# **Domande che Natali fa all’orale**

1. Fammi vedere il progetto (sostanzialmente il progetto con il file.qak)
2. Fammi vedere gli artefatti risultanti dall’analisi dei requisiti e del problema (metamodelli e modelli con basi di conoscenza, eccetera)
3. Fammi vedere almeno un test plan che avete realizzato (i test plan si fanno dopo l’analisi dei requisiti)
4. Come vi siete organizzati? (Scrum)
   1. Dopo ogni sprint avete fatto un nuovo progetto? E perché? (Sì, così si può tornare indietro e si può vedere l’aggiunta incrementale di informazioni nel progetto)
   2. Per ogni sprint serve il suo product backlog: conservalo da qualche parte
5. Per quale motivo avete usato questo linguaggio e questa tecnologia?
6. È conveniente usare il Model Driven Software Development? Perché?
7. Web of Things, cos’è? L’avete usato? Perché? (Pro/Contro, fare l’es. di Docker)
8. Fatemi vedere il work plan che avete prodotto all’inizio subito dopo la fase di analisi (è circa un PB con la tecnica dello zooming top-down)
9. Consegnatemi una pagina F/R con ciò che ha fatto ognuno di voi

# **ANALISI DELLA TRACCIA D’ESAME**

Stanza dedicata con:

* Frigo intelligente: ha una rappresentazione del suo interno e risponde via COAP a domande sul suo contenuto (interazione M-M e H-M)
* Lavastoviglie: non è smart, ma se lo fosse? Considerare l’eventualità…
* Dispensa: non è smart, ma se lo fosse? Considerare l’eventualità…
* Robot: butler/maggiordomo detto RBR
* Tavolo 🡪 al centro

La disposizione di frigo, lavastoviglie e dispensa è variabile, ma il prof ha detto che possiamo supporre che la configurazione sia quella in figura. Possiamo però prevedere una variazione, che non impatterebbe sul nostro problema se noi supponiamo di avere la posizione dei vari elettrodomestici. Se all’esame perciò lui dovesse prendere il nostro robot e dovesse metterlo nella sua scatola e non nella nostra, basta che ci dica la posizione degli elettrodomestici e tutto dovrebbe funzionare.

Dobbiamo avere un software sul frigo e uno sul RBR + la business logic sul PC.

Modello della stanza: la stanza ha una dimensione (larghezza e lunghezza), al suo interno c’è la casa del robot (RH) e al centro c’è un tavolo (che si può spostare se qualcuno ci prende contro, quindi ha senso che venga riconosciuto in una fase di discover dal robot). Supponiamo anche di avere la posizione dei vari elettrodomestici.

La stanza è divisa in celle quadrate che si suppone siano grandi come il robot. Per misurare i gradi di rotazione del robot bisogna guardare quanti secondi ci mette, stessa cosa per misurare l’avanzamento di un passo: bisogna guardare quanti secondi ci mette a percorrere una distanza pari alla sua stessa lunghezza. (Per la stanza fare riferimento alla distanza di Manhattan)

Il maître è dentro la stanza e può dare al robot tre CMD:

* PREPARE: robot va dalla casa alla dispensa e dalla dispensa al tavolo (il robot ha capacità infinita o deve fare tanti giri in base a quanta roba sta caricando? Il prof dice che si suppone che il robot abbia capacità infinita, quindi basta un solo giro). Poi va dal frigo al tavolo (o il contrario? Rifletterci e pensare, ma ha più senso fare in questo ordine: prima porto i piatti, poi porto il cibo). Infine torna a casa, in RH.
* ADD FOOD: durante la festa aggiunge cibo se c’è, quindi il viaggio è da casa al frigo, dal frigo alla tavola e dalla tavola a casa.
* CLEAR: il robot deve rimettere il cibo non mangiato in frigo e i piatti in lavastoviglie. (il prof per DISHES intende proprio le stoviglie, non è cibo secco. Possiamo però considerare la possibilità che nella dispensa ci siano anche cibi secchi tipo i cracker. Farlo presente in analisi)

Scenario o sequenza di passi cronologici in generale:

* Tempo T0: il maître mette in frigo e in dispensa tutto il necessario. Si ha un set prefissato di items. La stanza è vuota, c’è solo il maître. La lavastoviglie è vuota. RBR è in RH.
* Il maître invia PREPARE e attende la fine del task. RBR poi torna a RH.
* Inizia la festa: il maître invia ADD FOOD specificando UN FOOD-CODE. Se c’è in frigo, il robot parte, se no manda un warning. Domanda: è possibile che il maître possa non sapere se manca un cibo? Lui dovrebbe poterlo controllare da smartphone… Da analizzare.
  + Caso 1: il cibo c’è, allora il robot parte
  + Caso 2: il cibo non c’è, robot invia un warning
    - Motivo della mancanza: è finito
    - Motivo della mancanza: non c’è mai stato
* Finisce la festa, il maître invia CLEAR e aspetta la fine del task. Poi RBR torna a casa.

In ogni istanza il maître può fare:

* CONSULT: consulta le risorse nel frigo, sul tavolo, in lavastoviglie e in dispensa. Ha senso aggiungere la possibilità che possa controllare anche ciò che c’è in viaggio, cioè che è in mano al robot? Sì, se no potrebbe essere sparito dal sistema e ciò non può accadere.
* STOP/REACTIVATE di qualsiasi task. RBR una volta fatto reactivate deve ricordarsi quello che stava facendo, quindi dobbiamo pensare a come memorizzare il suo goal corrente, il suo task e/o la sua destinazione. (Es. stava andando al tavolo per cosa? Per portare cibo? Per sparecchiare?)

RBR deve evitare ostacoli mobili come umani o animali nella stanza.

Il software sul frigo deve fare:

* EXPOSE: via COAP mostra il contenuto sullo smartphone del maître
* ANSWER: via COAP dice al maître o al robot se c’è un codice al suo interno

Come aggirare gli ostacoli mobili? HP grossolana: il robot va verso un ostacolo, quando è vicino all’ostacolo e ha superato una certa soglia, si ferma e aspetta. Se dopo aver aspettato un po’, questo non si muove, prova a girarci intorno (andrà a destra o a sinistra in base al goal).