**Items overview for the Runtime Engine**

As discussed in the Sept-13 meeting, an Item should be specified pretty high-level, and then interpreted into the low level for the specific simulator.

|  |
| --- |
| **Strategic**  No specific mentions, just attributes of the item (difficulty, groupings, crossing, etc) |
| **Operational**  Specific mentions of what is going on in an item, but not WHO is involved |
| **Tactical**  Specific mention of the active participants of an item |

As each item will eventually need to be able to be translated into a Tactical description, we’re able to make an item contain AT LEAST Strategic components, with Operational or Tactical components being optional. This means that as the runtime engine consumes an item, if the operational or tactical sections of the item were not defined in the item definition, then the runtime engine will populate that item with the appropriate types of info. This allows for an item to be defined either in the item, or by the runtime engine code.

For each level, there’s a certain set of attributes that should be filled in. These attributes should be “inherited” by the lower levels (so “ID” is defined in Strategic, an as Operational and Tactical are all “lower levels” of detail, will inherit that attribute).

Strategic

* ID – unique identifier
* Crossing – Yes/No
* Groupings – One/Two
* Threat – Unambiguous/Ambiguous/None
* Player Resource Availability – Yes/No
* Teammate Resource Availability – Yes/No
* Difficulty - -1.0 – 1.0

Operational

* What is happening – Non-threatening Pirate, threatening pirate
* Where is it happening – Within X meters of the sensors of player B, within X m of Port Y, in the northeast quadrant, etc

Tactical

* What objects are involved? E.g. Pirate1, MerchVessel1, MerchVessel2
* What is each object doing? E.g. Pirate1 revealed, then moves towards MerchVessel1, attacks within range, MerchVessel1 revealed, then moves towards Port X
* Where is each action being taken? E.g. Pirate 1 reveals within X meters of MerchVessel1, MerchVessel1 reveals at 75, 156, etc
* When is each object doing this? Pirate1’s move is 15 seconds after it is revealed

What objects are involved

What is each object doing

Where is each action being taken

When is each action being taken

What is happening

Where is it happening

ID

Crossing

Groupings

Threat

Player Resource Availability

Teammate Resource Availability

Difficulty

Use Case Examples:

**Specify an Item with Strategic Level info**

* An item is defined with the following attributes:
  + ID – Item18
  + Crossing – No
  + Groupings – One
  + Threat – Yes, ambiguous
  + Player Resource Availability – Yes
  + Teammate Resource Availability – Yes
  + Difficulty – 0.25
* When this item is read into the Runtime Engine, the runtime engine needs to convert these high level attributes into low level actions.
* A process in the code will be tasked with looking at all of the attributes, and determining which objects to create/move/attack, where those objects will be, and what actions will be taken. This should all be according to the constraints outlined in the above attributes (if crossing, set locations for actions to be near the crossing zone; if it’s an ambiguous threat, make sure to reveal without pirate IFF squawk; etc).
* Once the item is processed, and the tactical level info is filled in, then the item is passed to the process which performs the work based on the item.

**Specify an Item with Tactical Level info**

* An item is defined with the following attributes:
  + ID- Item30
  + Crossing – Yes
  + Groupings – One
  + Threat – Yes, ambiguous
  + Player Resource Availability – No
  + Teammate Resource Availability – Yes
  + Difficulty – 0.65
  + Operational Specific:
    - What is happening? Ambiguous pirate attack
    - Where is it happening? Northwest of port XYZ
  + Tactical Specific:
    - Objects: P1 (pirate), P2 (pirate), MV1 (merchant)
    - Actions: Reveal(P1, Loc1), Reveal(P2, Loc 1), Reveal(MV1, Loc2), Move(P1, Intercept(MV1), T1), Move(P2, Intercept(MV1), T1)
    - Locations: Location1={within 50m of Port XYZ}, Location2={within 1m of 150, 265}
    - Times: Time0={0 sec after item starts}, Time1={10s after item starts}
* When this item is read into the Runtime Engine, the runtime engine has all of the low level info that it needs, so the item can be passed to the process which performs work based on the item.