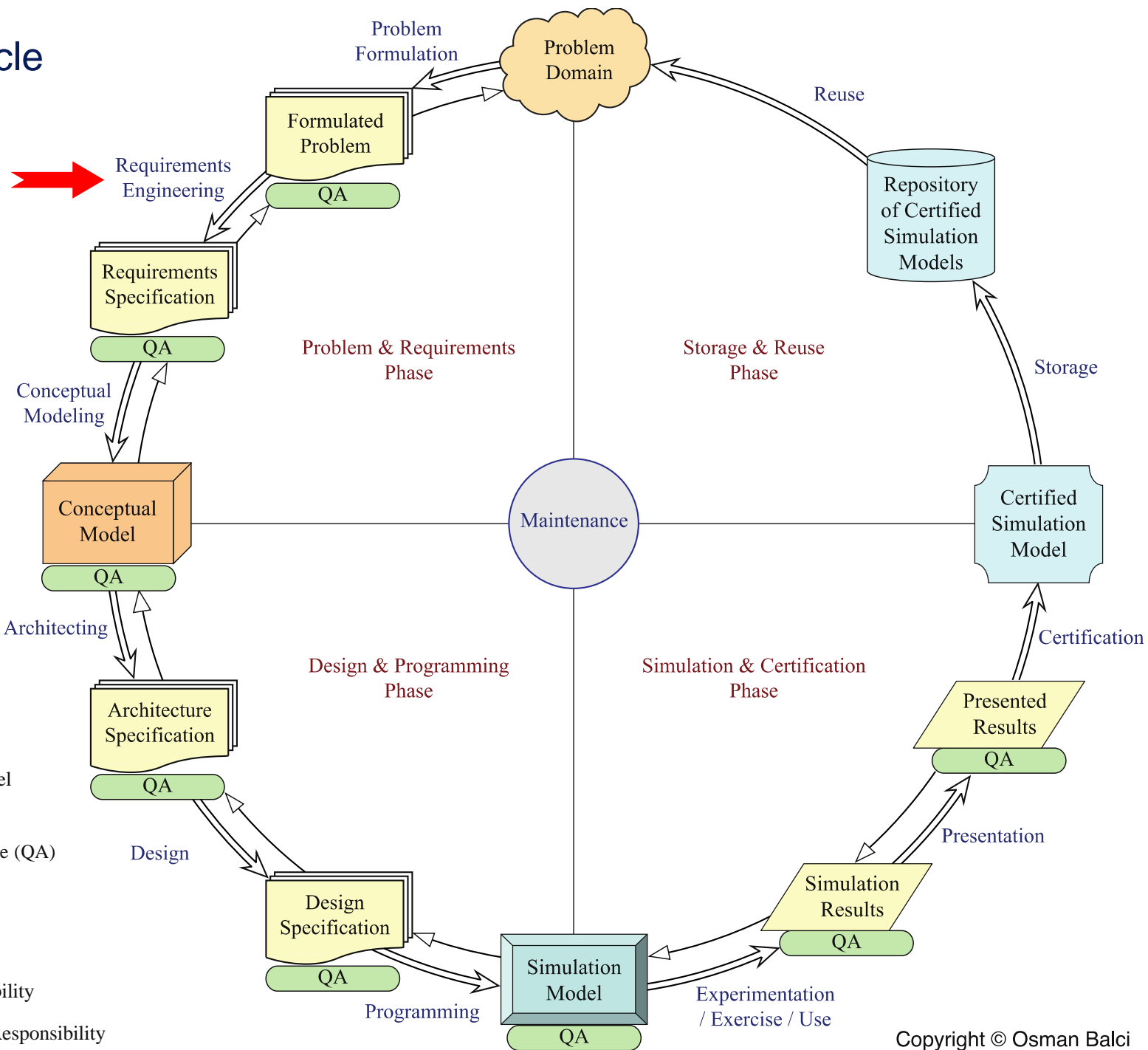
OSMAN BALCI

Professor

Department of Computer Science
Virginia Polytechnic Institute and State University (Virginia Tech)
Blacksburg, VA 24061, USA

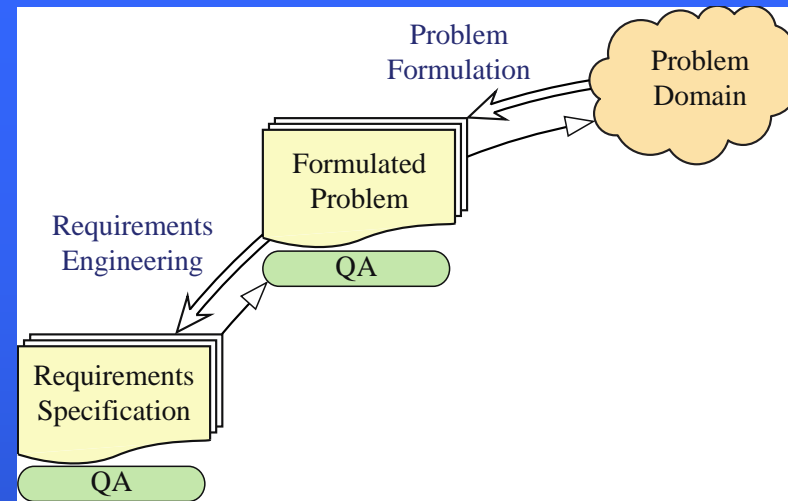
<https://manta.cs.vt.edu/balci>

M&S Life Cycle

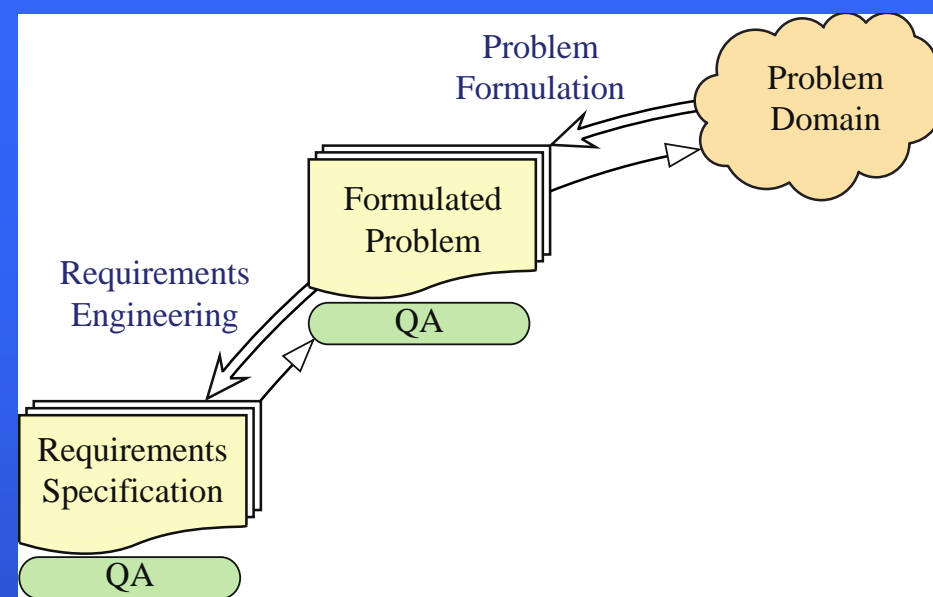


Requirements Engineering

- **Requirements Engineering** is the process of
 - a) elicitation of requirements based on the formulated problem and the problem domain, and
 - b) specification of the requirements in an authoritative manner.
- This process takes the **formulated problem** and the problem domain (**universe of discourse**) as input and produces an **M&S requirements specification document (RSD)** as the output work product.
- The M&S RSD becomes part of a legal contract between the M&S developer and the M&S sponsor.
- The M&S RSD must clearly dictate the **M&S Acceptability Criteria** with respect to which the delivered M&S will be judged.

