SPRINT 5 – Ottimizzazione del comportamento del waiter nella gestione di più clienti.

**Interleaving e parallelizzazione dei task del waiter**

Nel momento in cui suona il campanello, quali sono i possibili task che il waiter sta facendo?

Sono interrompibili da questo oppure no? Posso pensare che anziché interromperli alcuni task possono essere fatti in parallelo? Perché magari sono poco impegnativi computazionalmente… (Esempio: mentre servo un cliente rispondo alla smartbell dicendo: Si c’è posto/No).

Idea di fondo: Forse solo il task clean e rest sono interrompibili perché gli unici due non dedicati ad un cliente. Magari il clean può essere diviso in 3 sottotask non interrompibili…

Altra situazione:

* io sto pulendo un tavolo e l’altro tavolo è libero e pulito, ha senso che mi interrompa e accompagni il cliente al tavolo libero e poi riprenda a lavare il tavolo riprendendo da dove aveva lasciato, e non da zero.

Sono tutte considerazioni che posso fare nell’analisi del problema… e quindi fare una considerazione esempio sul fatto che il problema richieda di salvare il tempo di pulizia già fatta… oppure dire che questo determinato interleaving e priorità tra le azioni del waiter sia dettata da un -motivo economico: voglio sfruttare al massimo le risorse del locale per poter guadagnare di più.-

Quali sono i messaggi/richieste che possono causare un interruzione?

*(richiesta proveniente dalla smartbell)*

-enterRequest?

*(richiesta proveniente dal client)*

-readyToOrder

-readyToPay

-Scadenza del timer maxStayTime! Ha l’autorità di interrompere più task delle altre richieste? Come trattarla?

Poi?

-OrderReady ?

-TimeToRest?

Quindi andiamo verso l’idea che il waiter possa essere composto da più ‘entità computazionali’ (più attori) lavoranti in parallelo. Da una prima visione dall’esterno lo vediamo come unica entità…ma poi all’interno è fatto da più componenti (Es: basicrobot – waiterwalker – waiterlogic – waiter che risponde alle domande di ingresso… ecc) 😊

-Magari nello sprint in cui affronto l’interleaving delle attività mi accorgo che devo predisporre uno stop del corpo (succede quando risetto il goal e faccio un replan), cosa che magari non serviva negli sprint precedenti. “L’analista può riporsi il problema mente-corpo e dire: “E’ ancora opportunuo tenere partizionati i due e cambiare magari un po’ il corpo” oppure “no, mente e corpo non ha più senso perché…”

Per rendere interrompibile un movimento del waiter occorre **cambiare il modello del waiterwalker (!?) in modo che ascolti i messaggi movetoCell** anche quando sta eseguendo il piano precedente. Valutare se serve memorizzare il goal precedente per riprenderlo eventualmente dopo.

* Raggiungere il teatable nella cella più vicina alla posizione corrente del waiter: Si potrebbe scrivere magari una funzione findNearestTableCell che, data la posizione attuale del waiter e il tavolo che si vuole raggiungere, dica qual è la cella più vicina.