Architettura di xmLegesEditor

- *xmLegesEditor* è strutturato secondo un'architettura a *componenti*
- implementazione del pattern IOC “Inversion of Control” (Direct Injection) http://www.martinfowler.com/articles/injection.html
- ServiceManager (semplificato) ispirato al framework *Apache Avalon* implementato nel package `it.cnr.ittig.services.manager` vedi: http://excalibur.apache.org/framework/index.html per documentazione sul framework di Apache
- Implementa un set ristretto di interfacce; mantiene la struttura
- L'applicazione si “costruisce” all'avvio istanziando un insieme di componenti specificati da un file di composizione (`xmLegesEditor.xml`) letto dal ServiceManager
- main class: `it.cnr.ittig.services.manager.Run`

  arguments : `xmLegesEditor.xml /images/editor/xmLegesEditor.png`

Lifecycle dei Componenti

- Il *lifecycle* di un componente specifica i metodi che possono essere chiamati su di esso e l'ordine in cui possono essere chiamati.
- Ogni componente espone i suoi *lifecycle methods* implementando le *lifecycle interfaces*. Le interfacce definite in `it.cnr.ittig.services.manager` sono:

Serviceable	(il componente può richiedere altri servizi)
Loggable	(al componente viene passato un Logger)
Configurable	(il componente può essere configurato)
Initializable	(il componente può eseguire del codice in fase di inizializzazione)
Startable	(il componente può eseguire del codice in fase di *start (dopo* initialize*)* o *stop (prima* di dispose*))*
Disposable	(il componente può eseguire del codice in fase di *dispose* (shutdown o crash dell'applicazione))

Lifestyle dei Componenti

ACTIVATION: “startup” o “lazy” [DEFAULT: “lazy”]

startup; componente istanziato all'avvio dell'applicazione

lazy: componente istanziato su richiesta di un altro componente

LIFESTYLE: “singleton” o “pool” [DEFAULT: “singleton”]

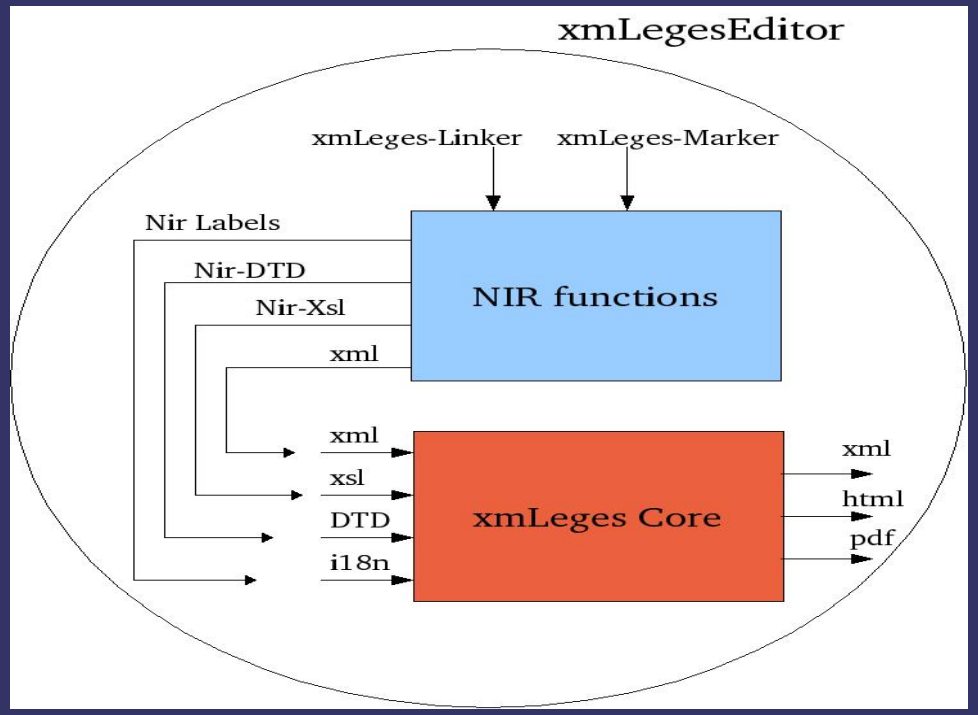
singleton: è ammessa una sola istanza del componente

pool: ad ogni richiesta del servizio viene creata una nuova istanza

```
<component  activation="startup" lifestyle="singleton"
  class="it.cnr.ittig.xmleges.core.blocks.lookandfeel.LookAndFeelImpl" >
    <configuration>
      <lookandfeel>
            com.jgoodies.plaf.plastic.Plastic3DLookAndFeel
      </lookandfeel>
    </configuration>
 </component>
```

