Task 20 - Spike (ext): Messaging Extended

Context: Message systems can support a number of very useful features, such as broadcasting, filtering messages based on game-world properties, repetition of messages, and scheduling message release.

Knowledge/Skill Gap: The developer has implemented or is using an existing messaging system and needs to include additional useful features.

Goals/Deliverables:
[DESIGN] + [CODE] + [SPIKE REPORT]

"Extend" the previous spike to include one or more of the following additional message system features. Update your design documents to reflect the changes of your new system.

Possible extension features/support:
- Broadcast messages (specified by the sender)
- Filtering of messages before delivery/pickup (by the blackboard/dispatch system, not the sender),
- Scheduling of messages for the future,

You may want (or need) to include the ability for the sender to "delete" or cancel messages in support of the above features.

Examples for filtering of message could be based on game entity values/types, or locations

You need to produce:
1. Updated design documents (UML class, module, sequence etc) as applicable, clearly showing what you have had to add to support your additional features, and an
2. Updated working code demonstration within the Zorkish game example.

Recommendations:

- Only take on this extension challenge if you are comfortable with your other core work so far. You can always come back to this when you are better prepared.
- Combat! A message system is often a key part of any combat system. This might be a good target for you.