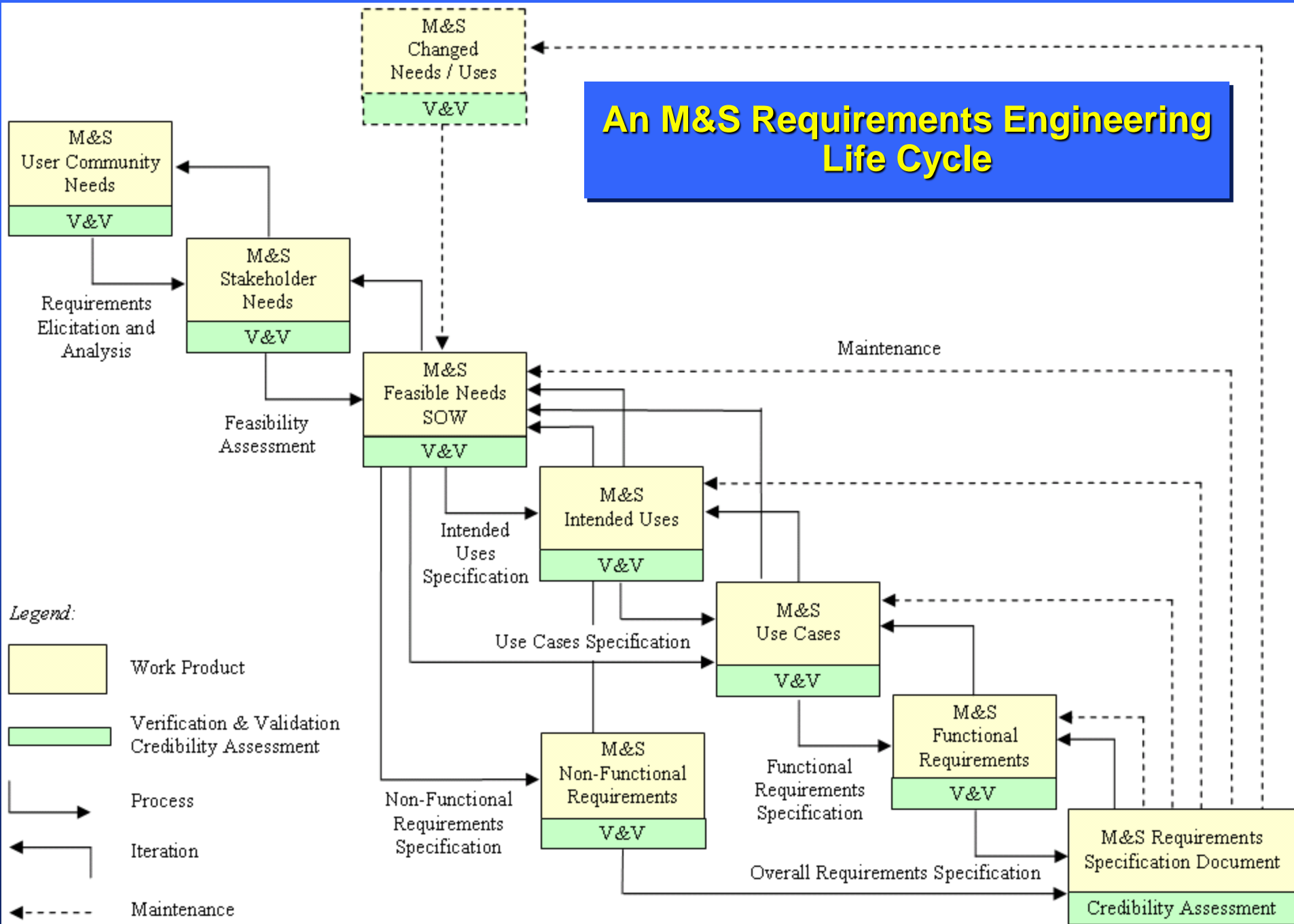


Requirements Engineering

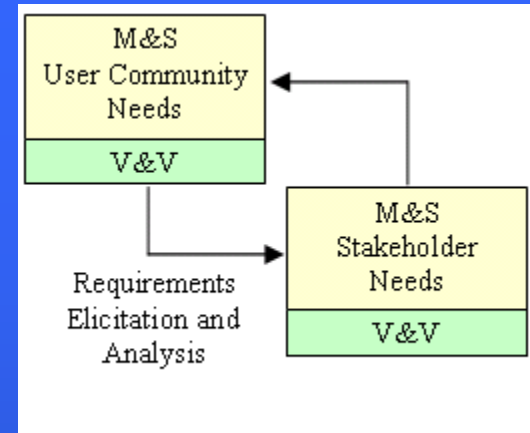


- **Requirements Engineering QA** integrates the assessments of
 - a) quality of the M&S RSD work **product**,
 - b) quality of the requirements engineering **process**,
 - c) quality of the **people** employed in requirements engineering, and
 - d) **project** characteristics related to the life cycle stage for requirements engineering.

An M&S Requirements Engineering Life Cycle



Example M&S User Community and Stakeholders

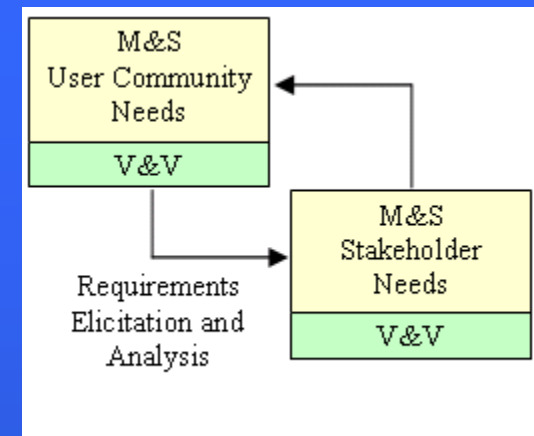


- **Topic:** Missile Signature Modeling and Simulation
- **M&S User Community:** **Missile Defense**
- **Example M&S Stakeholders:**
 - Air Force Research Laboratory (AFRL)
 - Army Research Laboratory (ARL)
 - Missile and Space Intelligence Center (MSIC)
 - Missile Defense Agency (MDA)
 - Naval Air Warfare Center (NAWC)
 - Naval Research Laboratory (NRL)
 - Naval Surface Warfare Center (NSWC)
 - Office of Naval Intelligence (ONI)
 - U. S. Army Aviation and Missile Command (AMCOM)

M&S Requirements Elicitation and Analysis

■ Example applications for simulation-based military training:

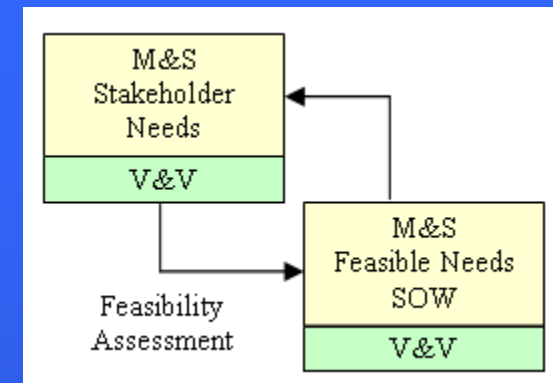
- Modular Semi-Automated Forces (**ModSAF**)
- One Semi-Automated Forces (**OneSAF**)
- Joint Semi-Automated Forces (**JSAF**)



- Customer does **not** know what the “real” requirements must be!
- Stakeholders do **not** know what they really want.
- Stakeholders express requirements in their own terms.
- Different stakeholders may have conflicting requirements.
- Organizational and political factors may influence the M&S requirements (especially for simulations for training).
- Requirements change during the M&S development life cycle.
- New stakeholders may emerge demanding changes.

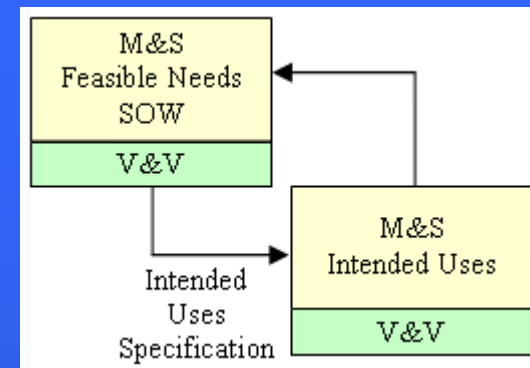
Feasibility Assessment

- Given the Formulated Problem, assess the feasibility of providing an M&S-based solution.
- Can the M&S-based solution be provided
 - Under a budget acceptable to upper management, sponsor or customer?
 - Within a desired period of time / schedule?
 - Using the current software / hardware technology?
 - In a way to possess acceptable Interoperability with existing (legacy) M&S applications?



Intended Uses Specification

- Simulation Model **Intended Use (IU)** refers to the explicitly and clearly defined purpose for which the simulation model is intended for use.
- **Intended Uses refer to the objectives under which the simulation model is created for use.**
- **The Intended Uses must be well defined!**
- **Intended Uses (Objectives) become the point of reference for the rest of the simulation model development life cycle.**
- **V&V and QA are carried out throughout the development life cycle with respect to the defined Intended Uses (Objectives).**



Intended Uses

Develop NMDMSA for the intended use of NMD system design simulation

by analogy



Develop a vehicle for the intended use of transportation

Develop NMDMSA for the intended use of DRR support

by analogy



Develop a vehicle for the intended use of land transportation

Develop NMDMSA for the intended use of providing DRR support for the assessment of:

1. Integrated engagement planning
2. Utilize surveillance data
3. Perform sensor operations
4. Perform engage
5. Perform hit assessment

by analogy



Develop a vehicle for the intended use of land transportation with:

- Air bags
- Four-wheel drive
- Integrated Cellular Antenna
- Power anti-lock brakes
- Theft Deterrent System

How to run the model?

What experiments to perform under what input data and conditions to obtain the output data required for a particular DRR assessment?

by analogy



How to operate the vehicle?

What to transport?
Transport from where to where?
Transport under what conditions?

-
- NMD = National Missile Defense
NMDMSA = National Missile Defense M&S Application
DRR = Deployment Readiness Review

Hierarchy of Decomposition of the Intended Uses

- At the highest level, the NMD M&S Application purpose is decomposed into:
 1. NMD system performance assessment
 2. NMD ground and flight test prediction, planning, and design
 3. NMD system integration support
 4. NMD DRR support
 5. Operational Test Agency (OTA) analysis
- We call the sub-purposes at the highest level as the NMD M&S Application **Domains of Applicability**.
- Each domain of applicability should be decomposed further.

Decomposition of NMD M&S Domains of Applicability into Intended Uses

DRR System Function <i>NMDMSA Sub-Domain of Applicability</i>	NMD System Technical Performance Measure	NMDMSA Intended Use
1. System Operations Activation		
2. Readiness Operations		
3. System Status	• Health and Status Reporting Time	
	• Critical Data & Display Accuracy with NMD & ITW/AA	
	• Display Concurrency of NMD / BMC2	
4. Collateral Mission		
5. Control of Defense	• System Activation Time	
	• Situational Awareness	X
	• DEA Granting Time	X
	• Management by Exception	X
6. Integrated Engagement Planning	• Probability of Engagement Planning	X
	• Engagement Planning Time	X
7. Surveillance Data	• Probability of Warning	X
	• Reported Position Accuracy	X
	• Reported Velocity Accuracy	X
8. Sensor Operations	• XBR Probability of Acquisition	X
	• UEWR Probability of Acquisition	X
	• XBR Probability of Sensor Track Reporting	X
	• UEWR Probability of Sensor Track Reporting	X
	• XBR Position Track Accuracy	X
	• UEWR Position Track Accuracy	X
	• XBR Velocity Track Accuracy	X
	• UEWR Velocity Track Accuracy	X
	• XBR Probability of Sensor Discrimination	X
	• UEWR Probability of Sensor Discrimination	X
9. Engagement	• Probability of Kill Single Shot – P_{KSS}	X
10. Hit / Kill Assessment	• Probability of Hit – P_{HT}	X
11. Launch Essential Maintenance		
Overall System Wrap-Up	• Probability of Integrated System Effectiveness – P_{ISE}	X
	• Engagement Timing Margin	X
	• Graceful Degradation	
	• Integration of Operator Interfaces	
	• NMD and ITW/AA Defended Area	

Simulation Model Intended Use Documentation Template

BEST Model Intended Use	
Intended Use ID:	<e.g., IU001>
Intended Use Creator:	<name, organization, e-mail>
Intended Use Name:	<e.g., Correlated EO and RF Signatures>
Intended Use Category:	<i>Example:</i> <ul style="list-style-type: none"> Analysis, Evaluation, Comparison, Sensitivity Analysis, Prediction, Ranking and Selection, etc.
Intended Use Brief Description:	
<p>This section provides the highest-level description of the IU from the perspective of the user. The brief description should be similar to the specification of an objective in one or few sentences.</p> <p><i>Example:</i></p> <ul style="list-style-type: none"> Develop correlated, passive EO imagery and active (monostatic) RF signatures for missile hardbodies from launch through reentry. 	
Intended Use Overview:	
<p>This section provides a detailed overview of the IU from the perspective of the user. The overview should be similar to the detailed specification of an objective and must be accurate, complete, clear, and unambiguous.</p>	
Model Input Description:	
<p>This section provides a technical description of the model input, which may include scenarios, input conditions, parameter values, random variable characterizations, files, and databases.</p> <p><i>Examples:</i></p> <ul style="list-style-type: none"> Sensor Position Sensor Line of Sight Radiance Mode Initial Body Orientation Angles (α, β, γ) Initial Angular Rotation Rates ($\omega_x, \omega_y, \omega_z$) Solar Perturbation Options Aerodynamics Coefficient File Name Aerocoeficient Data Bases 	

Simulation Model Intended Use Documentation Template (continued)

Model Description:

This section describes the components that make up the model. The components represent problem domain elements. The description is intended to communicate to the reader what exactly the model represents.