Divisione dei Package

- xmLegesCore
  - *xmLegesCoreApi*
  - *xmLegesCoreImpl*
- xmLegesEditor
  - *xmLegesEditorApi*
  - *xmLegesEditorImpl*



xmLegesCore: Interfacce ed implementazioni delle funzionalità generiche di Editor XML Visuale

```
it.cnr.ittig.xmleges.core.services.*     (API)
it.cnr.ittig.xmleges.core.blocks.*       (IMPL)
```

xmLegesEditor: Interfacce ed implementazioni delle funzionalità specifiche di Editor Legislativo NIR

```
it.cnr.ittig.xmleges.editor.services.*     (API)
it.cnr.ittig.xmleges.editor.blocks.*       (IMPL)
```

Principali Componenti di: xmLegesCore 1/4

it.cnr.ittig.xmleges.core.services.

- *EventManager*: servizio per la gestione degli eventi
- *ActionManager*: servizio per la gestione delle azioni
- *Bars:* servizio per la creazione delle barre dei menu, degli strumenti e di stato.

```
<configuration>
    <menus>
        <menu action="menu.file">
        <item action="file.new" />
            ....
        </menu>
    </menus>
     <tools>
        <toolbar name="file" floatable="false" rollover="true">
        <item action="file.new" />
        .......
        </toolbar>
    </tools>
<configuration>
```

Principali Componenti di: xmLegesCore 2/4

it.cnr.ittig.xmleges.core.services.

- *DocumentManager:* servizio per la gestione del documento: apertura, validazione, chiusura, gestione transazioni, undo/redo
- *SelectionManager:* servizio per la gestione delle selezioni dei nodi del documento e del testo all'interno di nodi di tipo testo.
- *I18n* servizio per la gestione dell'internazionalizzazione delle stringhe, delle icone, delle immagini e degli oggetti.
- *DtdRulesManager* servizio per che gestisce le regole scritte nella DTD di un documento XML: legge una DTD e trasforma il content di ogni elemento in un automa a stati finiti.

Principali Componenti di: xmLegesCore 3/4

it.cnr.ittig.xmleges.core.services.

- *Frame*: servizio per la gestione della finestra principale dell'applicazione. Ogni pannello di modifica che intende essere visualizzato deve registrarsi sul frame (CONFIGURABLE – dislocazione dei pannelli sul frame)

```
<configuration>
    <title>xmLegesEditor</title>
        <!-- where=top-center | top-left | bottom-center | bottom-left -->
        <!-- index=numero -->
```

- *XsltMapper:* servizio per effettuare la mappatura bidirezionale tra nodi DOM e HTML per la visualizzazione e l'editing su pannelli XsltPane.

- *XsltPane:* servizio per la gestione di un pannello di testo per la modifica di un documento XML. Il pannello presenta il testo in formato HTML tramite un foglio di trasformazione XSLT. La modifica apportata al testo presentato è automaticamente validato e riportato sul DOM del documento

Principali Componenti di: xmLegesCore 4/4

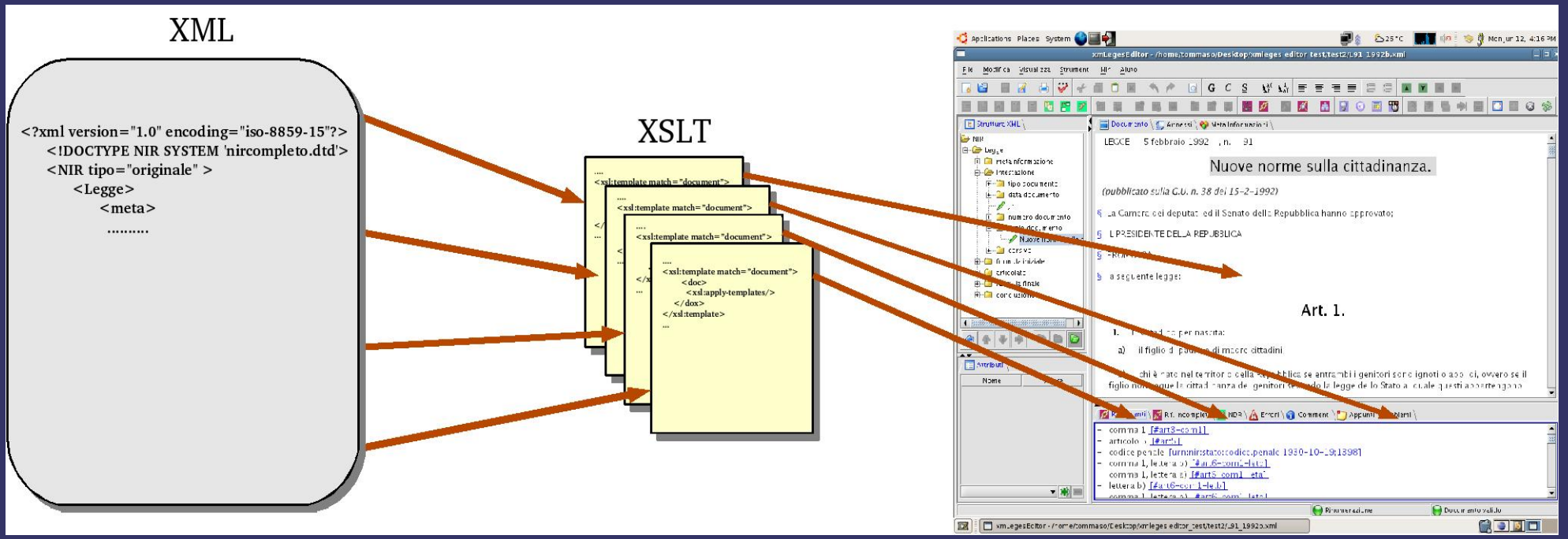
it.cnr.ittig.xmleges.core.services.

