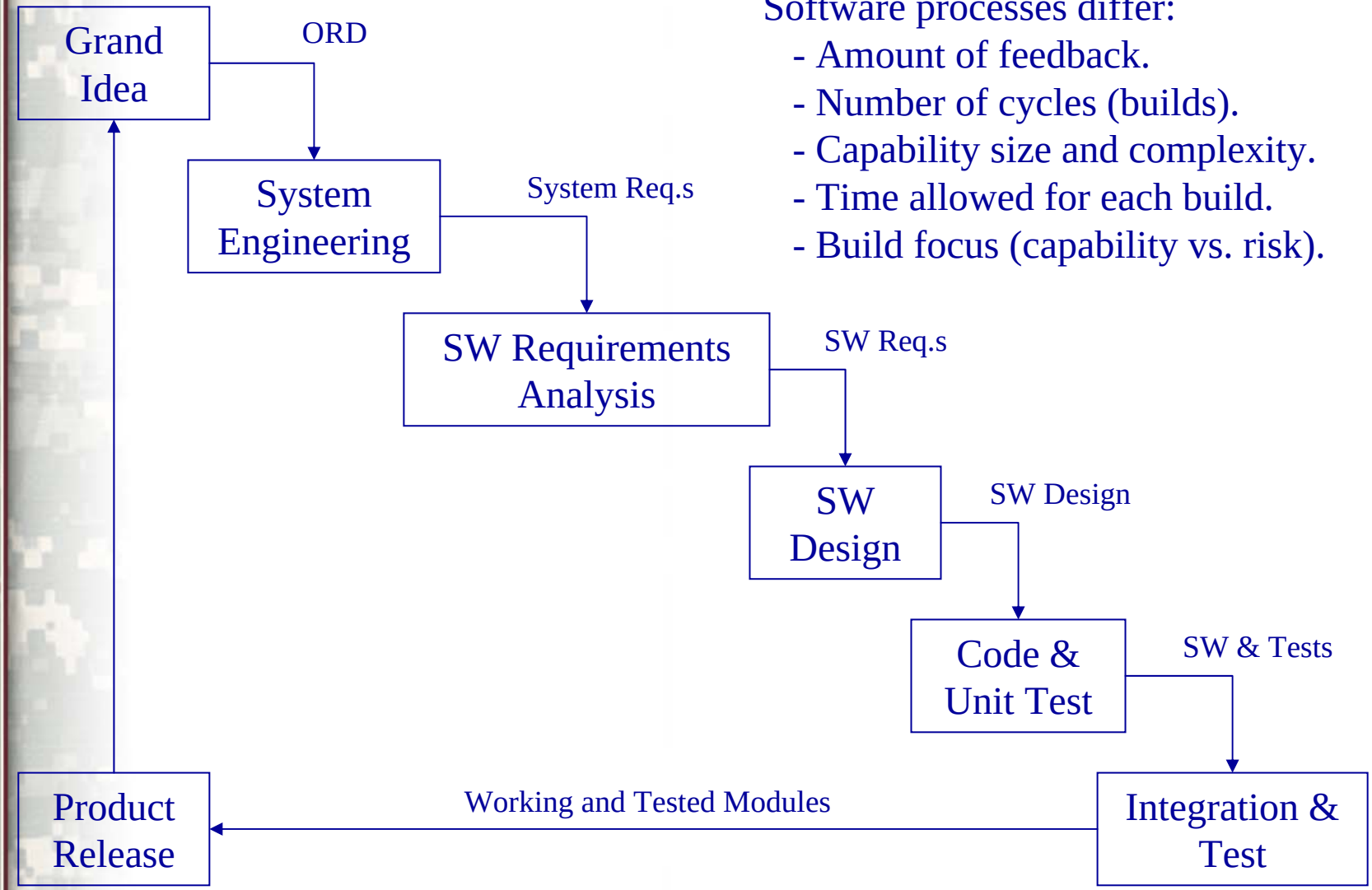
Conceptual Modeling in OneSAF Software Development

Outline

- Stereotypical Software Development Process
- OneSAF Software Development Process
- The Problem
- Conceptual Modeling
- OneSAF Abstract Model
- Example Conceptual Models
- Lessons Learned
- Conceptual Model Benefits

