

Generatore di interfacce per industria 4.0

MMI40

14 January 2020

Contents

1	Introduzione	1
2	Struttura attuale DM	2
3	UX design	2
3.1	Personas	2
3.2	Consigli	3
4	Wizard	3
4.1	Comandi	3
4.2	Esempio di processo di esecuzione	4
4.3	Consigli	4
5	Configurazione YAML	5
6	Struttura	5
6.1	Consigli	5
7	Lex Bot	5
8	Requisiti	5
9	Dizionario del progetto	6

1 Introduzione

L'obiettivo del progetto è la creazione di un framework che permetta a un utente "esperto in tecnologie" (dipendente Zerynth) di generare velocemente interfacce web per l'industria 4.0. L'output del progetto sarà una demo funzionante del sito della Pierucci Agricoltura s.r.l che nel seguito sarà indicata semplicemente con il nome **Pierucci**.

Il percorso di realizzazione di un sito utilizzando il framework MMI40 è il seguente (figura 1):

- il dipendente Zerynth interroga il committente del sito in modo da costruisce la UX per la commessa del sito.
- Il dipendente Zerynth utilizza il Wizard per produrre una configurazione YAML. Il Wizard utilizza una struttura precedentemente definita.
- La configurazione YAML viene data in input a un Lex Generator o a un React Generator che producono rispettivamente un Lex Bot o un React Site.

2 Struttura attuale DM

La DM ospita due tipi di dati dei device:

- **Time Series etichettate.** Per esempio, il device manda qualcosa del tipo: "all'ora X timestamp c'è una chiave che si chiama temperatura con una serie di valori in JSON" (*timestamp, Temp:{...}*). L'utente può anche decidere di organizzare i dati secondo una location: in questo caso, i vari device invieranno *timestamp, Stanza1:{ Temp:{...}, Umidità:{...}, ...}*. Le time series sono **molte**, sono **indicizzate sul tempo**, sono **immutabili**.
- **Stato del device. Sono documenti versionati. Esiste un documento per ogni device.** Per esempio, un device può essere collegato a un led il cui stato è ON oppure OFF. Per permettere il controllo del led, la DM mantiene uno stato con una variabile LED che può assumere valori ON/OFF. E' possibile richiedere l'ultimo documento, ma anche i documenti precedenti.

I due tipi di dati si visualizzano in modo diverso, si interrogano in modo diverso e avranno una UI e una UX diversa.

Lato UI, c'è poco. Si scrive una configurazione in YAML, in cui ad oggi abbiamo inserito la descrizione strutturale e stilistica della UI. Ma, come fanno HTML e CSS, sarebbe più consono **distinguere l'aspetto strutturale dall'aspetto stilistico**.

Un altro limite è il fatto che la configurazione YAML attuale permette di generare esclusivamente pagine statiche (nessuna interazione con l'utente, assenza di javascript, non ha come input variabili che possono configurare il risultato).

Per esempio, abbiamo tre dashboard sulla temperatura di tre stanze diverse (stanza1, stanza2, stanza3). Allo stato attuale è necessario creare tre configurazioni, una per ogni stanza. Quello che bisognerebbe fare è creare una singola configurazione, che prende in ingresso le tre stanze, e permette di generare una pagina per ogni stanza. **In altre parole bisognerebbe progettare configurazioni yaml parametriche.**

Quindi in teoria serve, una configurazione strutturale e una configurazione stilistica yaml. Queste configurazioni vengono date in input a una struttura di istanziazione con il quale generare le pagine.

3 UX design

3.1 Personas

L'unica personas del progetto è l'"esperto di tecnologie" che nel seguito verrà indicato con il nome "dipendente Zerynth".

3.2 Consigli

- Guardando la UX del sito Pierucci e immedesimandosi in un dipendente Zerynth è possibile capire i bisogni della personas che intendiamo modellare.

La UX del sito Pierucci non è ben fatta perchè il tool usato per crearlo è diventato uno strumento limitante. Per esempio, sul sito Pierucci tutto è **organizzato per dispositivi**. Ma ai clienti di Pierucci, che sono agricoltori e non tecnici, dovremmo offrire viste riassuntive (controllo crescite / controllo irrigazione) piuttosto che viste sulla sala1, sulla sala2 ecc. Per come era fatto il sistema precedentemente (senza etichette), la cosa più facile era creare una pagina in cui venivano mostrati tutti i dati di un dispositivo.