Example model components:

- Environment
- Observer
- Trajectory
- Phenomenology Components
 - a. Foreground
 - i. Aurora
 - ii. Missile Plume
 - iii. Cloud Robust
 - b. Background
 - i. Earthlimb Fast
 - ii. Terrain Fast
 - iii. Space
 - c. Laser Foreground
 - i. Laser Hardbody
 - d. Non Imaging
 - i. NucPemRF
 - ii. NucHSRF
 - iii. Radar Hardbody

Model Experiment Description:

This section describes the designs of desired experiments to be conducted with the model.

Examples:

- An experiment designed to perform sensitivity analysis to determine the effect of changes in the “sensor position” input parameter value on the model’s output “radiance images”.
- An experiment designed to establish relationships among a set of model components such as observer, trajectory, and missile plume.
- An experiment designed to perform comparative evaluation of a set of alternatives such as different strategies, algorithms, physics models, or techniques.
- An experiment designed to measure target temperature history in the presence of solar noise.

Simulation Model Intended Use Documentation Template (continued)

Model Output Description:

This section provides a technical description of the expected model output.

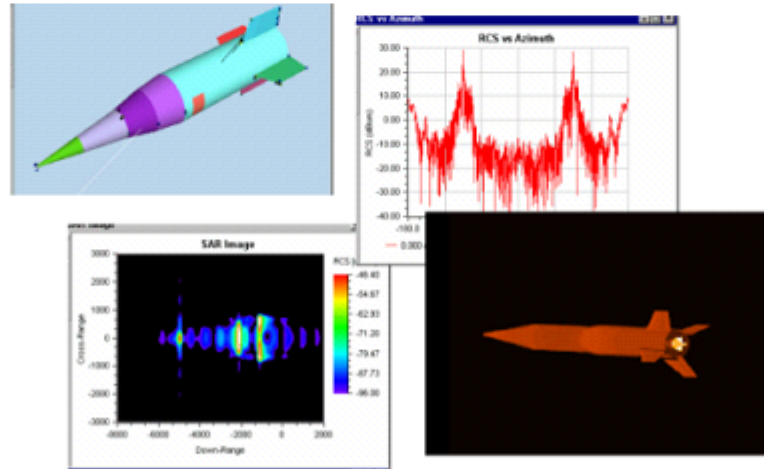
Examples:

- Radiance images
- Target temperature history
- Target orientation history
- Transmission data file
- Radar cross section file
- Diagnostic output file(s)
- Component output databases

Presentation of Model Results:

This section describes the expected forms of presenting the model results. The presentation can be in a variety of forms including tabulated data, statistical confidence intervals, various forms of graphs, 2D or 3D graphics, and other visual representations.

Examples:



Acceptability Criteria:

This section lists the criteria by which the model acceptability will be judged with respect to the documented intended use.

Example Simulation Model Intended Use Documentation

BEST Model Intended Use	
Intended Use ID:	IU002
Intended Use Creator:	
Intended Use Name:	Evaluate discrimination algorithm performance
Intended Use Category:	Ranking and Selection
Intended Use Brief Description:	
<p>Generate correlated IR/RF time-varying signatures to external sensor models to evaluate approaches to object typing (discrimination). Assumes:</p> <ol style="list-style-type: none">1. User has sensor model to apply BEST data to2. User has Signal/Data processing algorithms representing discrimination algorithms/decisions.3. This is not a TOM correlation or data association task, but each object is assumed to be cleanly resolved (both IR and RF). Therefore the application is best associated with post-deployment phase after some object separation has occurred.	
Intended Use Overview:	
<p>BEST will be used to provide time-varying signature data in the EO and RF bands from user-specified midcourse (ballistic) objects, including RVs, replicas, balloons, and debris.</p> <p>The EO signature data will be in-band intensity versus time over the full flight path as seen from the prospective sensor position. This will be done for 2 satellite sensors viewing the same complex. Each satellite has similar sensors (wavebands and data collection times). Backgrounds will not be calculated as they will be assumed to have been removed by image processing schemes (residual noise effects can be included in the sensor noise model).</p> <p>The RF signature data will consist of I/Q versus time pairs for a waveform and PRF appropriate for the discrimination function. It will be generated for an assumed ground-based radar location. The user may elect to run BEST in Monte Carlo mode to generate large numbers of realizations of target objects.</p> <p>The IR and RF signature data will be correlated (i.e. from the same basic physics) so that joint discrimination decision logic can be fairly evaluated.</p>	

Example Simulation Model Intended Use Documentation (continued)

Model Input Description:

- Trajectories or state vectors (post-burnout) for desired objects
- Physical characteristics (geometry, material, thermal) of objects
- Location (fixed, trajectory, or state vector) and properties of sensors
- Sensor data collection properties
- Environmental options (cloud cover, atmospheric properties)

Model Description:

- Correlated IR/RF signatures for each threat object.
- Appropriate upwelling radiance upon target (Earthshine)
- 6-DOF kinematics of objects
- Stars/zodiacal not required
- Atmospheric transmission not required (all assumed exoatmospheric)

Model Experiment Description:

Representative threat packages will be used to evaluate the effectiveness of discrimination algorithms for GMD and space-based sensors. Radar cross section data as a function of time will be generated and input into high-fidelity sensor models that execute candidate discrimination algorithms using various parameters.

Model Output Description:

- Radar cross section (I/Q) data as expected at the RF sensor location and appropriate for the radar operating in discrimination mode. (RF data will need to be post-processed to generate circular-polarized signatures)
- EO signature strings (time histories) of each target as seen by each of the two orbiting platform sensors.

Presentation of Model Results:

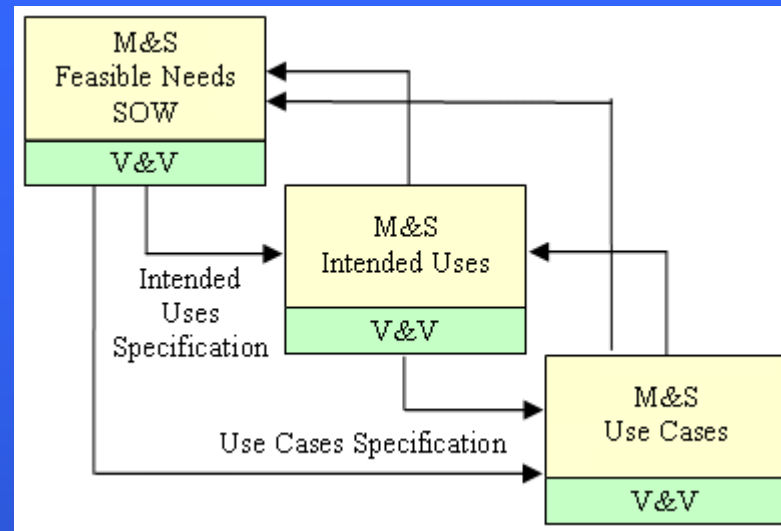
Text or binary files as required by external sensor models.

Acceptability Criteria:

- Object kinematics is accurate.
- Generated signatures are accurate (within acceptability limits for the discrimination algorithm used).

Use Cases Specification

- M&S Use Cases are created under the guidance of the **M&S Feasible Needs** specification based on the well-defined **M&S Intended Uses**.



- A **Use Case** specifies the behavior of a simulation model or a part of a model and is a description of a set of sequences of actions, including variants, that a simulation model performs to yield an observable result of value to an actor.
 - Examples:

Design Experiment

Validate User

Generate
Random Number

- A use case carries out some tangible amount of work.

Actors

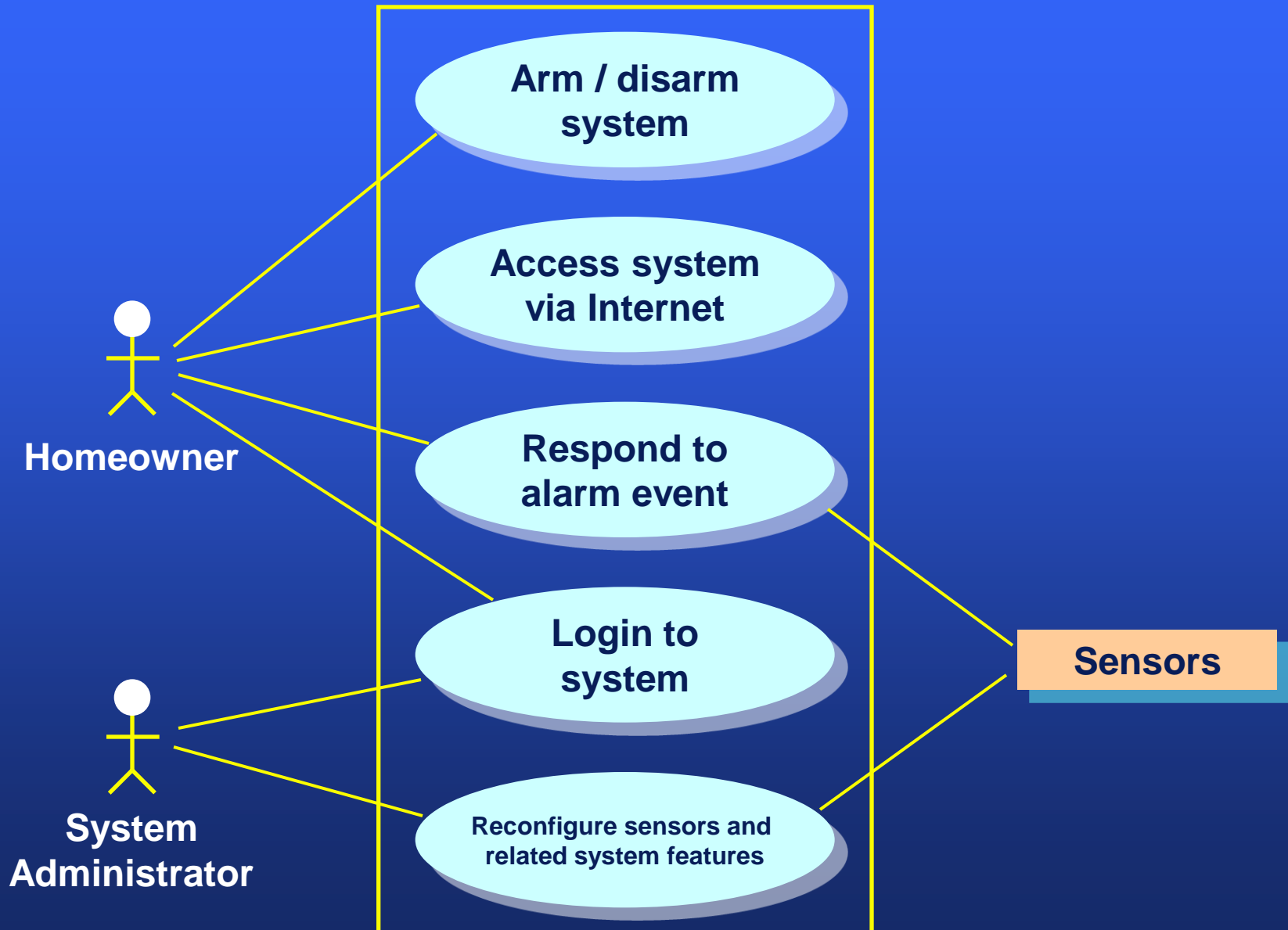
- An **Actor** represents a coherent set of roles that users of use cases play when interacting with those use cases.