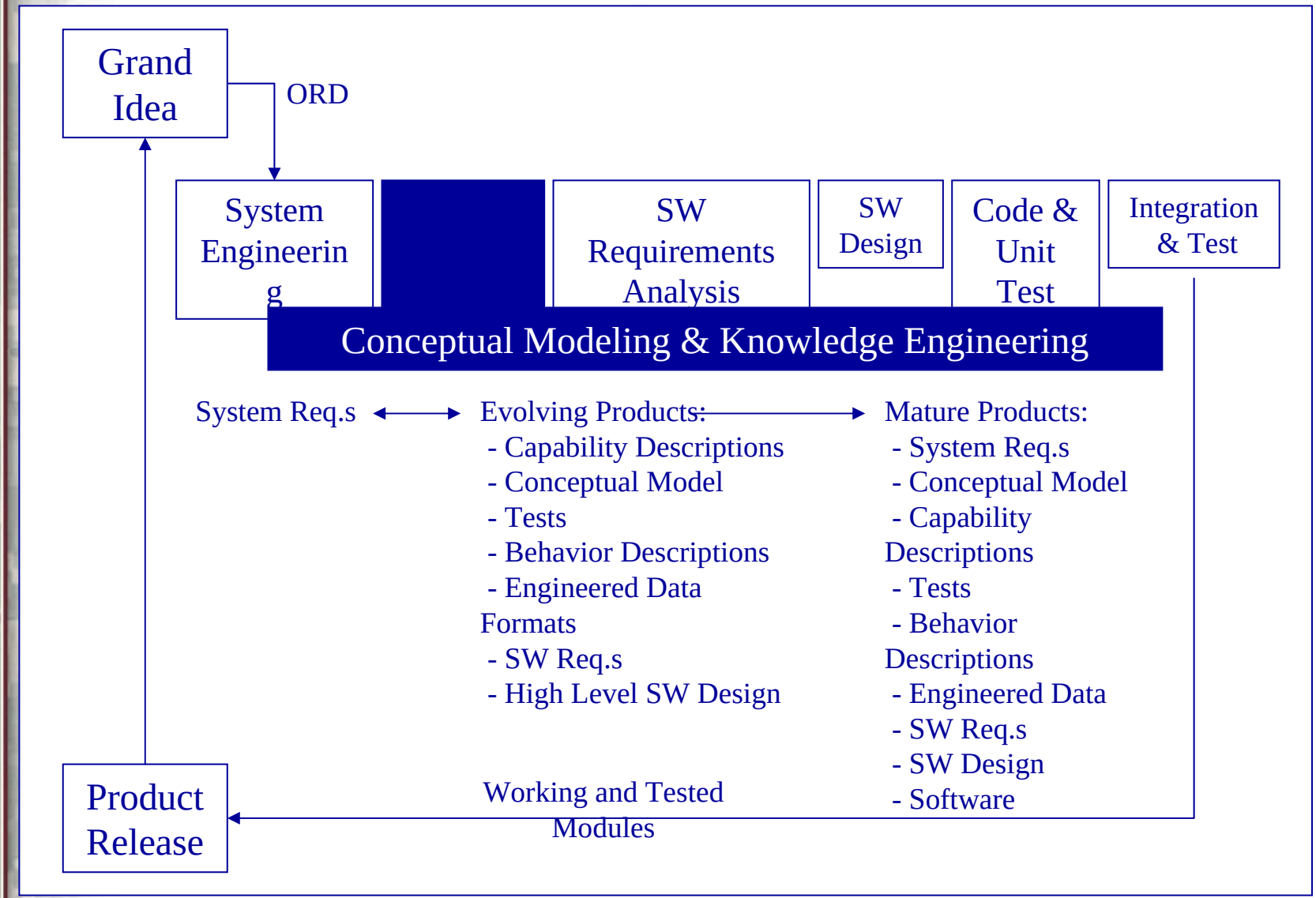
Stereotypical Software Development Process



Software processes differ:

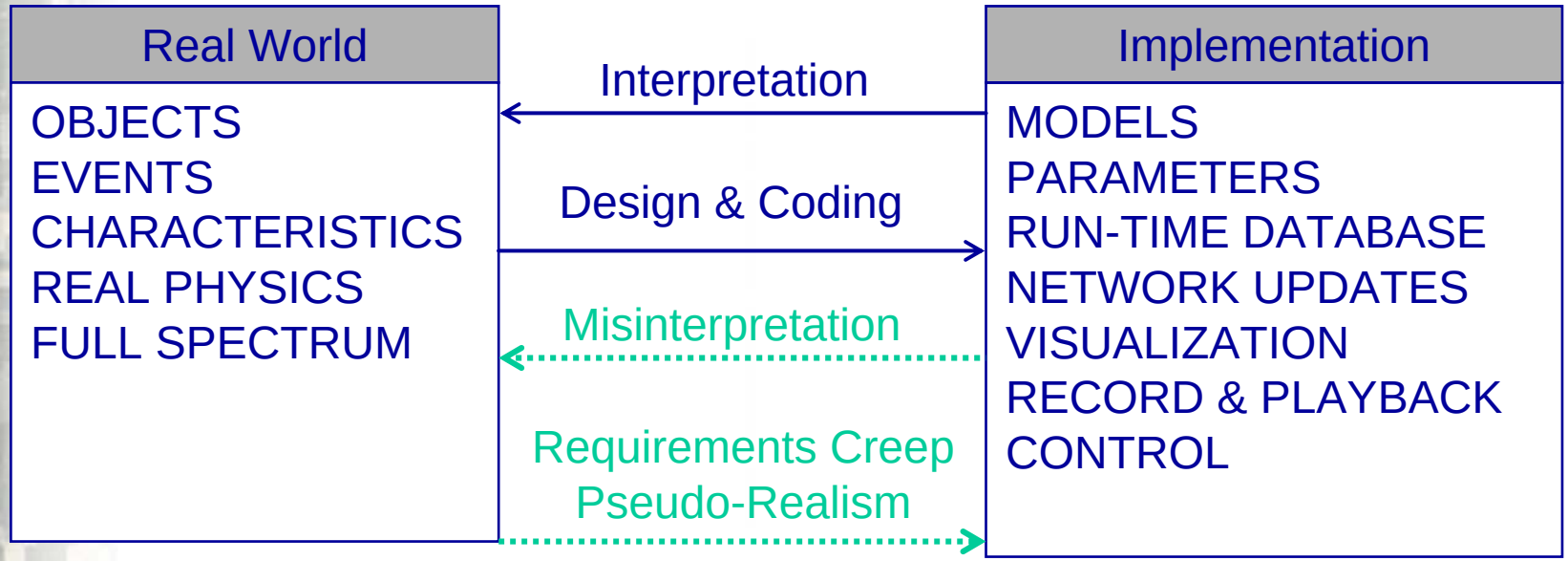
- Amount of feedback.
- Number of cycles (builds).
- Capability size and complexity.
- Time allowed for each build.
- Build focus (capability vs. risk).

