

# Validazione basata su regole di documenti per il progetto Smart Publishing Management



Tesi presentata da: Tommaso Ognibene

Relatore: Chiar.mo Prof. Fabio Vitali

**Sessione:** I - 2015/2016



## Il progetto Smart Publishing Management

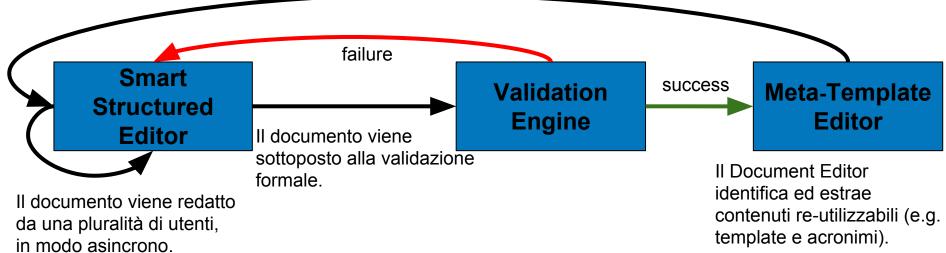
- Un sistema di gestione documentale per Alstom.
- Non un mero sistema informatico, ma un sistema informativo: definisce best practices e tre ruoli formali all'interno dell'organizzazione aziendale.
- Realizza i seguenti desiderata:
  - Un processo di produzione documentale univoco.
  - Un formato dati XML ad-hoc per la rappresentazione dei documenti.
  - Un ambiente di group editing online trusted.
  - Un sistema di templating e re-utilizzo di componenti documentali.
  - Un motore basato su regole per la validazione dei documenti.
- La tesi si compone di due parti:
  - Disamina dei requisiti teorici per l'intero progetto.
  - Presentazione del prototipo del motore basato su regole.



## 3 ruoli - 3 strumenti software

Ruolo	Strumento
Document Writer	Smart Structured Editor
Process Manager	Validation Engine
Document Editor	Meta-Template Editor

Nuovi contenuti estratti arricchiscono lo Smart Structured Editor.





</title>

</add>

## Requisiti teorici (1)

#### Temporal XML XBIT (XML-based Bitemporal Data Model) MXML (Multidimensional XML Model) **Overlapping Markup** add Milestones: CLIX e ECLIX **GODDAG** (General Ordered-Descendant Directed Acyclic Graph) Esempio: <add> title Modulo di Registrazione e Monitoraggio Variabili <title> Modulo di Registrazione <add> Cronologica <del> del add con Storicizzazione </add> degli Eventi </de|> Cronologica con Storicizzazione degli Eventi e Monitoraggio Variabili

Un esempio di GODDAG.



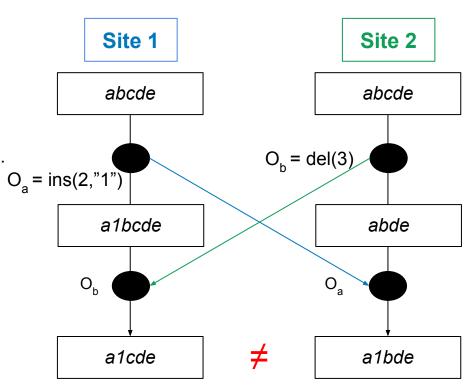
## Requisiti teorici (2)

#### Operational Transformation (OT)

- Un metodo per garantire consistency maintenance in group editors.
- Rispetto al locking, OT ottiene convergence e intention preservation senza il sacrificio della responsiveness e del concurrent work.
- OT è adottato dai principali strumenti di realtime collaborative editing, ad esempio Apache Wave e Google Docs.

#### Office Open XML (OOXML)

- Open Packaging Conventions.
- Standards ISO/IEC 29500:2008 and ECMA-376.



Il problema dell'*intention preservation*.



## II Validation Engine

#### Stack tecnologico:

Back-end: Node.js, Express (development framework), Passport.js (authentication middleware),

 Advance DB (development framework)

<u>MongoDB</u> (document-oriented database).

Front-end: <u>Jade</u> (template engine), <u>iQuery</u>.

Package manager: Bower, NPM.

Revision control system: Git.

#### Casi d'uso:

#### **Document Writer:**

- Carica un documento.
- Seleziona le regole.
- Esegui la validazione.
- Leggi e scarica i risultati.