- **Examples:**

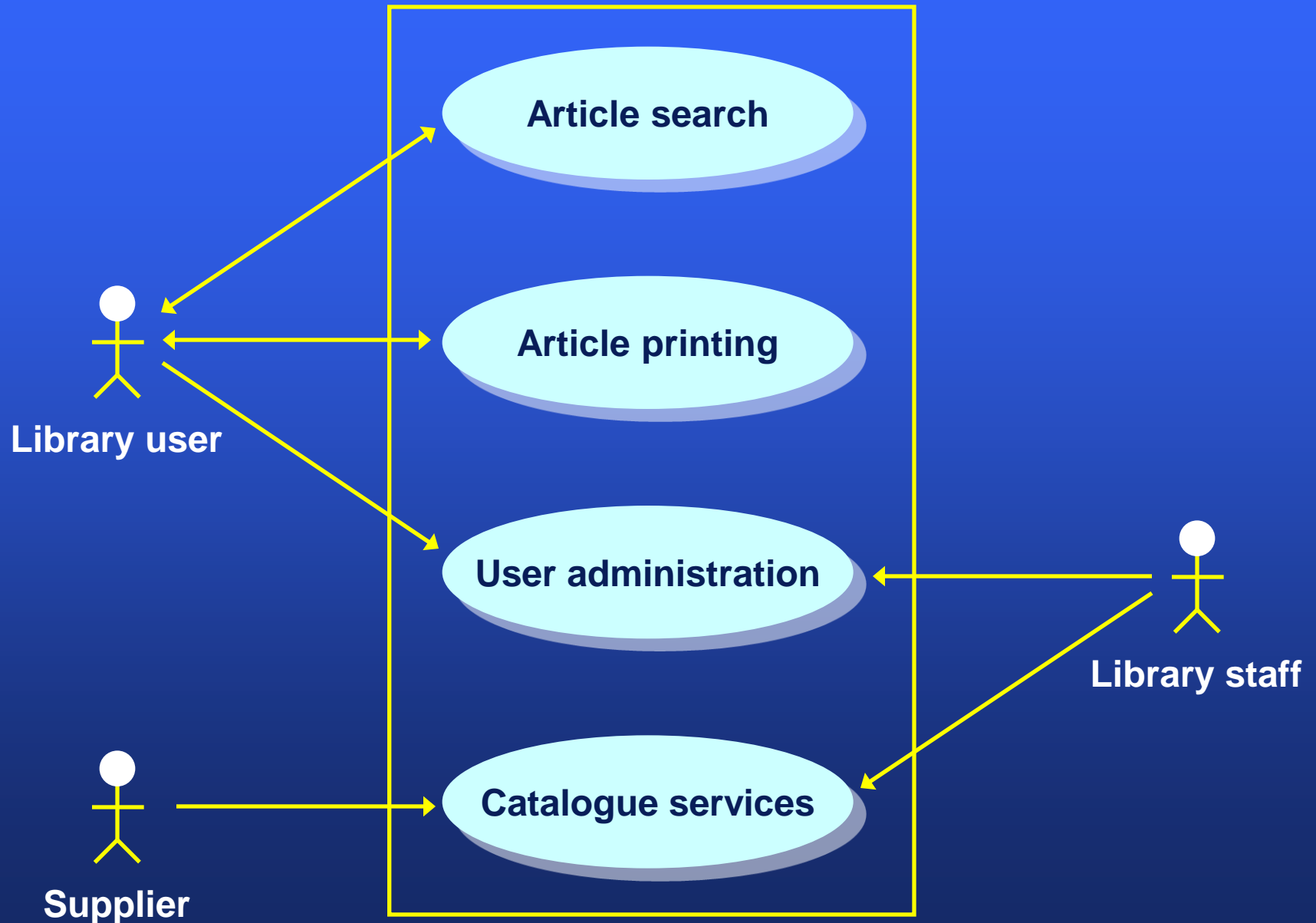


- For example, a user can be a **Simulation Experiment Designer** as well as an **Administrator User** for a simulation system.
- An **Actor** can be
 - A user (human)
 - A model component
 - A submodel
 - A submodel of another submodel, or
 - Another model as a whole for a distributed simulation system.
- **Actors** may be connected to use cases only by associations.

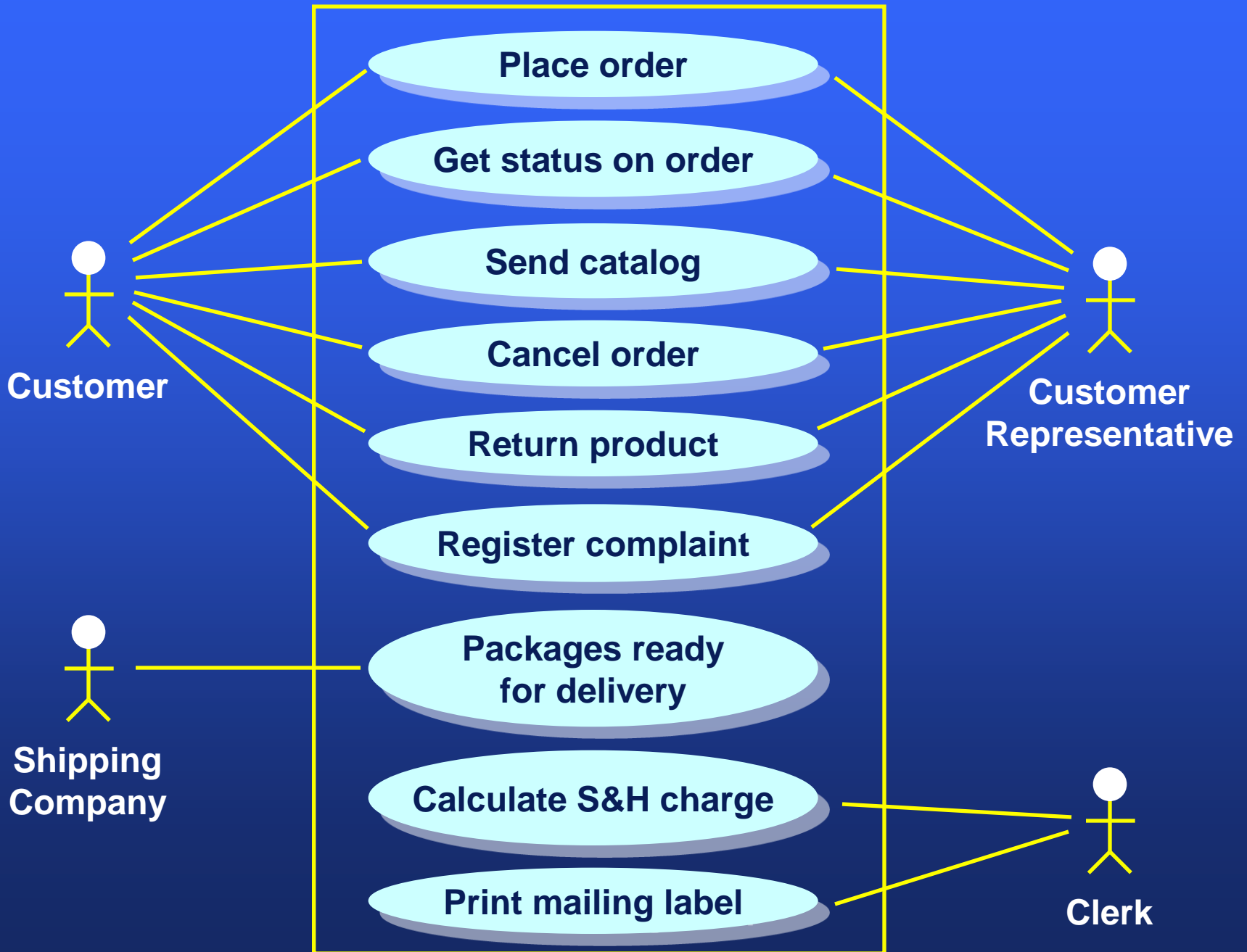
A Use-Case Diagram for a Home Security System