OneSAF Software Development Process



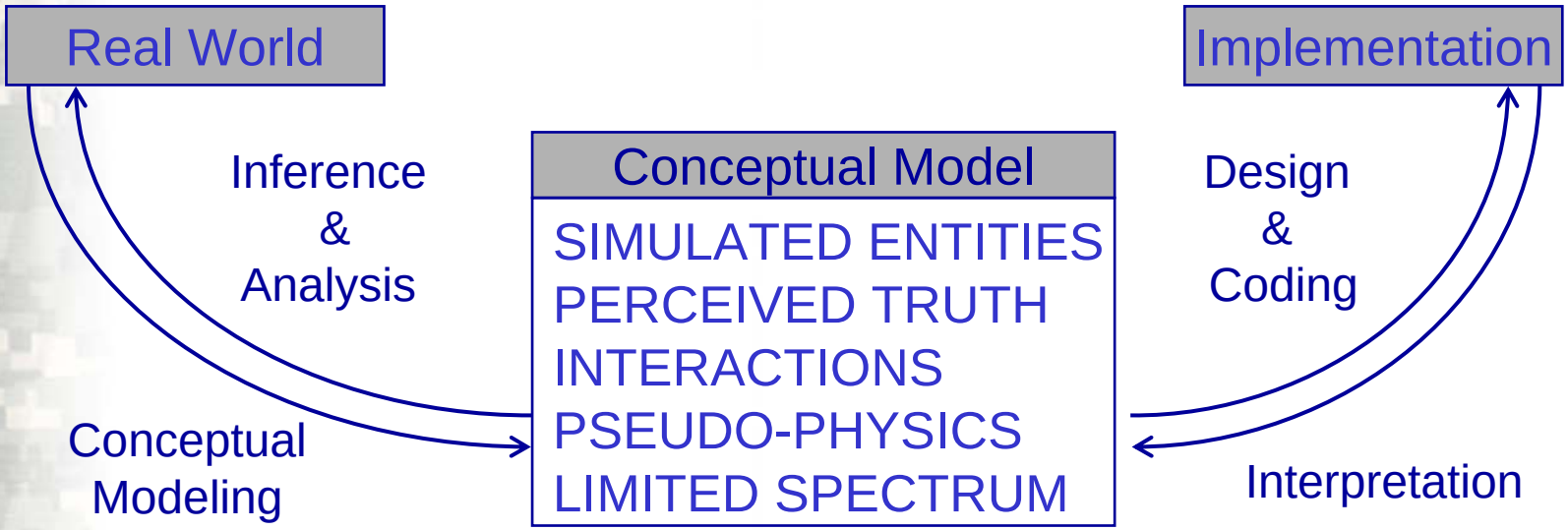
The Problem

Simulating the real world is difficult and error prone.



A major challenge is creating computationally amenable descriptions of the infinitely rich real world for the software development team to work with.

Conceptual Model



A well-conceived, consistent intermediate model eliminates many problems by providing a model of the battlespace usable by all participants (customer, domain expert, developer, and user).

OneSAF Conceptual Modeling

- A OneSAF Conceptual Model is an:
  - implementation independent
  - computationally amenable

  formulation of the battlespace.
- The OneSAF Conceptual Model sits between the rich detail of the real world and the computational structures of the simulation.
- OneSAF uses conceptual modeling to increase the efficiency and effectiveness of its team members.

OneSAF Conceptual Modeling and Command

"An order shall contain everything that a commander cannot do by himself, but nothing else." Moltke the elder

A conceptual model shall contain everything a software developer cannot do by himself, but nothing more."

- The software developer is not training to be a military commander.
- The software developer does not know what's important to model.
- The software developer, given an important topic, doesn't know the details.
- The software developer does know how to create computational structures and algorithms.

Hence, a good conceptual model and derived knowledge engineered products give the software developer all the information he needs, but does not know, to develop software and nothing more.

OneSAF Conceptual Modeling

A Conceptual Model is not a:

- Set of Requirements although a Conceptual Model is derived in part from system requirements and is used in deriving detailed requirements.
- Set of KA/KE documents although a Conceptual Model is derived in part from and refers to such documents.
- Software Design although a Conceptual Model constrains and focuses software design.
- SOM or FOM (Simulation or Federation Object Model) although a Conceptual Model provides information for SOM and FOM development.

Some Benefits

Shows before, during and after development:

- How the requirements are understood.
- What is being modeled.
- What models are implicit and explicit.
- The levels of fidelity.

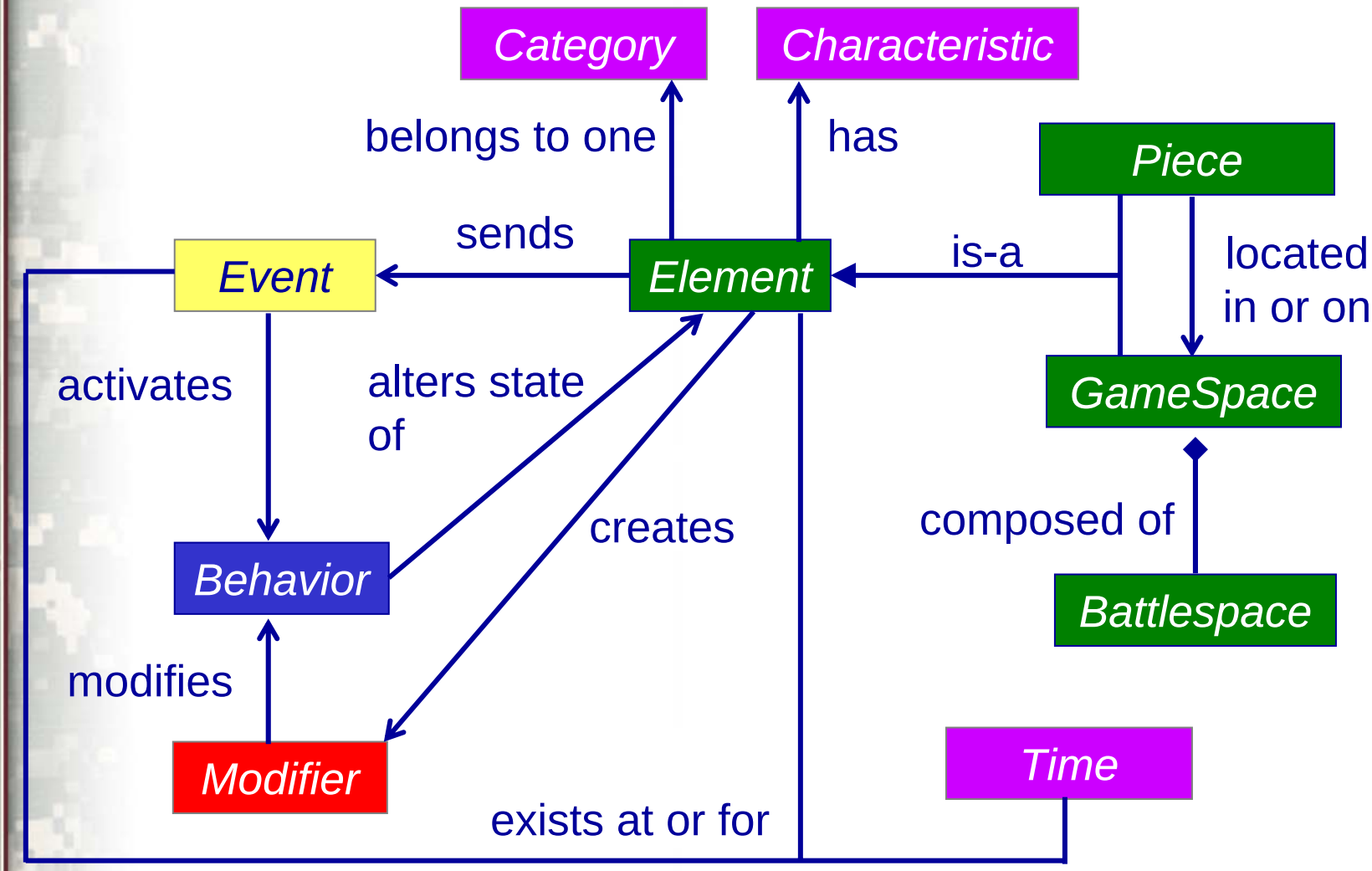
Provides structure to help organize design:

- Helps identify correlation and “Fair Fight” issues.
- Bounds the simulation complexity.
- Helps organize and constrain implementation.

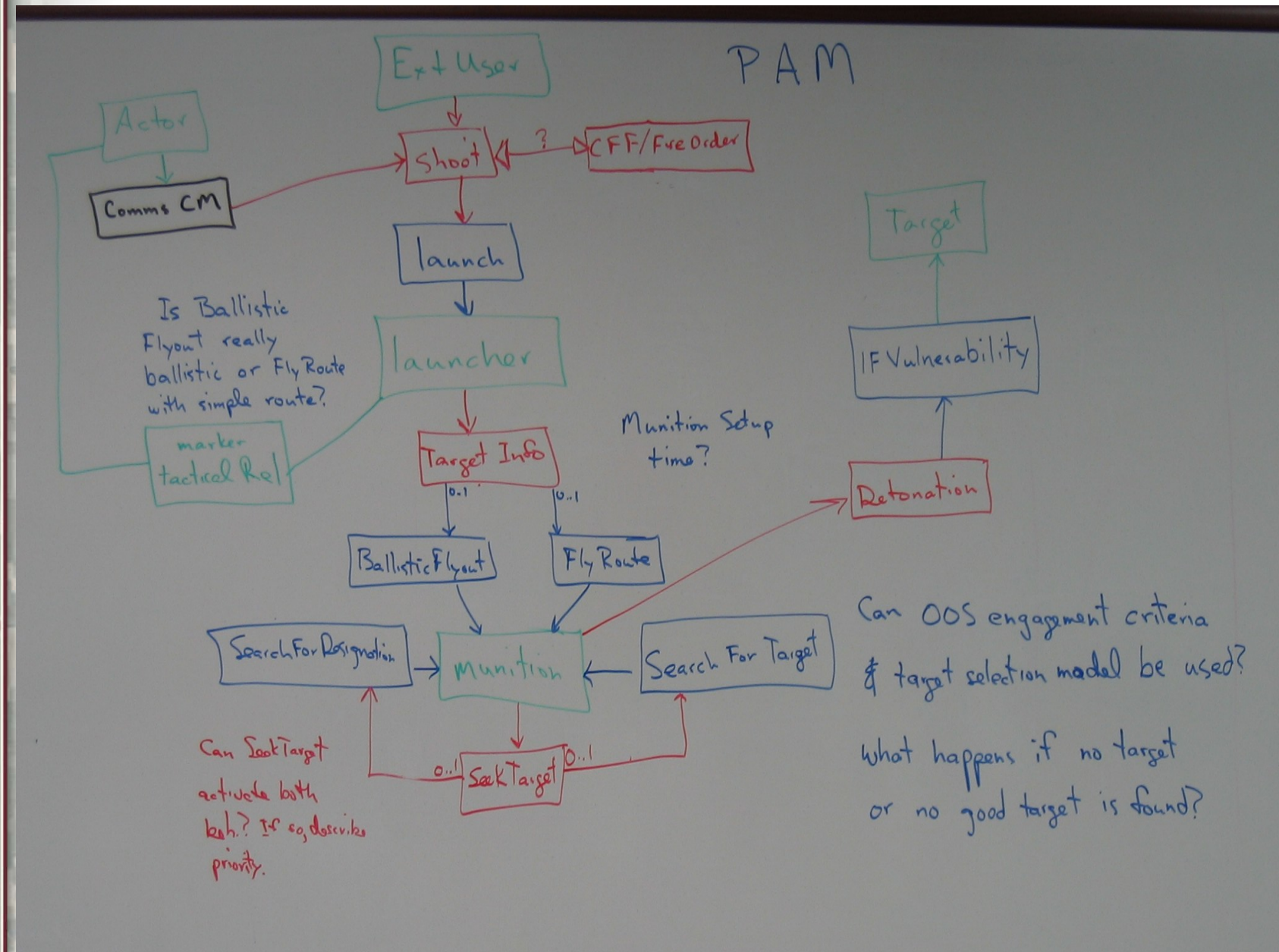
Assists developers and domain experts understand one another by providing a model intermediate between detailed real world and detailed design descriptions.

