

Patterns, Analisi di dominio e Robustness Diagram

Gregorio Piccoli
Padova, 5 Dicembre 2017

ZUCCHETTI SpA

Prima software house italiana, per storia e dimensione:

- 30 anni
- 400M Euro di fatturato
- 3500 persone
- 130000 clienti attivi
- Presenza in 50 paesi

... siamo una “software house”, non un “system integrator” ...

ZUCCHETTI SpA

Tre grandi divisioni responsabili dei prodotti:

1. Gestionali: contabilità, acquisti, vendite, magazzino, ...
2. Risorse umane: paghe, stipendi, presenze, controllo accessi, ...
3. Fiscale: dichiarazione dei redditi, 730, conservazione sostitutiva, ...

e con molti prodotti a contorno: Business Intelligence, robotica, IoT, sicurezza, ...

Facciamo applicazioni di Business!

ZUCCHETTI SpA

Fare applicativi
gestionali è ...

ZUCCHETTI SpA

Fare applicativi
gestionali è ...

NOIOSO!

PERCHE' IL GESTIONALE E' NOIOSO

- Le dimensioni dei prodotti sono enormi: decine di milioni di linee di codice, migliaia di tabelle
- Si rifanno sempre le stesse cose: innumerevoli gestioni di archivi, stampe, visualizzazioni, calcoli molto semplici, controlli di input
- Gli algoritmi sono sempre gli stessi: la partita doppia è stata inventata a Venezia nel 1500, l' MRP negli anni 50
- Bisogna essere tempestivi: specifiche dettate a pochissimo tempo dalla loro adozione
- Bisogna essere corretti: gli errori possono portare a danni economici per i nostri clienti

PERCHE' IL GESTIONALE E' NOIOSO

Quando ci troviamo di fronte ad un dominio “noioso” vuol dire che ci sono molte strutture che si ripetono e si assomigliano, quindi è il terreno dove ricercare quali sono i “patterns” ricorrenti.

DESIGN PATTERNS

Patterns del libro “Design Patterns”, di solito indicati con “GoF”:

1. Strutturali
2. Creazionali
3. Comportamentali

Molto legati alla realizzazione del codice.

Soluzioni di comprovata efficacia ad un problema che si presenta in modo ricorrente.

PATTERNS ARCHITETTURALI

Patterns non compresi nel “GoF” e di livello più vicino all’ architettura dell’ applicazione che non alla progettazione delle componenti:

- Data Access Object
- Active Data Object
- Object Relational Mapping
- Model, View, Controller
- Model, View, Presenter

ANALYSIS PATTERNS

I pattern di analisi appartengono all' "Analisi di Dominio", cioè all' esplorazione dell' insieme di tutti i problemi che potremo trovare in uno specifico campo alla ricerca dei tratti comuni.

Nelle procedure di business abbiamo sempre tre livelli:

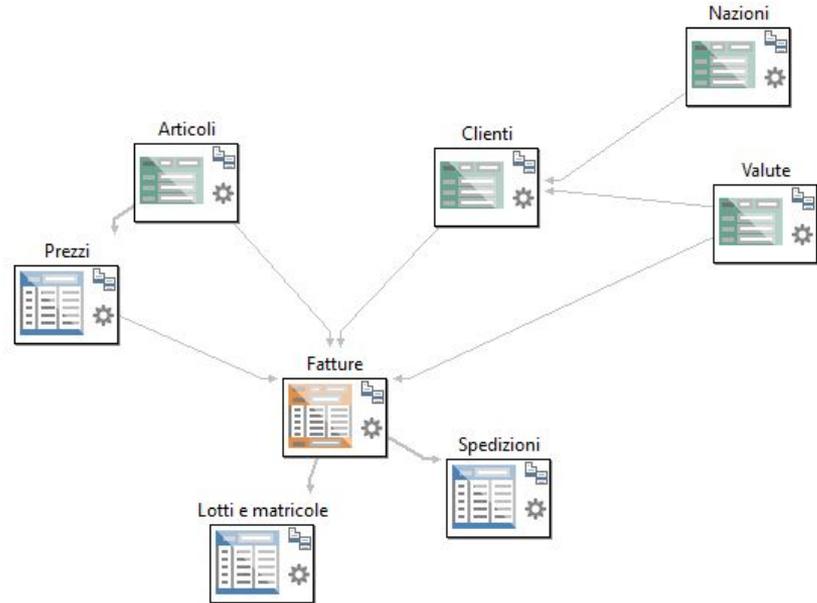
1. Archivio e transazione
2. Logica di business
3. Interfaccia utente

I patterns che abbiamo identificato operano trasversalmente a tutti tre i livelli e vengono utilizzati dall'analisi fino alla codifica passando per la progettazione.

ANALYSIS PATTERNS

Patterns principali:

1. Anagrafica (Master)
2. Movimentazione (Detail)
3. Master/Detail
4. Link
5. Padre/Figlio



ROBUSTNESS DIAGRAM

Abbiamo ottenuto una grande industrializzazione del processo produttivo dei programmi gestionali realizzando uno strumento di sviluppo che implementa questi pattern e guida lo sviluppatore dalla fase di analisi alla codifica.

Abbiamo un secondo strumento per realizzare le parti che non sono gestionali (portale, visualizzazioni, grafici, ...); non ha la stessa efficienza perché non si basa sui patterns di un dominio.