- *AttributesPane*: servizio per la visualizzazione e la modifica degli attributi di un nodo.
- *TreePane:* servizio per visualizzare il documento DOM sottoforma di albero
- *Form*: servizio per la costruzione di una form e visualizzazione in una finestra di dialogo.

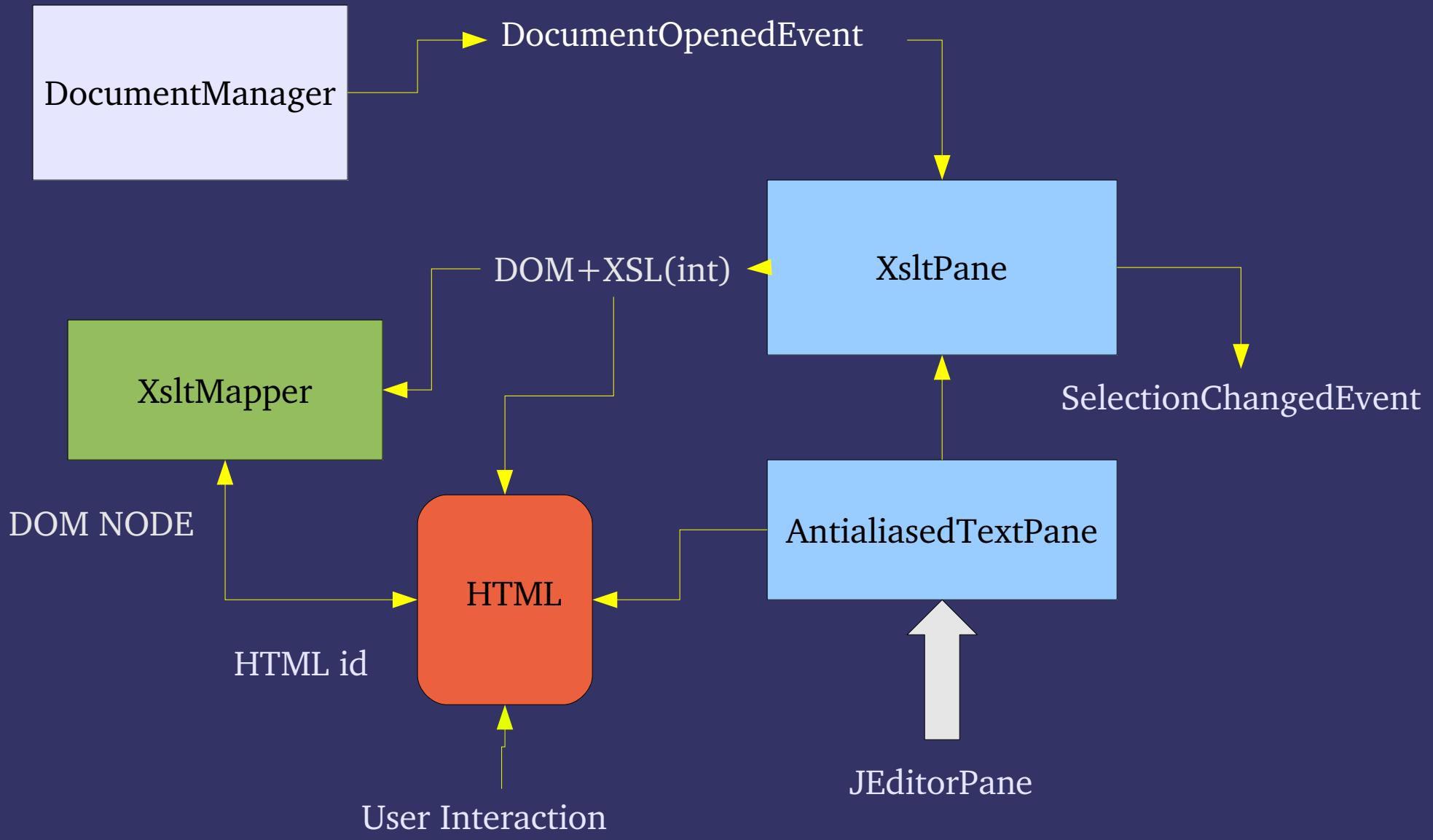
Tipologie di Eventi scambiati tramite EventManager