Abstract Model

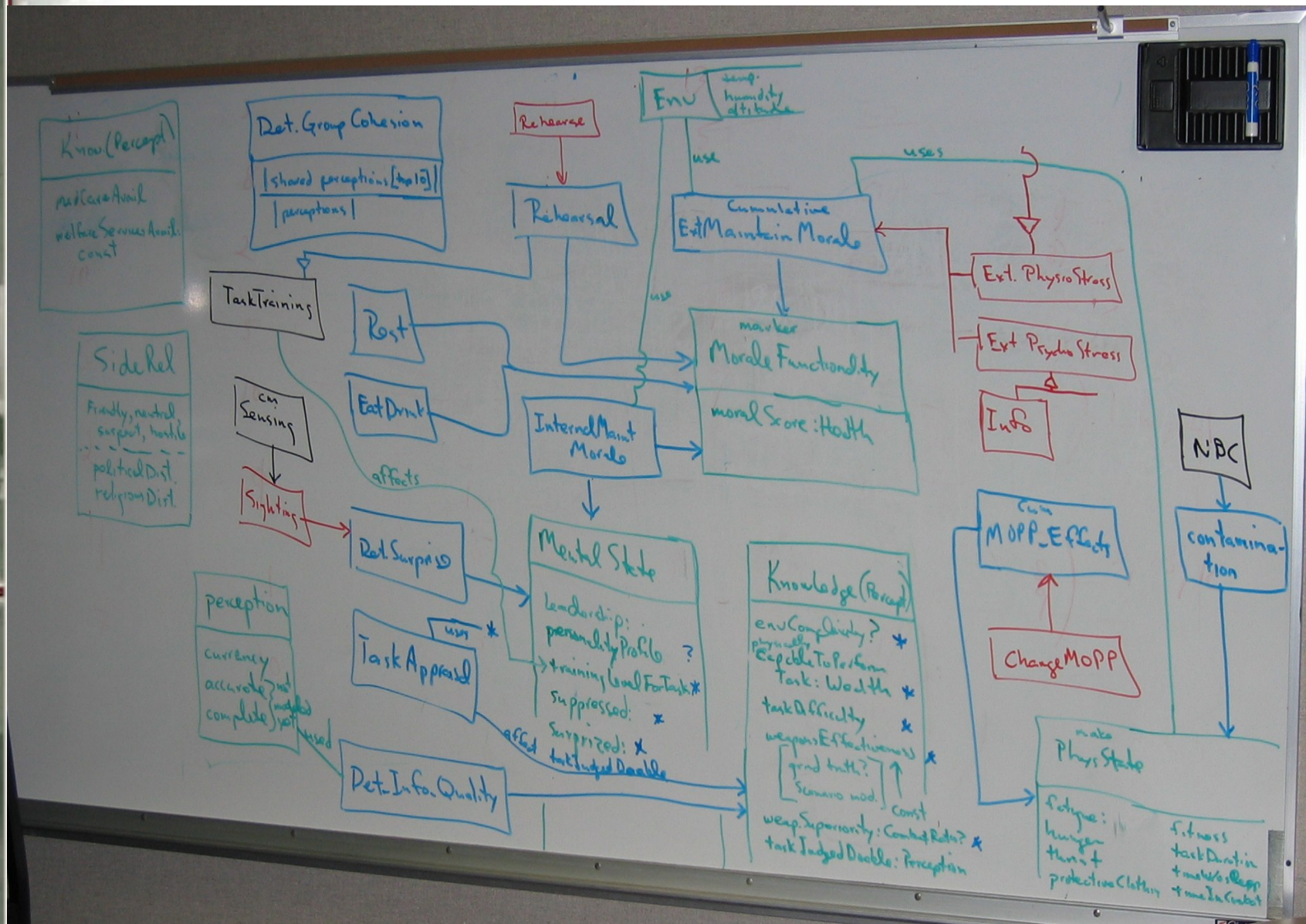
OneSAF Conceptual Modeling Language (CML)



