

RELAZIONE FINE FASE 2 – ISS25

Repository GitHub: <https://github.com/gioliee/IngegneriaDeiSistemiSoftware25/tree/main>

In questa seconda fase, innanzitutto, si sono eseguiti e analizzati molteplici casi d'uso del modello ad attori per prenderne familiarità. Il modello ad attori con una metafora possiamo esprimerlo come il “mondo dei vivi”, che ci fa abbandonare il “mondo dei morti” del modello ad oggetti. Queste due metafore servono per far intendere il comportamento di questi due approcci all'interno di un sistema: il modello ad oggetti, infatti, incapsula dati e comportamenti all'interno di classi portando a sistemi rigidi e difficilmente adattabili a contesti dinamici utilizzando pressoché comunicazioni sincrone, all'opposto, il modello ad attori rappresenta entità autonome che comunicano tra loro attraverso lo scambio di messaggi asincroni (comunicazione inoltre su differenti protocolli, ad esempio, nel nostro caso TCP, MQTT e COAP), inoltre possiamo definire un attore come un automa a stati finiti, il quale è in grado di intervenire sul proprio stato interno rispondendo a messaggi/eventi esterni senza dover rendere esplicite tutte le transizioni, il comportamento dell'attore è quindi definito dal suo stato attuale e dal tipo di messaggio ricevuto. A questo cambio di modello è seguito anche un cambio di approccio da bottom up a top down. Questa transizione era necessaria affinché si portasse ad analizzare il problema in quanto tale per poi capire quale tecnologia avesse gli strumenti migliori a risolverlo, se una sua immediata risoluzione non era possibile con ciò che il mercato offre allora si va a creare il cosiddetto abstraction gap, ovvero la distanza tra il problema da risolvere e le astrazioni disponibili nelle tecnologie correnti, il quale deve essere colmato per arrivare a una reale risoluzione e per farlo ci sono diverse strade librerie, altre tecnologie o lo sviluppo di software custom, come QAK, utilizzato nel corso, che semplifica la progettazione, comunicazione e il deployment di attori su nodi diversi, supportando la costruzione di sistemi distribuiti ed eterogenei.

In secondo luogo, questa seconda fase è stata anche utilizzata per introdurre il raspberry e il suo utilizzo come nodo esterno per andare sempre di più a lavorare nel concreto con i sistemi distribuiti.

Tutti questi nuovi approcci hanno portato, come detto in precedenza, all'esecuzione di diversi esperimenti che utilizzassero questi nuovi approcci. In particolare, soprattutto all'inizio, si è ripreso in mano il gioco ConwayLife

trasformandolo in un modello ad attori, questo ha permesso non solo di analizzare le macro-differenze tra il modello ad attori e quello ad oggetti ma anche a costruire un sistema distribuito su vari nodi; infatti, ogni raspberry doveva rappresentare una cella della griglia abbandonando la centralizzazione della logica, ma attribuendo autonomia agli attori (in questo caso le celle). L'attenzione poi si è spostata verso l'utilizzo del raspberry e del modello ad attori con QAK nell'uso di sensori e robot virtuali, sempre in modo incrementale. Questo perché gli attori QAK possono modellare bene agenti situati in quanto percepiscono input, prendono decisioni internamente allo stato e agiscono producendo output. Questo aspetto è molto importante in quanto ogni componente può essere trattato come un microservizio autonomo e visto che il microservizio può essere deployato su un nodo differente ecco che entra in azione il raspberry.

In conclusione, possiamo affermare che questa seconda fase ha dato una svolta nell'approcciarsi al problema col top-down e una visione di programmazione differente con gli attori, attivamente comunicativi, andando ad esplorare nuove tecnologie hardware (come raspberry e sensori annessi) e software.