A Use-Case Diagram for a Library System



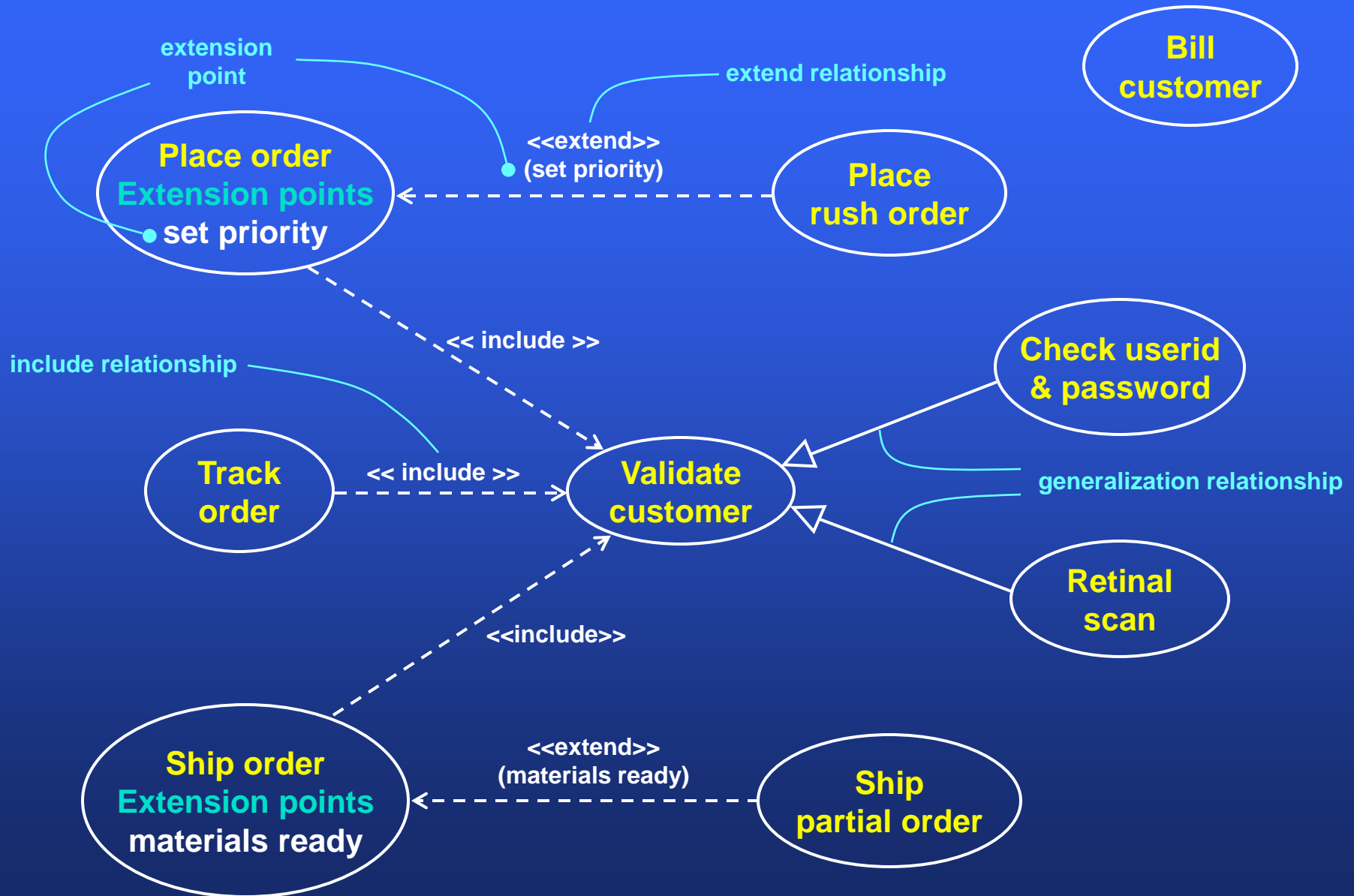
A Use-Case Diagram for an E-Commerce System



Organizing Use Cases

- Use cases can be organized by specifying
 - **Generalization Relationship:**
 - ❖ It is similar to generalization of classes.
 - ❖ A child use case inherits the behavior and meaning of the parent use case.
 - ❖ The child may add to or override the behavior of its parent.
 - **Include Relationship:**
 - ❖ A base use case explicitly incorporates the behavior of another use case at a location specified in the base.
 - **Extend Relationship:**
 - ❖ A base use case implicitly incorporates the behavior of another use case at a location specified indirectly by the extending use case.

Dependency Relationships Among Use Cases



A Use Case Description Template

■ Use Case Name

- A brief description. Usually a paragraph or less.

■ Actors

- A list of the actors who communicate with this use case.

■ Preconditions

- A list of conditions that must be true before the use case starts. Precondition describes the state the system must be in at the start of the entire use case.

■ Flow of Events of the Primary Scenario

- Most probable flow of events defining the main or primary scenario.

■ Flows of Events of the Secondary Scenarios

- **Alternative scenarios:**

An **Alternative Scenario** is one that allows a different sequence of events than what was described for the primary scenario.

- **Exception scenarios:**

An **Exception Scenario** is one where we handle an error, an interrupt, or an exception.

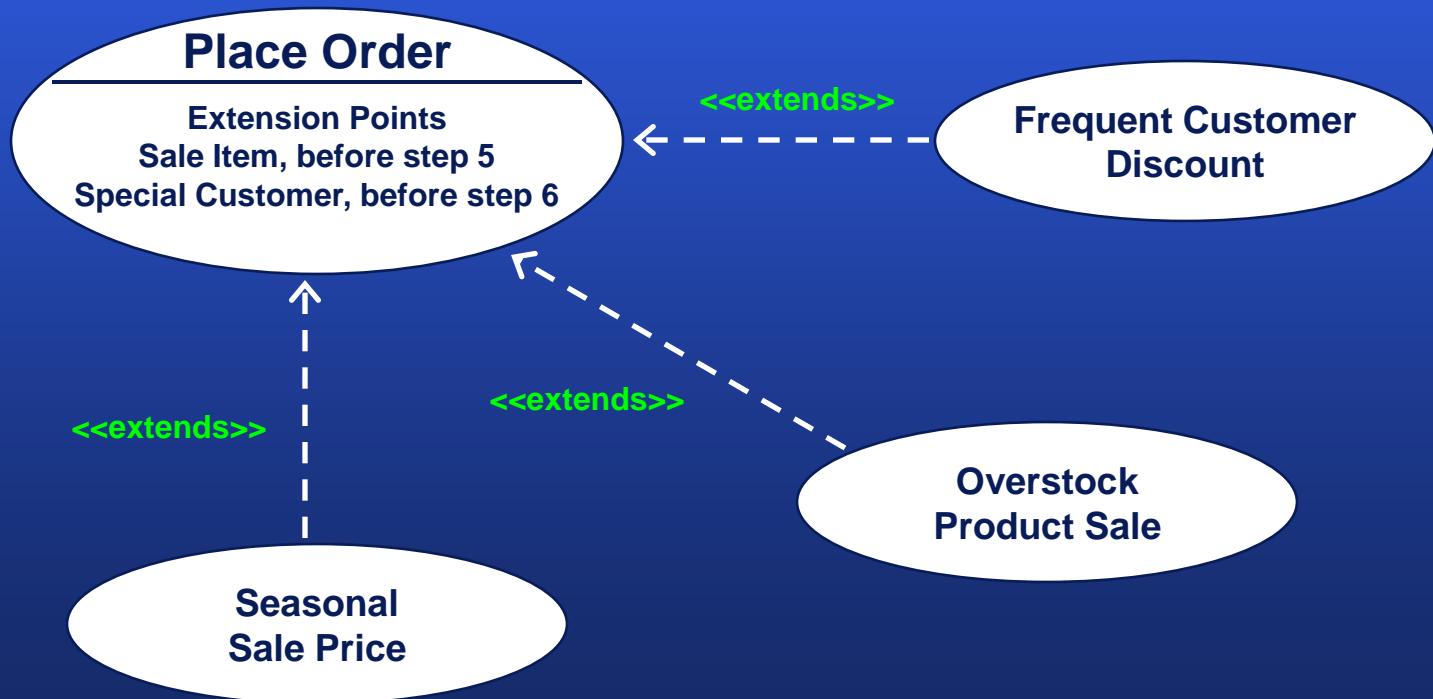
Scenarios

- A **Scenario** is a specific sequence of actions that illustrates behavior.
- Scenarios are to use cases as instances are to classes, meaning that **a scenario is basically one instance of a use case.**
- Example: the use case **Hire Employee**
 - Scenario 1: hire a person from another company
 - Scenario 2: hire a person by transferring from one division to another
 - Scenario 3: hire a person who is a foreign national
 - Scenario 4: hire a person who just graduated from college
- Each of the above variants can be expressed in a different sequence of actions, called Scenario.

A Use Case Description Template (continued)

■ Extension Points

- If the use case has extension points, list them here.
- **Extend** specifies that the target use case extends the behavior of the source.



A Use Case Description Template (continued)

■ “Used” Use Cases

- If the use case uses other use cases, list them here.

■ Postconditions

- A list of conditions that must be true when the use case ends, regardless of whether the primary scenario or an alternative scenario is executed.
- Execution of an exception scenario indicates errors that must be corrected so that either the primary scenario or an alternative scenario execution can be completed.
- Postcondition describes the state the simulation model must be in at the end of the use case.

Use Case Documentation Template

Use Case Documentation Template	
Use Case ID:	<e.g., UC001>
Use Case Name:	<A brief description; usually a paragraph or less.>
Actors:	
<A list of the actors who communicate with this use case.>	
Preconditions:	
<A list of conditions that must be true before the use case starts. Precondition describes the state the system must be in at the start of the entire use case.>	
Flow of Events of the Primary Scenario:	
<Most probable flow of events defining the main or primary scenario.>	
Flow of Events of the Alternative Scenarios:	
<An Alternative Scenario is one that allows a different sequence of events than what was described for the primary scenario.>	
Flow of Events of the Exception Scenarios:	
<An Exception Scenario is one where an error, an interrupt, or an exception is handled.>	
Extension Points:	
<If the use case has extension points, list them here. Extends are used when you have an optional sequence of events you want to include in a use case.>	
“Used” Use Cases:	
<If the use case uses/includes other use cases, list them here.>	
Postconditions:	
< A list of conditions that must be true when the use case ends, regardless of whether the primary scenario or an alternative scenario is executed. Execution of an exception scenario indicates errors that must be corrected so that either the primary scenario or an alternative scenario execution can be completed. Postcondition describes the state the system must be in at the end of the use case.>	

Example: “Place Order” Use Case

■ Use Case Name: **Place Order**

- This use case describes the process by which a user selects a number of products, enters shipping and billing addresses, and provides payment information for placing an order.

■ **Actors:** Customer, Phone Order Clerk, Mail Order Clerk

■ **Precondition:** A valid user (i.e., one of the actors) has logged into the system.

■ **Primary Scenario**

1. The use case starts when the user selects Place Order.
2. The user enters the name, contact information, shipping address, and billing address.
3. If the user enters only the zip code, the system supplies the city and state.

Example: “Place Order” Use Case (continued)

4. **While** the user enters product codes
 - a. The system supplies a product description and price
 - b. The system adds the price to the total

End

5. The user enters credit card payment information.
6. The user selects Submit.
7. The system verifies the information, saves the order as pending, and forwards payment information to the accounting system.
8. When payment is confirmed, the order is marked as confirmed, an order ID is returned to the user, and the use case ends.