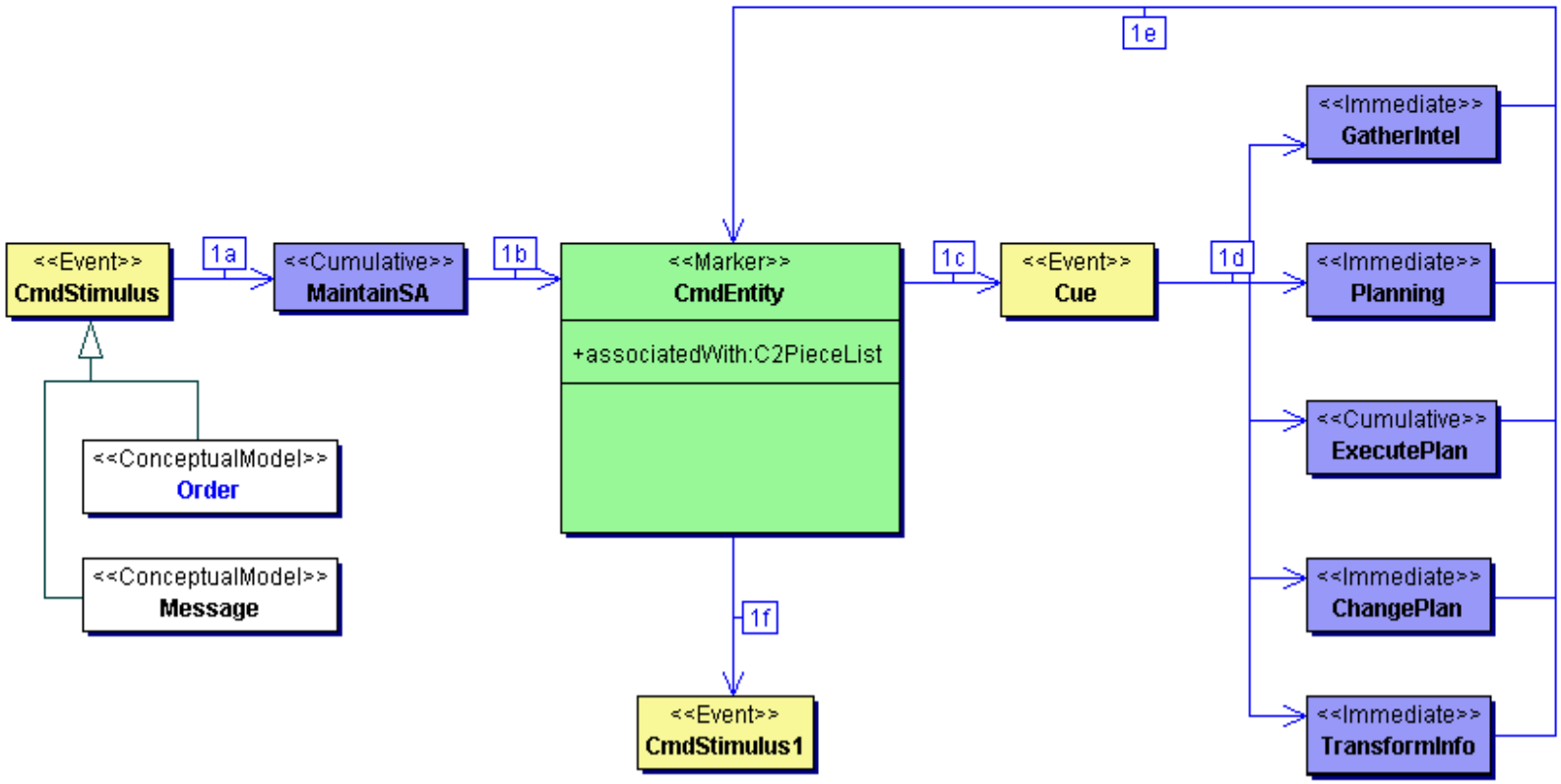
Precision Attack Munitions Example



Morale Example



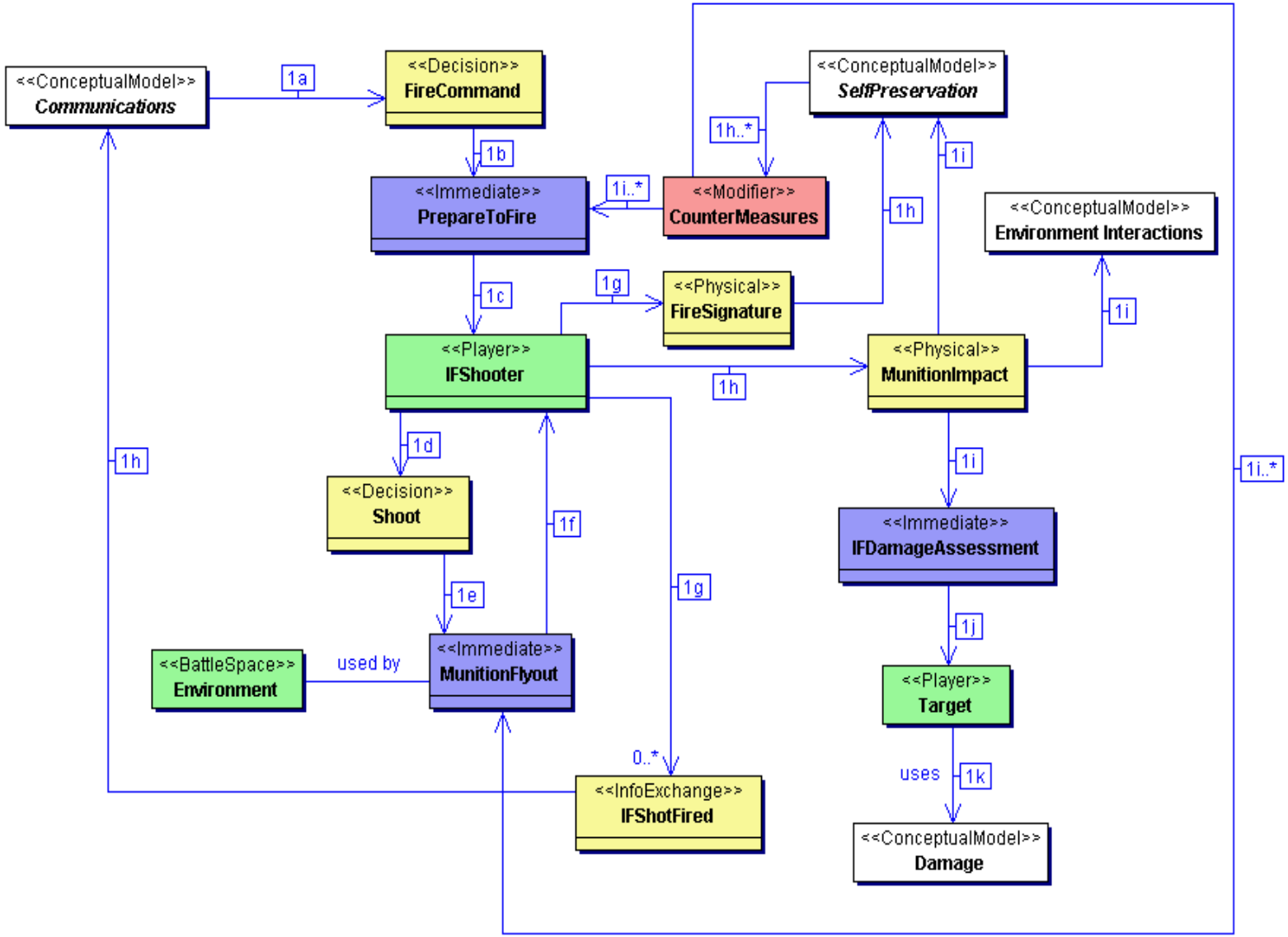
Conceptual Model:
Command Entity Example



Command Entity Example

- Elements:
  - CmdEntity
- Events
  - CmdStimulus
  - Cue
- Behaviors:
  - MaintainSA
  - GatherIntel
  - Planning
  - ExecutePlan
  - ChangePlan
  - TransformInfo
- Other Conceptual Models:
  - Order
  - Message

Indirect Fire Shooter Example



Indirect Fire Shooter Example

- Elements:
  - IFShooter
  - Target
  - Environment
- Events
  - FireCommand (Decision)
  - Shoot (Decision)
  - FireSignature (Physical)
  - MunitionImpact (Physical)
  - IFShotFired (InfoExchange)
- Behaviors:
  - PrepareToFire
  - MunitionFlyout
  - IFDamageAssessment
- Modifier:
  - CounterMeasures
- Other Conceptual Models:
  - Communications
  - Damage
  - SelfPreservation
  - Environment Interactions

First Lesson Learned

Begin conceptual modeling at the same time and in conjunction with Knowledge Acquisition and Knowledge Engineering (KAKE).