4 Wizard

4.1 Comandi

Per il Wizard abbiamo scelto di optare per comandi che consentano una costruzione molto veloce del progetto. Nella creazione di una pagina l'utente deve scegliere esclusivamente qual è il layout della pagina, il nome della pagina e infine gli elementi che popoleranno il layout della pagina. Nel seguito i parametri fra [] non sono obbligatori.

- `createProject(name,[configuration]);` crea un nuovo progetto di nome `name`. Se viene indicato il nome di una configurazione, e, tale configurazione esiste, viene creato un progetto non vuoto con le pagine che caratterizzano la configurazione scelta. In caso contrario...
 - `name`: nome del progetto
 - `configuration`: vengono fornite delle configurazioni precostruite che permettono di avere sin dall'inizio già uno scheletro del progetto che l'utente intende costruire. Le configurazioni sono associate a un identificatore alfanumerico. Di default questa proprietà è settata sul valore `BLANK` che significa progetto vuoto, quindi è presente solo un primo nodo radice.
- `newPage(name,layout);` crea una nuova pagina con il layout e il nome indicati.
 - `name`: nome della pagina
 - guarda layout di pagina nel dizionario.
- comandi per aggiungere elementi di pagina: `add+"nome elemento"`.
- Operazione di navigazione all'interno del progetto stile Unix (`cd`, `ls`, ecc...)

4.2 Esempio di processo di esecuzione

Nel seguito **Z** sta per dipendente Zerynth, **W** sta per Wizard:

1. **Z** apre **W**
2. **Z** scrive `createProject("name")` con il quale crea un nuovo progetto vuoto.
3. **Z** scrive `newPage("page1",login)` attraverso cui crea una nuova pagina con layout di pagina login.
4. **Z** scrive `newPage("home",home)` per generare una seconda pagina chiamata home con layout di pagina home.
5. **Z** si sposta in home eseguendo `cd home`.
6. **Z** scrive `addChart("Temperatura")`. Viene quindi aggiunto un elemento chart nel body della home.
7. **Z** scrive `addLink("Logout",api.logout)`. Viene aggiunto un link che consente di eseguire il logout nel menù laterale della home

4.3 Consigli

- Riduciamo il coding al minimo ma comunque produciamo delle configurazioni leggibili e modificabili. Perché facendo fare tutto al generatore poi precludiamo delle possibilità di customizzazione. Il generatore deve essere un **project generator** dove il dipendente indica la struttura del sito e della pagina. Se poi il dipendente necessita di ulteriori customizzazioni allora modifica il codice. Ricordiamoci che i nostri utenti sono programmatori. Il generatore deve essere uno strumento di supporto che riduce i tempi inutili (quelli in cui si fa sempre le stesse cose).
- Il Wizard potrebbe essere realizzato a linea di comando:
 - Si sceglie un linguaggio altamente espressivo e facile da capire che può essere ad esempio yaml.
 - In questo modo non ci sono problemi di UX e navigabilità.
 - Per esempio, una persona lancia il comando `createNewConfiguration (titolo,etichette)` e viene creata una prima configurazione YAML. Oppure, ho una riga di comando `get(Etichette)` che trova la lista delle etichette per il progetto corrente. Oppure `createNewMenu()` che crea un menu nella pagina corrente.

5 Configurazione YAML

La configurazione YAML consente di disaccoppiare il mondo della progettazione dal mondo dell'istanziamento. In questo modo con la medesima configurazione YAML, dovrebbe essere possibile creare generatori di interfacce di ogni tipo.

La configurazionebe YAML è un'insieme di file JSON relativamente dettagliate per permettere ai vari generatori di produrre un output adeguato.

6 Struttura

6.1 Consigli

- Guardando il sito Pierucci, si dovrebbe riuscire a decidere quali sono gli elementi base del framework. Per esempio, ci vuole un descrittore di pagina Main? Si potrebbe pensare a una configurazione Yaml che si chiama Main in cui è contenuta la sezione menu nella quale indico la struttura e il contenuto. Poi quando si clicca submit, vengono creati tanti Yaml per quante pagine sono state indicate nel menu. Questo è quello che si chiama **workflow di UX**.

Bisogna chiederci come pensa l'operatore Zerynth. Molto probabilmente la prima cosa che pensa è la struttura di cui ha bisogno. Non facciamolo partire dalla creazione dei file, ma bensì da un embrione dal quale produrre tutti gli altri contenuti.

