



Decomposition Diagrams

BOX AND ARROW DIAGRAMS

Break down into parts

- Data to tables/fields
- Code to packages/classes/methods/fields
- processing to states
- processes, threads
- system->subsystem->subsubsystem->

Decomposition - an important skill!



Decomposition in **ANALYSIS**

Catalog the parts that are there
in the existing system.



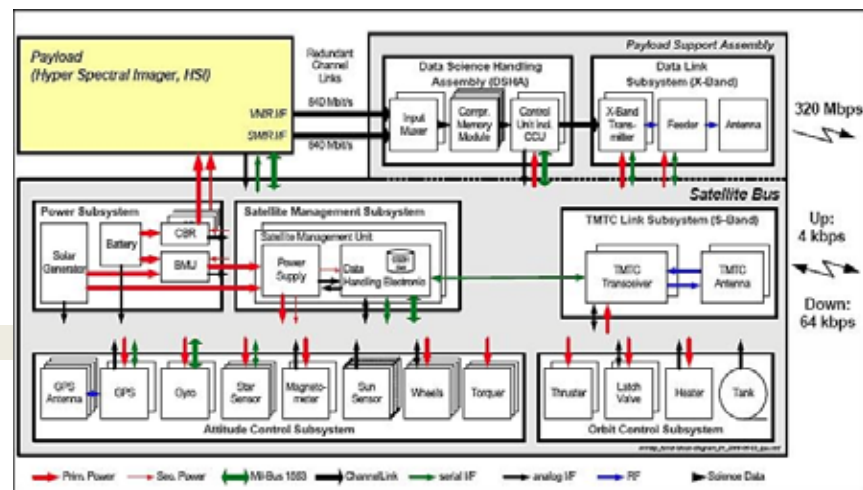
Decomposition in **DESIGN**

Imagine the parts that you think
the to-be-built system should have

Decomposition Diagram

- For our project, we will decompose each component into modules.
- These will be depicted by rectangular boxes.
- The modules will be connected by arrows.

These are decomposition diagrams.



What does the arrow mean?



A instantiates B
A calls B
A writes to B (a data store)
B is reading from A (a data store)
A depends on B

In our diagram, we will
use the arrow to mean
ALLOWED TO USE.

Allowed-to-use relationship

- not same as a use relationship
 - during implementation may not actually use
- not same as calls relationship
 - one may call a method in an upper layer without using the upper layer. Example is an event based system – which will call the event handler without knowing anything about the event handler. During compilation – the event handling layer code is not required.