ActionRegisteredEvent
DocumentChangedEvent
DocumentClosedEvent
DocumentOpenedEvent
DocumentSavedEvent
PaneActivatedEvent
PaneDeactivatedEvent
PaneFocusGainedEvent
PaneFocusLostEvent
PaneStatusChangedEvent
SelectionChangedEvent
ExecFinishedEvent
ExecStartedEvent

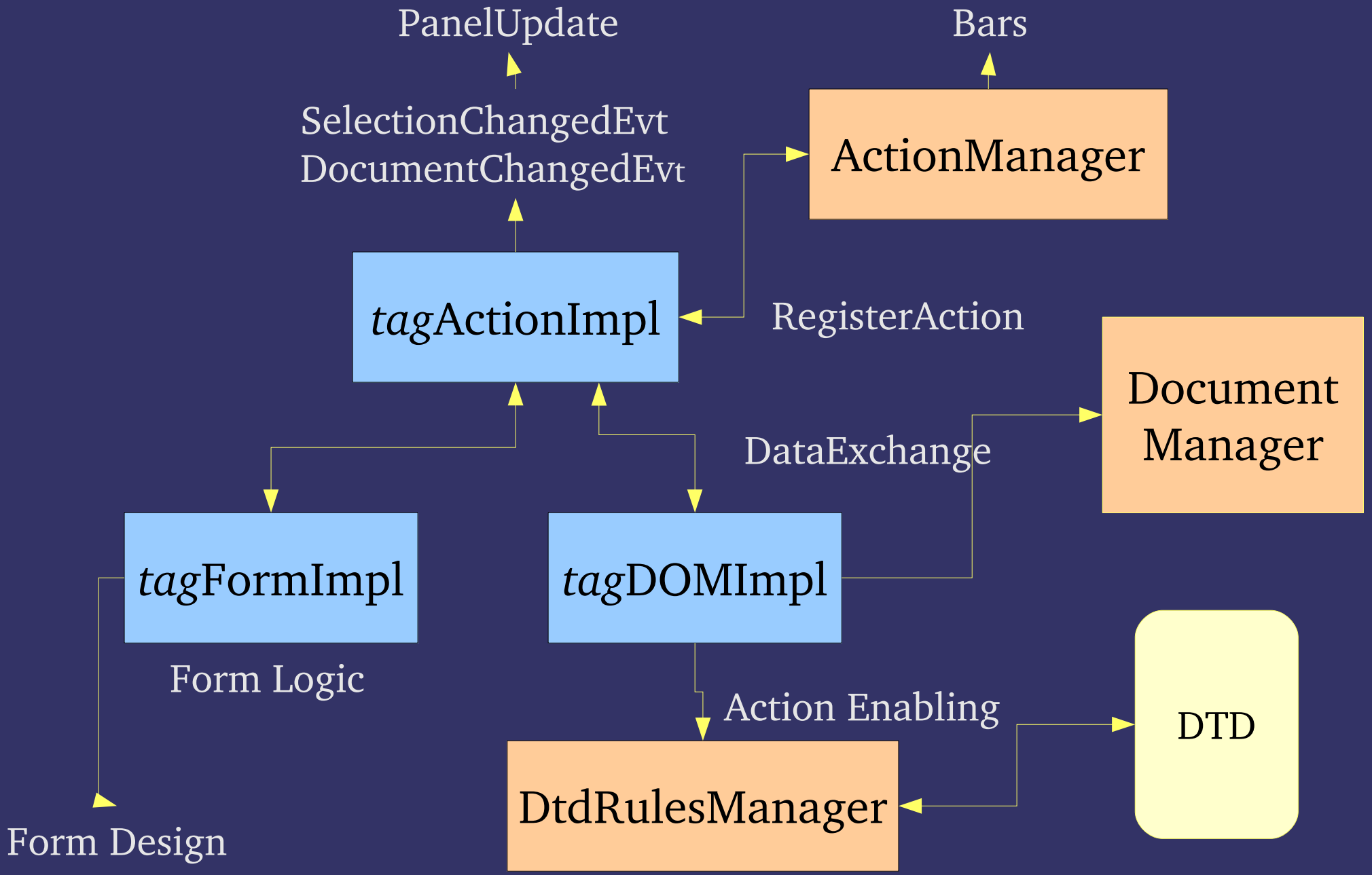
Gestione dei Pannelli di editing



Gestione dei Pannelli di editing



Struttura di una Funzione DOM Generica



UI Design

Design delle form realizzato con Abeille Forms Designer

dalla documentazione:

*"Abeille Forms Designer is a tool for quickly designing cross-platform graphical user interfaces. Abeille is a Swing based application and uses Swing components. Abeille depends on the **JGoodies FormLayout Manager***
"

Tool OpenSource: vedi https://abeille.dev.java.net/

I componenti *tag*FormImpl richiedono il componente Form, recuperano i componenti Swing dalla Form e ne implementano la logica

L'implementazione richiede la libreria `formsrt.jar` per la lettura dei file `.jfrm` generati da Abeille

Principali Componenti di: xmLegesEditor 1/3

it.cnr.ittig.xmleges.editor.services.

- *NirXsltPanes*: servizio per la gestione dei pannelli Xsl di editing

```
<configuration>
    <pane name="editor.panes.documento">
        <xslt name="edit-documento"/>
    </pane>
    <pane name="editor.panes.annessi">
        <xslt name="edit-annessi"/>
    </pane>
    .....
```

- *NirXslts*: servizio per la gestione dei file di trasformazione XSLT e fogli di stile CSS necessari per i pannelli di modifica del testo

Principali Componenti di: xmLegesEditor 2/3

it.cnr.ittig.xmleges.editor.services.

- *Provvedimenti:* Servizio per la lettura da file dei tipi di provvedimento gestibili
- *Template:* Servizio per la gestione dei template per i nuovi documenti o gli allegati.
- *UtilUrn:* Servizio per la fornitura di utilità specifiche per lo standard URN Nir per la rappresentazione dei riferimenti normativi.