Without conceptual modeling, KAKE and Model Development resources are wasted:

- producing documentation and data irrelevant to requirements and simulation capabilities.
- revisiting and completing KAKE artifacts for consumption.

Second Lesson Learned

Include a few developers in the conceptual modeling/KAKE effort.

- These “junior conceptual modelers”/developers increase the efficiency and effectiveness of development by:
  - understanding the topic through the give and take among conceptual modeling, SE and KAKE and
  - completing RA and top level design early.
- So far, OneSAF junior conceptual modelers/ developers have completed:
  - not only the RA and high level design but also
  - the detailed design,
  - and, in some cases, began coding prior to the scheduled development build.

Third Lesson Learned

Finally, the adage “a picture is worth a thousand words” was confirmed.

- Neither the KAKE nor the developers have time or interest in learning another artificial language.
- The simple, colored OneSAF CML proves adequate for communication without excessive formalism.
- The Conceptual Modeler uses the OneSAF CML to elicit information and record modeling decisions in working groups.
- In some cases, it is unnecessary to reveal a conceptual model. Instead, the Conceptual Modeler uses the OneSAF CML to understand the topic and then suggests to KAKE the relevant topics and data to describe.

Conceptual Model Benefits

Conceptual modeling applied throughout development:

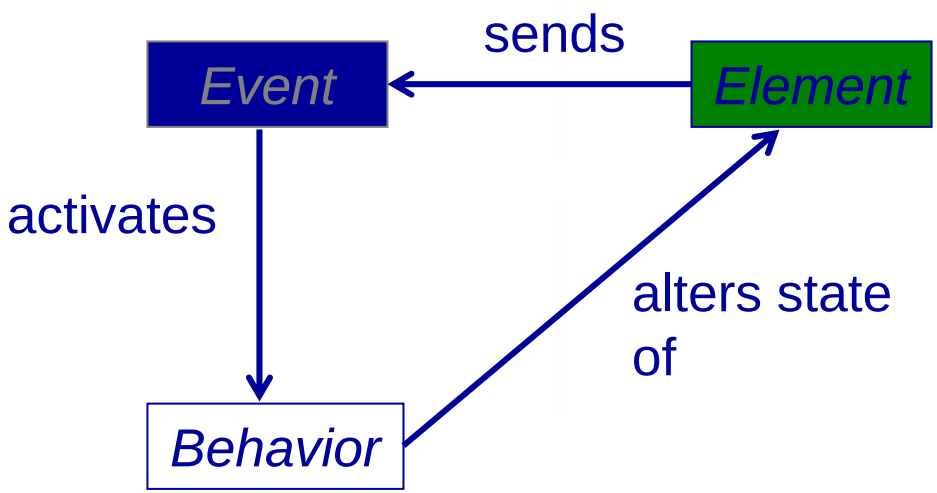
- improves KAKE and developer productivity,
- provides a common frame of reference for all shareholders,
- minimizes requirements creep by limiting KAKE to relevant issues,
- provides a sufficiently detailed description of the modeling solution to minimize misinterpretations and reinterpretations of the requirements.

Backup Slides

Abstract Model

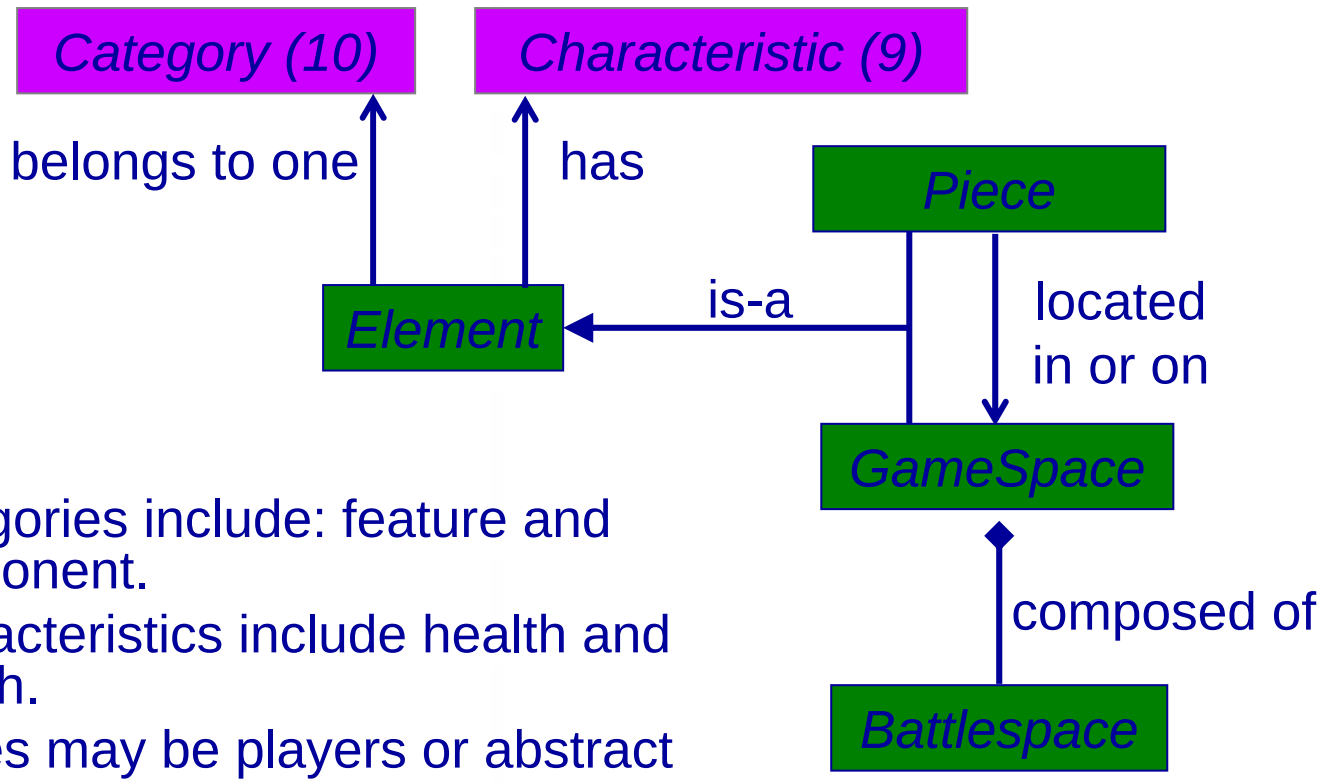
OneSAF Conceptual Modeling Language

- Colored graphical notation using:
  - colored boxes,
  - arrows,
  - a small subset of UML (Universal Modeling Language).
- The central flow of information is:



Elements

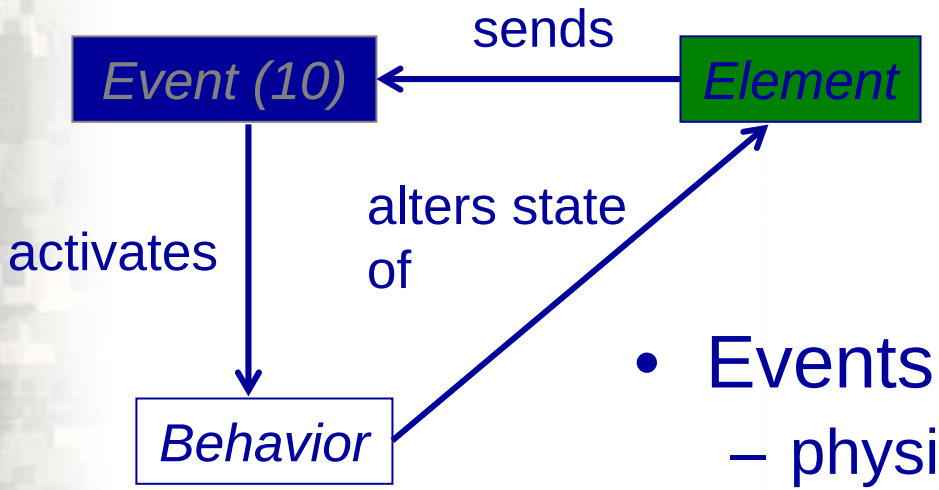
OneSAF Conceptual Modeling Language



- Categories include: feature and component.
- Characteristics include health and wealth.
- Pieces may be players or abstract markers.
- Gamespaces may be physical environments or abstract zones.

Events and Behaviors

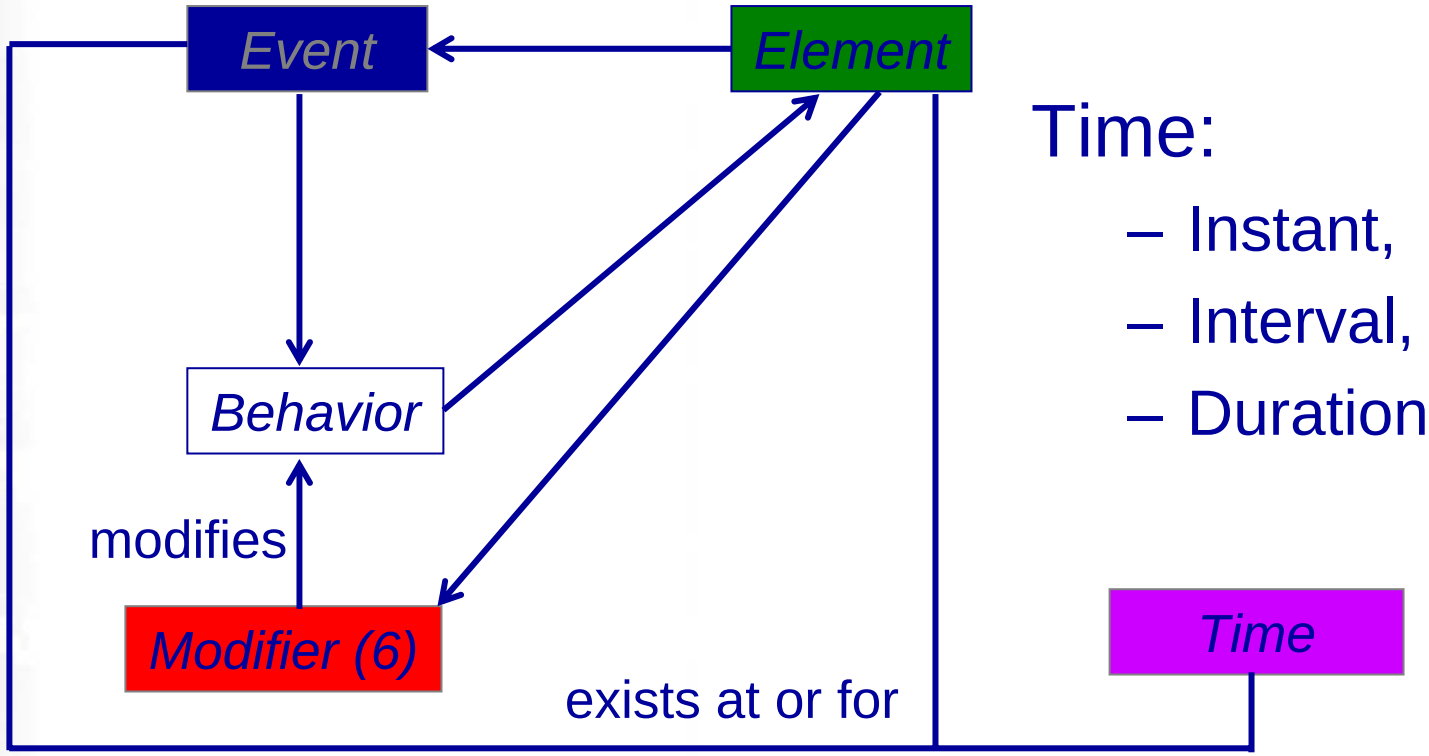
OneSAF Conceptual Modeling Language



- Events include:
  - physical,
  - state change,
  - element coordination.
- Behaviors are:
  - immediate (instantaneous)
  - cumulative (have duration).

Modifiers and Time

OneSAF Conceptual Modeling Language



Time:

- Instant,
- Interval,
- Duration.

Modifiers are: Cancel, Overwrite, Delay, Alter Logic, Alter Data, and Randomize