Example: “Place Order” Use Case (continued)

■ Secondary Scenarios

- **Alternative Scenarios:**
 - ❖ User sends order by mail
 - ❖ User phones in order
- **Exception Scenarios:**
 - ❖ User selects Cancel at any time before selecting Submit
 - ❖ Shipping and billing addresses are different
 - ❖ Invalid credit card number
 - ❖ Network connection is lost while ordering
 - ❖ User can't login (e.g., precondition is unsatisfied)
 - ❖ Shipping address is incomplete
 - ❖ Product is out of stock

Example: "Place Order" Use Case (continued)

■ Extension Points

- Frequent Customer Discount
- Overstock Product Sale
- Seasonal Sale Price

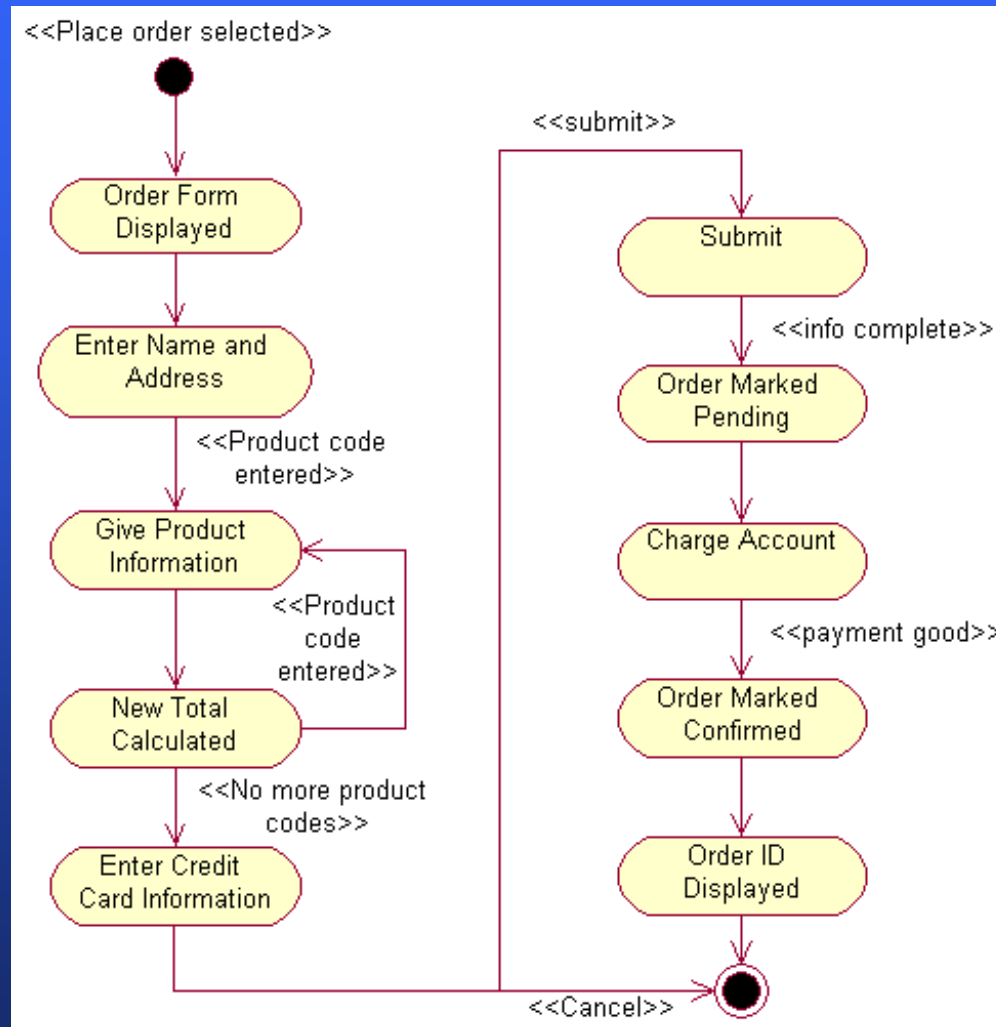
■ "Used" Use Cases

- Give Product Information
- Update Account

■ Postconditions

- The order is saved in the system and marked as confirmed.

“Place Order” Use Case Activity Diagram



Example: "Login" Use Case

■ Use Case Name: Login

- This use case describes the process by which users log into the order-processing system. It also sets up access permissions for various categories of users.

■ Actors

- User of any type such as Customer, Phone Order Clerk, Mail Order Clerk, and Administrator.

■ Precondition

- Login screen is displayed.

■ Primary Scenario

1. The use case starts when the user gets the Login Screen.
2. The user enters a username and password.
3. The system verifies the information.

Example: "Login" Use Case (continued)

4. The system sets access permissions.

5. The system displays the Main screen.

6. The user selects a function.

7. **While** the user does not select Exit **Loop**

If the user selects Place Order **then** use Place Order

else if the user selects Return Product **then** use Return Product

else if the user selects Cancel Order **then** use Cancel Order

else if the user selects Review Order **then** use Get Status on Order

else if the user selects Send Catalog **then** use Send Catalog

Example: “Login” Use Case (continued)

else if the user selects Contact Customer Service **then** use Register Complaint

else if the user selects Run Sales Report **and** user is a Customer Rep **then** use Run Sales Report

The user selects a function.

8. **End Loop**

9. The use case ends.

Example: “Login” Use Case (continued)

■ Secondary Scenarios

- Alternative Scenarios:

- ❖ None

- Exception Scenarios:

- ❖ Bad username

- ❖ Bad password

- ❖ User does not have a username and password for our system

- ❖ Network connection is lost while logging on

- ❖ User selects a function for which s/he is not allowed access

Example: “Login” Use Case (continued)

■ Extension Points

- None

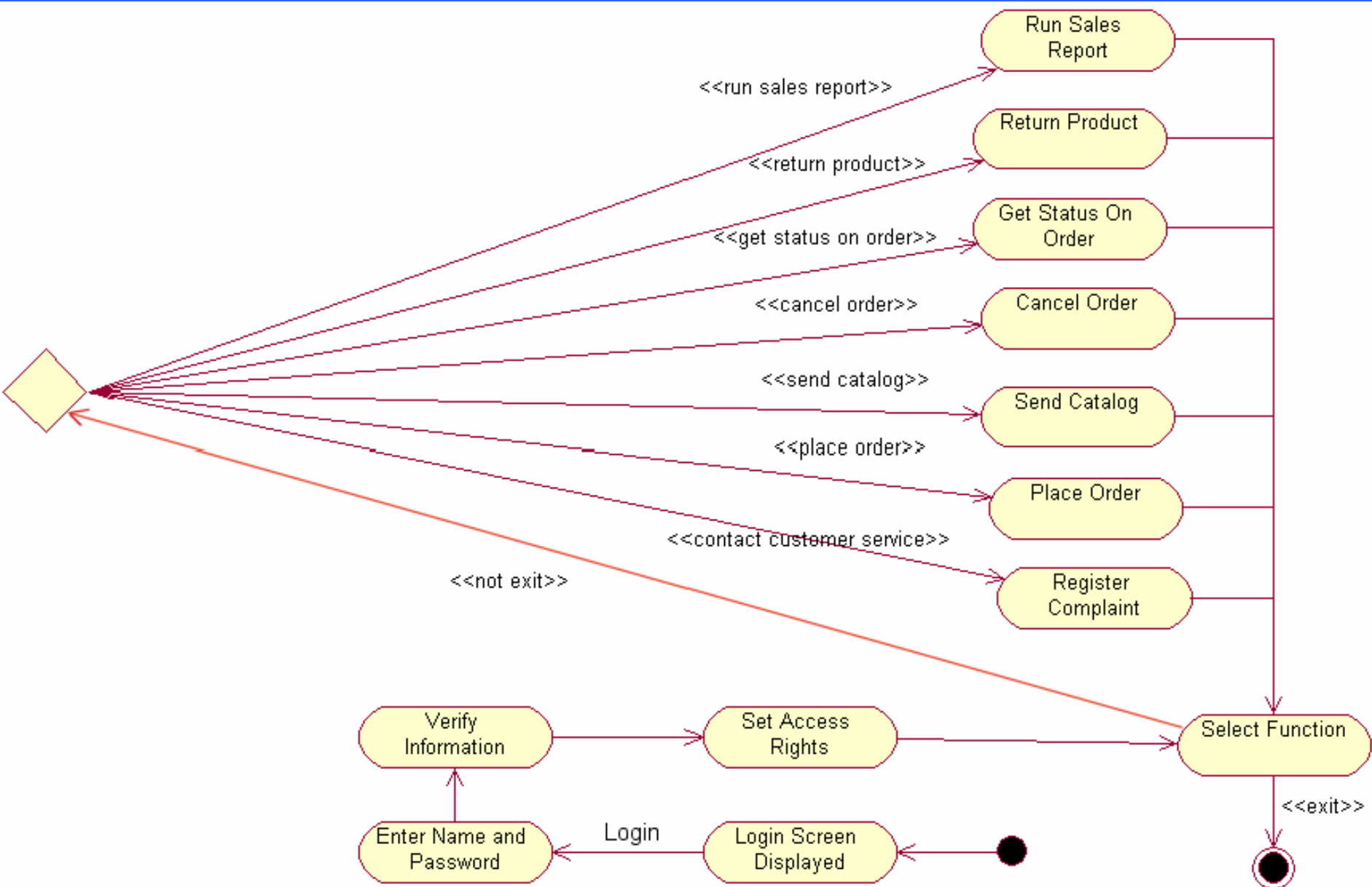
■ “Used” Use Cases

- Cancel Order
- Get Status on Order
- Place Order
- Register Complaint
- Return Product
- Run Sales Report
- Send Catalog

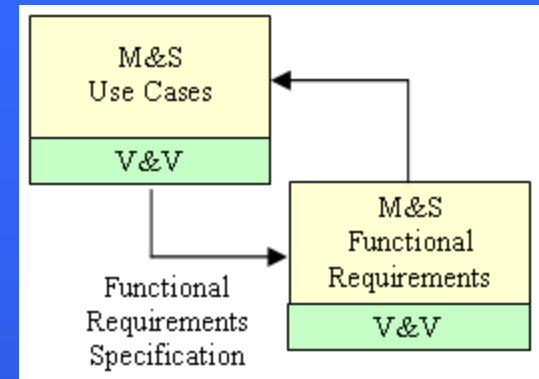
■ Postconditions

- A function or Exit is selected on the Main Screen.

“Login” Use Case Activity Diagram



Requirements Identification



- Given a **Use Case**, identify the functional and non-functional requirements associated with that Use Case.
- Requirements are classified into two major categories:
 - **Functional Requirements**
 - **Non-Functional Requirements**
- **Use Case-based Requirements Engineering** is considered the Best Practice!