- *NirUtilDom:* Servizio per la fornitura di utilità per funzioni DOM specifiche per lo standard NIR

Principali Componenti di: xmLegesEditor 3/3

Funzionalità di supporto alla gestione dei tag delle DTD NIR

*it.cnr.ittig.xmleges.editor.services.action.**
*it.cnr.ittig.xmleges.editor.services.form.**
*it.cnr.ittig.xmleges.editor.services.dom.**

annessi(Action,Form,Dom)
autorita
fileExport
fileNew
link
liste
meta
ndr
partizioni
rinumerazione
rinvii
rifIncompleti
spostamento
tabelle
testo
vigenza
linker
revisioni
marker

validazione “a priori”

- l'abilitazione delle action su menù e toolbar è condizionata all'interrogazione del DtdRulesManager per verificare “a priori” se l'operazione richiesta darà luogo o meno a un documento valido secondo la DTD caricata. Altrimenti i pulsanti sono disabilitati.
- nelle funzioni DOM ogni operazione sull'albero DOM (inserimento, append, cancellazione) del Documento viene eseguita a condizione che sia dichiarata ammissibile dal DtdRulesManager
- il DtdRulesManager fornisce funzionalità per creare il minimo template valido per un tag (elemento e figli ed attributi obbligatori)

-> risultato: partendo da documenti validi preesistenti o da template, non è possibile creare un documento non valido

Non è richiesta alcuna validazione a posteriori del documento;
Funzionalità di validazione XML trasparente per l'utente

stato del Progetto

- Trac su

https://svn.ittig.cnr.it/trac/xmLegesEditor

- migrazione Java_1.5 o 1.6 (attualmente ostacolata da variazioni non documentate su HtmlReader di JeditorPane per Java > 1.4)
- miglioramento efficienza, usabilità e debugging
- estensione del supporto a XML-Schema tramite lo sviluppo di un componente SchemaRulesManager
- per Senato: supporto a nuova DTD per la rappresentazione dei Disegni di Legge (attualmente dllight.dtd molto ristretta)
- integrazione di nuove funzionalià per il supporto dei metadati analitici previsti dalle DTD NIR
- ampliamento della documentazione