- Trovare una struttura dove da ogni ramo e foglia possa corrispondere una parte del bot lex e una parte della interfaccia React. Per esempio, le viste possono essere viste come intenti (se sono in questa pagina è come se stessi iniziando una specifica conversazione con un bot), i grafici come gruppi di sentenze, i dati come etichette/slot.
- Bisognerebbe generare un project graph in cui ogni concetto è mappato in un intento del bot o in una pagina

7 Lex Bot

L'interazione via conversational system è solo un altro modo che ha l'utente di Pierucci per ottenere le informazioni che gli interessano, per esempio l'utente di Pierucci richiede vocalmente la temperatura di un macchinario in un determinato momento e il chatbot risponde

8 Requisiti

- Deve essere possibile definire la struttura (Navigation). Cosa c'è nella Root?

- Il Wizard deve avere due concetti:
 - Tipologia di pagina (Dashboard di grafici, Pagina di riepilogo ecc.)
 - Elementi che compongono la pagina
- Strutture di sito predefinito. Per esempio, una struttura standard potrebbe essere quella che contiene una pagina di LOGIN e una pagina MENU.

9 Dizionario del progetto

nome	significato	sinonimi
Elementi di pagina	Gli elementi di pagina sono gli elementi base di una pagina web quali per esempio bottoni, headers, ecc.	
Elementi di layout	Gli elementi di layout sono strutture che permettono di indicare come posizionare gli elementi di pagina.	
Layout di pagina	Struttura predefinita che codifica dove devono essere posizionati gli elementi di pagina all'interno di una pagina. Un layout di pagina è costruito a partire dalla composizione di elementi di layout.	
Pagina	Una pagina è caratterizzata da un nome e da un layout di pagina il quale è composto da uno o più elementi di layout e da diversi elementi di pagina.	
Configurazione	Una configurazione rappresenta la struttura del progetto in toto.	

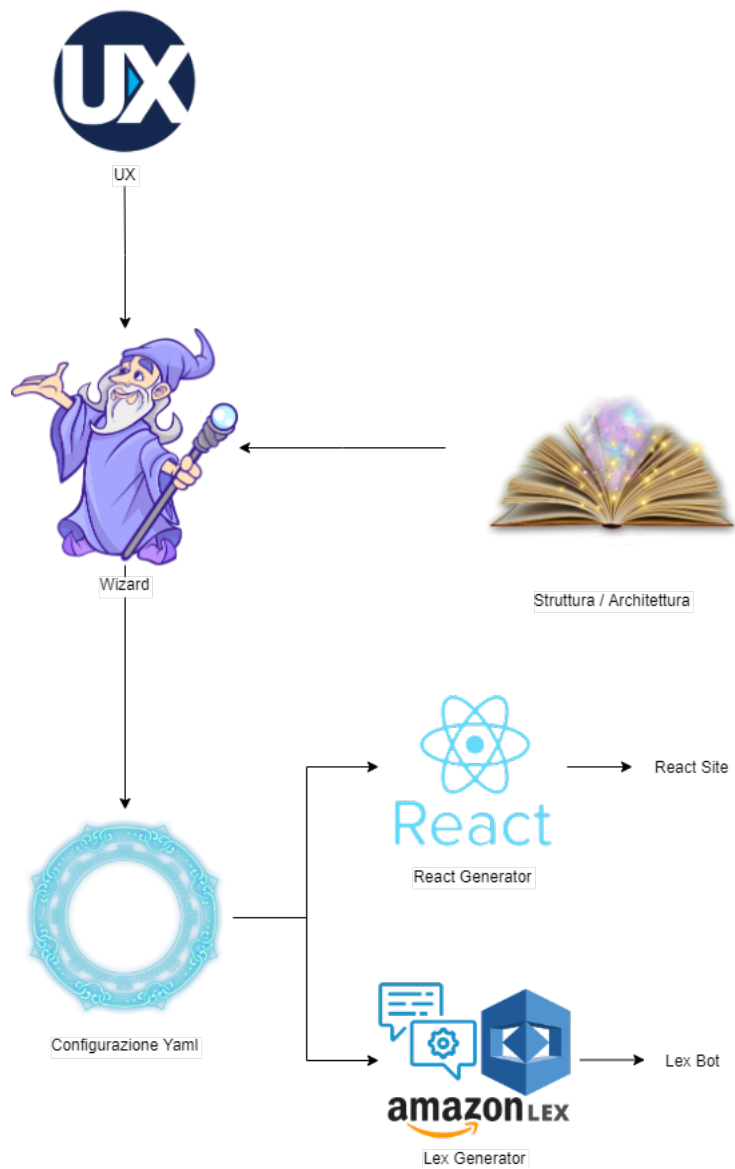


Figure 1: work flow generale per la creazione di un sito o di un bot utilizzando MMI40