Graph Specification TimeStamping

Table of contents

Message Passing

Once the graph is built, the communication between various entities occur by passing around messages (messages are entities themselves). Specifically, one component/codelet can publish a message entity and another can receive it. When publishing, a message should always have an associated `Timestamp` component with the name *"timestamp"*. A `Timestamp` component contains two different time values (See the `gxf/std/timestamp.hpp` header file for more information.):

1. `acqtime` - This is the time when the message entity is acquired, for instance, this would generally be the driver time of the camera when it captures an image. You must provide this timestamp if you are publishing a message in a codelet.

2. `pubtime` - This is the time when the message entity is published by a node in the graph. This will automatically get updated using the clock of the scheduler.

In a codelet, when publishing message entities using a `Transmitter (tx)`, there are two ways to add the required `Timestamp`:

1. `tx.publish(Entity message)`: You can manually add a component of type `Timestamp` with the name "timestamp" and set the `acqtime`. The `pubtime` in this case should be set to `0`. The message is published using the `publish(Entity message)`. **This will be deprecated in the next release.**

2. `tx.publish(Entity message, int64_t acqtime)`: You can simply call `publish(Entity message, int64_t acqtime)` with the `acqtime`. Timestamp will be added automatically.