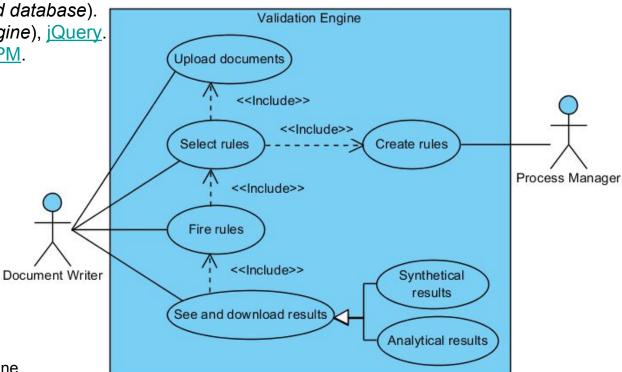
#### Process Manager:

Crea le regole.

#### Repository:

https://github.com/tomOgn/ValidationEngine

https://github.com/tomOgn/ValidationEngineUserManagement



**UML Use Case Diagram** 



## Le regole di validazione

- Due formati XML ad-hoc:
  - Collect-And-Check:
    - <collect>: identifica una lista di elementi del documento.
    - <check>: per ciascun elemento, verifica la sua conformità ad una regola sintattica.
  - Collect-And-Compare
    - <collect>: identifica due o più liste di elementi del documento.
    - <compare>: verifica la conformità delle liste tramite una regola di comparazione.
- Linguaggi sfruttati: Regex, XPath (1.0, 2.0), XSLT (1.0, 2.0).
- Esempio minimale:

- Esempio di maggiore complessità:
  - Verificare la corretta corrispondenza tra gli acronimi contenuti nel documento e gli acronimi definitivi nella relativa tabella degli acronimi.



## Schermate principali (0)



## Login Enter Username Password

Cancel

Submit



## Schermate principali (1)



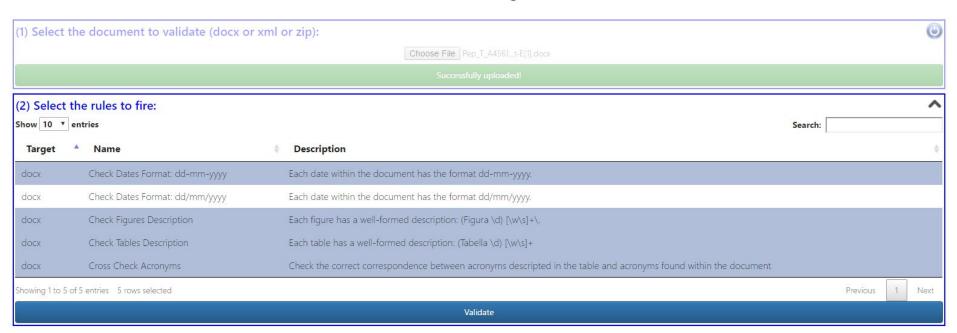




## Schermate principali (2)



Validation Engine





## Schermate principali (3)

## ALSTOM Validation Engine

(1) Select the document to validate (docx or xml or zip): Choose File | Pep T\_A4561\_t-E[1].docx (2) Select the rules to fire: Show 10 + entries Search: Target Name Description (3) View and download the results: Analytical View Synthetical View Show 10 ' entries Search: Document A Rule Matches Failed Pep\_T\_A456167it-E[1].docx Check Dates Format: dd-mm-yyyy 13 11 Pep\_T\_A456167it-E[1].docx Check Figures Description Pep\_T\_A456167it-E[1].docx Check Tables Description 26 17 307 Pep\_T\_A456167it-E[1].docx Cross Check Acronyms 343 Showing 1 to 4 of 4 entries Previous Download Synthetical View Download Analytical View



## Conclusioni

#### Prosecuzione del lavoro:

- Completamento del Validation Engine:
  - o aggiungere il supporto al formato XLSX per i documenti.
  - o aggiungere il supporto ai controlli inter-documentali.
  - o aggiungere il supporto al linguaggio XQuery all'interno delle regole.
- Sviluppare gli altri due strumenti: Smart Structured Editor e Meta-Template Editor.

#### Considerazioni personali:

- Relativamente a Node is come linguaggio server-side:
  - o Ambiente open-source, cross-platform.
  - Architettura event-driven.
- Relativamente a XML come linguaggio per rappresentare documenti e regole:
  - Copiosa letteratura scientifica.
  - Ampia versatilità ma limitata capacità espressiva.
  - Linguaggi di interrogazione e manipolazione potenti ma tuttora poco diffusi.
- Relativamente alla progettazione di un sistema informatico complesso:
  - o Capacità di analisi dei desiderata del cliente.
  - o Identificazione dei casi d'uso e delle *feature* da realizzare.
  - Definizione della migliore architettura software possibile.
  - o Organizzazione del lavoro in termini di fasi esecutive, *deployment* intermedi, ripartizione dei compiti e dei ruoli.