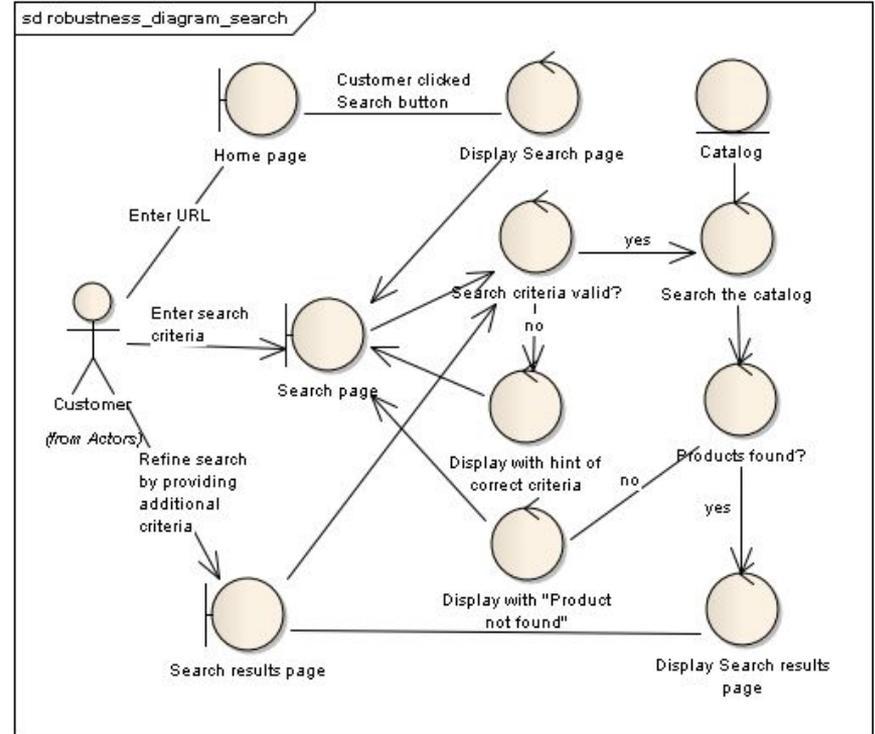
Proponendo il “IronWorks” ci siamo accorti che il “robustness diagram” è basato sui patterns identificabili con l’analisi di dominio più vasta possibile: l’intera informatica!



ROBUSTNESS DIAGRAM

Nel robustness diagram compaiono le interfacce con cui l'utente utilizzerà la procedura, i programmi che elaborano i dati e le entità che memorizzano i dati persistenti.

Tutti gli oggetti sono collegati tra loro per evidenziare quali sono le relazioni e le dipendenze.



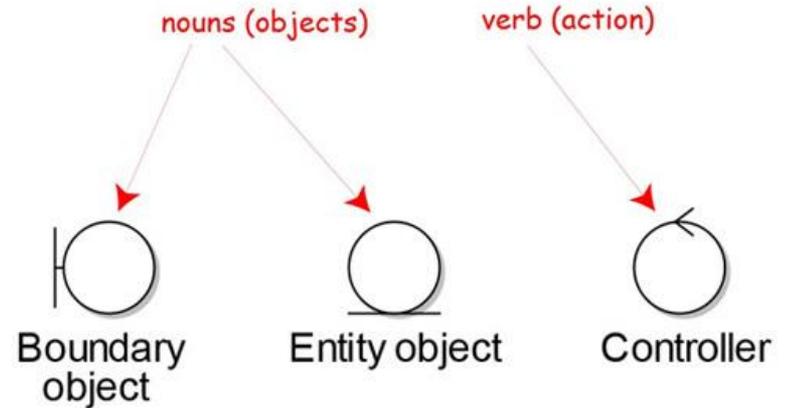
ROBUSTNESS DIAGRAM: GLI ELEMENTI

Boundary: le interfacce utente e anche le interfacce tra le procedure.

Entity: gli oggetti persistenti che sopravvivono alla singola computazione.

Controller: procedure che eseguono le computazioni e usano gli altri oggetti.

Actor: gli attori dei casi d'uso che interagiscono con le interfacce.

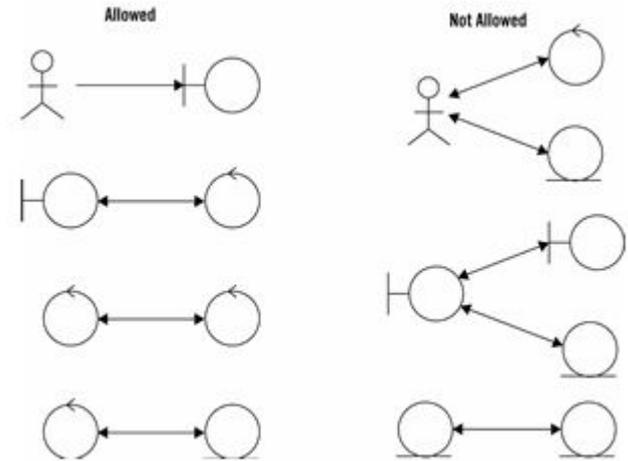


ROBUSTNESS DIAGRAM: LE REGOLE

Poche semplici regole:

1. L'attore interagisce con le interfacce
2. le interfacce interagiscono con i controller
3. i controller possono parlare tra loro
4. Le entità interagiscono con i controller

Nessun'altra connessione è permessa.



ROBUSTNESS DIAGRAM

Inserendo il “robustness diagram” nel nostro secondo strumento contiamo di limitare la “dispersione” dei metodi utilizzati dai singoli programmatori quando non sono guidati da pattern potenti e robusti.



ANTIPATTERNS DEL DOMINIO GESTIONALE

Ricette di comprovata inefficacia e che comportano problemi nel tempo:

1. Cut&Paste programming
2. “Quando va quanto basta non toccare che si guasta”
 - sindrome di Orietta Berti
3. Golden Hammer
4. Lava flow
5. Stovepipe
6. Functional decomposition
 - specialmente nel rapporto tra analisti e programmatori
 - si risolve con analisti funzionali e analisti tecnici



Domande?