Functional and Non-functional Requirements

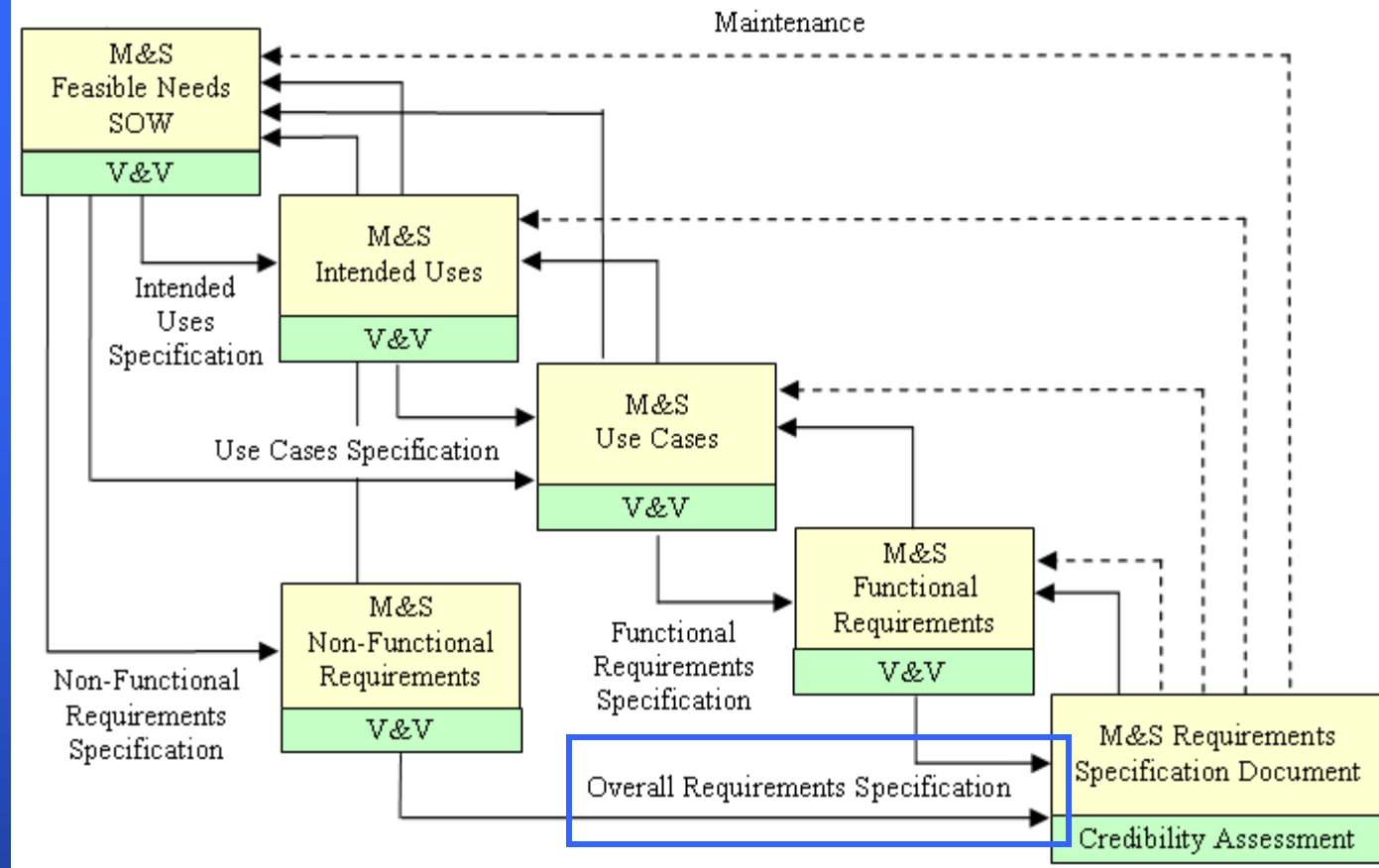
■ Functional requirements

- Statements of services the model should provide, how the model should react to particular inputs and how the model should behave in particular situations.
- Describe functionality or model services
- Functional requirements are requirements about the input-output transformations of the simulation model

■ Non-functional requirements

- Are requirements that are unrelated to functionality of the simulation model
- **Examples:** Requirements for
 - ❖ Interoperability, Performance, Usability, Standards, Delivery, Portability

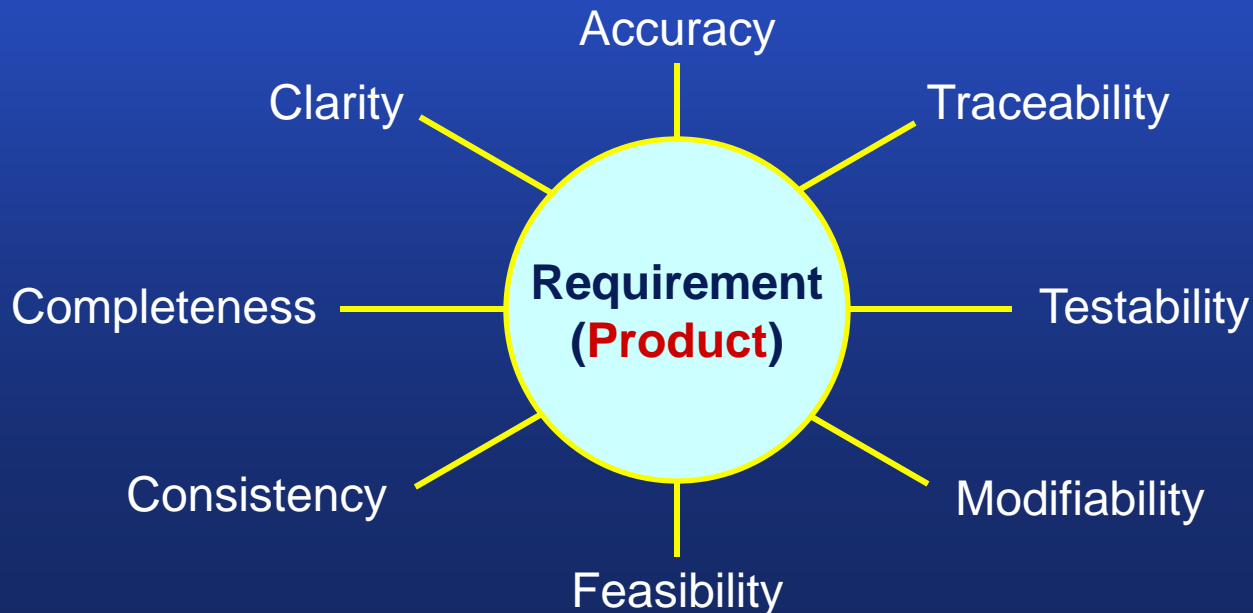
Overall Requirements Specification



- In the process of **Overall Requirements Specification**, we technically establish each requirement, identified in the previous processes, in a written form.
- By executing this process, we produce the **M&S Requirements Specification Document (RSD)**, which is often part of the legal contract signed between the customer and M&S developer.

Specification of a Requirement

- A requirement is specified using “shall”
 - **Example:** “A simulation model user **shall** be able to specify the number of simulation runs to perform.”
- A **requirement** should be specified using **active (direct) voice** as opposed to using **passive (indirect) voice**.
- A **requirement** is engineered as a **product** with required quality characteristics.



Example Bad Requirements

- “The user of the simulation model shall be authenticated with a username and password.”
- “The simulation model shall be very easy to use.”
- “All simulation submodels shall exchange data with each other.”
- “The training simulation system shall provide appropriate viewers for the user to read tutorial documents.”
- “Users shall be able to use the training simulation over the Internet.”

Example Functional Requirements

1. “The simulation model shall authenticate the user who shall enter a username and password.
 - a. Username shall consist of minimum 6 and maximum 12 alphanumeric characters.
 - b. Password shall consist of minimum 8 and maximum 24 characters of which
 - i. at least 2 must be uppercase letters,
 - ii. at least 2 must be lowercase letters,
 - iii. at least 2 must be numbers,
 - iv. at least 2 must be special characters selected from !, @, #, \$, %, &, *, +, =, and
 - v. no space is used.
 - c. The user’s account shall be locked upon 5 consecutive authentication failures.”

Example Functional Requirements

- “The simulation model shall enable the user to replicate simulation runs using the ‘method of batch means’ technique.”
- “The user shall be able to play, pause, and stop the visualization at any time during the course of simulation execution.”
- “Given the number of kill vehicles and their locations as input, the simulation model shall estimate the probability of killing an enemy missile as a 95% statistical confidence interval.”
- “Given the interarrival time probability distribution, arrival rate, and service rate, the simulation model shall estimate the average waiting time in the system.”

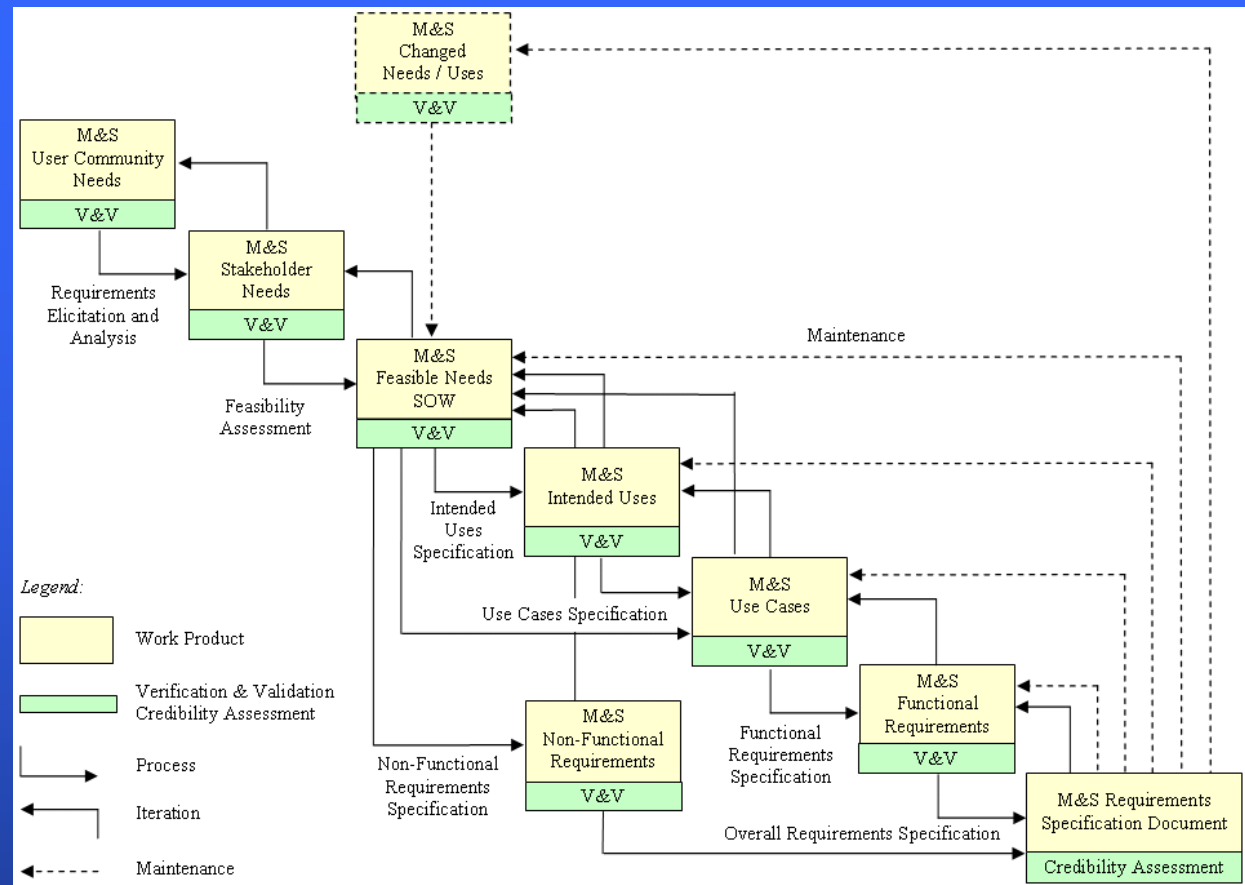
Example Non-Functional Requirements

- **“Interoperability among the distributed simulation models shall be accomplished using the High Level Architecture standard.”**
- **“The simulation model shall be developed using royalty-free software products.”**
- **“The web-based simulation system shall be usable with most recent versions of browsers such as Chrome, Firefox, Internet Explorer, and Safari.”**
- **“The weather forecasting simulation model experimentation shall be completed in no more than 6 hours.”**
- **“The simulation model shall be executable under Microsoft Windows Server 2008 operating system.”**
- **“All simulation documentation shall be provided in HTML-based hypertext format.”**
- **“All simulation input data shall be specified in XML format.”**

Why is Use Case-based Requirements Engineering the Best Practice?

- A Use Case represents a small amount of work the simulation model is required to perform. Thus, decomposing a complex simulation model functionality into **Use Cases enables the modularization needed to overcome the complexity.**
- Identifying the “real” functional requirements is always challenging for complex simulations, especially for those used for training purposes. A Use Case describes an interaction, and based on that description, **“real” functional requirements can be more successfully identified** and associated with that Use Case.
- Listing requirements one after the other, even in different categories, does not provide any help for transitioning from requirements to simulation model design. **Use Cases turn themselves into classes in an object-oriented design and significantly facilitate the transition into the model design phase.**
- Associating functional requirements with a **Use Case enables better life-cycle traceability:**
requirement ↔ use case ↔ class in design ↔ class in code.

Requirements Management



- A software product such as
 - IBM Dynamic Object Oriented Requirements System (DOORS) or
 - IBM Rational RequisiteProcan be used for requirements management.

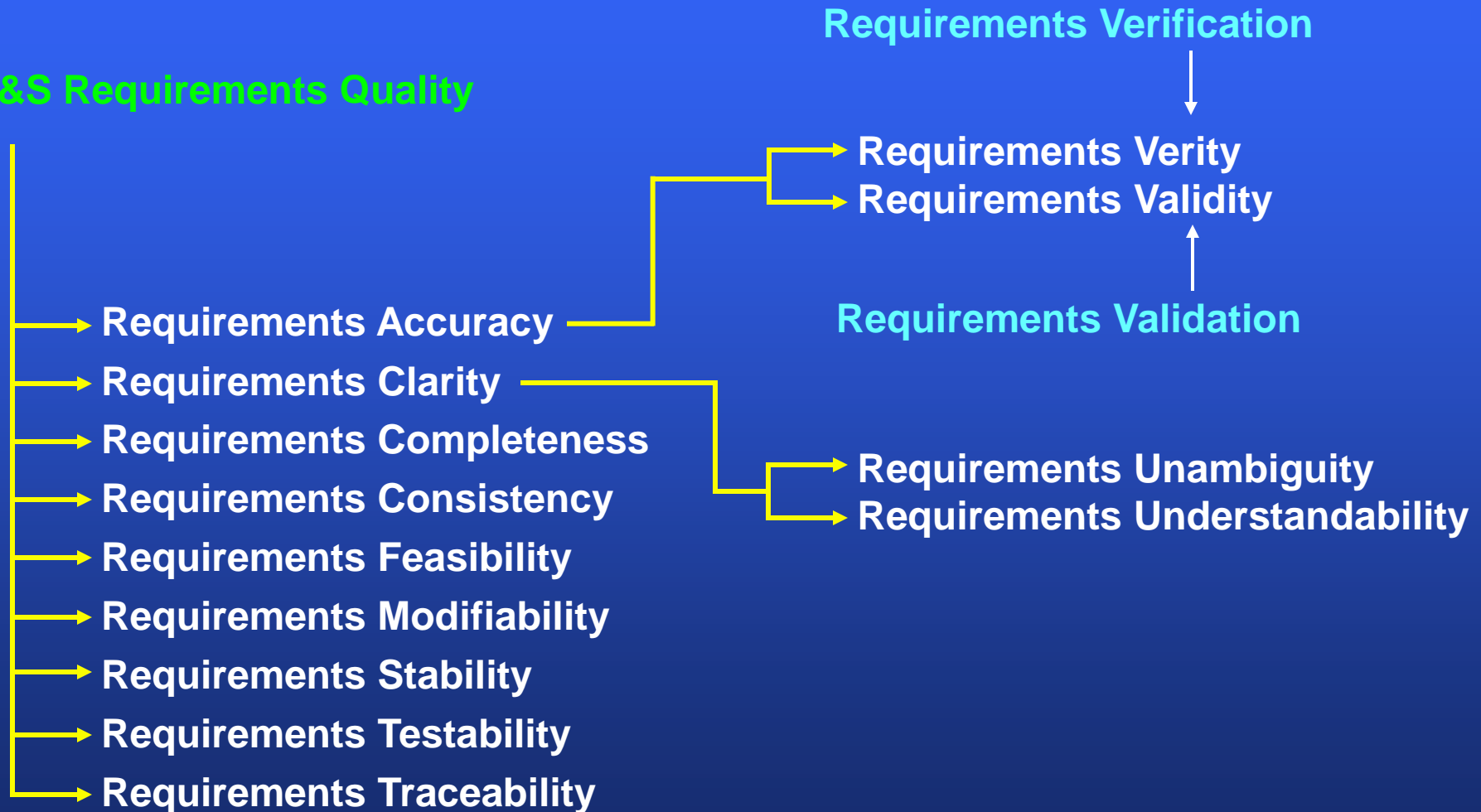
■ M&S Requirements Engineering Quality Assurance (QA)

integrates the assessments of

- quality of the M&S Requirements Specification Document (RSD) work **product**,
- quality of the requirements engineering **process**,
- quality of the **people** employed in requirements engineering, and
- **project** characteristics related to the life cycle stage for requirements engineering.

M&S Requirements Quality Assessment

M&S Requirements Quality



M&S Requirements Accuracy

is the degree to which the requirements possess sufficient transformational and representational correctness.

- **M&S Requirements Verity** is assessed by conducting M&S requirements verification.
- **M&S Requirements Verification**
 - is substantiating that the M&S requirements are transformed from higher levels of abstraction into their current form with sufficient accuracy.
 - deals with **transformational accuracy assessment**,
 - addresses the question of
Are we creating the M&S requirements right?
- **M&S Requirements Validity** is assessed by conducting M&S requirements validation.
- **M&S Requirements Validation**
 - is substantiating that the M&S requirements represent the real needs of the customer / client / sponsor with sufficient accuracy.
 - deals with **behavioral / representational accuracy assessment**,
 - addresses the question of
Are we creating the right M&S requirements?

M&S Requirements Clarity

is the degree to which the M&S requirements are unambiguous and understandable.

- **M&S Requirements Unambiguity** is the degree to which each statement of the requirements can only be interpreted one way.
- **M&S Requirements Understandability** is the degree to which the meaning of each statement of the requirements is easily comprehended by all of its readers.

- ❖ Always use **active (direct) voice** as opposed to passive (indirect) voice.
- ❖ **Active (direct) voice** is proven to provide better clarity.

M&S Requirements Completeness

- is the degree to which all parts of a requirement are specified with no missing information, i.e., each requirement is self-contained.
- **Examples:**
 - “radar search pulse rate must be 10” is an incomplete requirement because it is missing the “per second” part.
 - The requirement “missile kill assessment delay must follow the Uniform probability distribution” is incomplete because it is missing the uniform range parameter values.
 - Also use of the placeholder “TBD” (to be determined or to be defined), “TBR” (To be resolved), “TBP” (To be provided), and use of the phrases such as “as a minimum”, “as a maximum”, and “not limited to” are indications of incomplete requirements specification.

M&S Requirements Consistency and Feasibility

- **M&S Requirements Consistency** is the degree to which
 - a. the requirements are specified using uniform notation, terminology, and symbology, and
 - b. any one requirement does not conflict with any other.
- **M&S Requirements Feasibility** is the degree of difficulty of
 - a. implementing a single requirement, and
 - b. simultaneously meeting competing requirements.

Sometimes requirements conflict with each other. It may be possible to achieve a requirement by itself, but it may not be possible to achieve a number of them simultaneously.

M&S Requirements Stability and Modifiability

- **M&S Requirements Stability** is
 - a. the degree to which the requirements are changing while the M&S application is under development, and
 - b. the possible effects of the changing requirements on the project schedule, cost, risk, quality, functionality, design, integration, and testing of the M&S application.
- It is sometimes referred to as **M&S Requirements Volatility**

- **M&S Requirements Modifiability** is the degree to which the requirements can easily be changed.

M&S Requirements Testability

- is the degree to which the requirements can easily be tested.
- A testable requirement is the one that is specified in such a way that pass/fail or assessment criteria can be derived from its specification.
- For example:
 - the following requirement specification is **not** testable:
“The probability of kill should be estimated based on the simulation output data.”
 - The following requirement specification is testable:
“The probability of kill should be estimated by using a 95% confidence interval based on the simulation output data.”

M&S Requirements Traceability

- is the degree to which the requirements related to a particular requirement can easily be found.
- This is **Traceability among requirements!** It should not be confused with **Traceability throughout the life cycle.**
- Each requirement is given a unique code number in the requirements management software used. The unique numbers are used for cross-referencing and traceability among the requirements.
- A requirement is specified with cross referencing to the requirements it is related to so as to provide traceability.
- **Traceability among requirements enables maintainability.** If a requirement needs to be changed, we trace the requirements that are related to it, and also change those requirements if necessary.