

PROGETTAZIONE

NICOLA MORETTO (MATR. 578258)

24 ottobre 2012

Il documento riporta le informazioni di progettazione riguardanti l'interfaccia grafica per la visualizzazione e la navigazione dei contenuti.

VERSIONE	DATA	MODIFICHE
0.1	15-10-2012	Stesura iniziale del documento.
0.2	17-10-2012	Redatto il capitolo 2.
0.3	18-10-2012	Redatte le sezioni 3.4, 4.3 e 4.4.
0.4	19-10-2012	Redatti i capitoli 3 e 4.
0.5	20-10-2012	Revisione del documento.
1.0	20-10-2012	Pubblicazione della prima versione.
1.1	22-10-2012	Aggiunta la sezione 3.5.
1.2	23-10-2012	Aggiornamento del capitolo 3.
1.3	24-10-2012	Aggiornamento del capitolo 3.

Tabella 1: Registro delle modifiche

INDICE

1	INTRODUZIONE	6
1.1	Riferimenti informativi	6
2	ARCHITETTURA	7
2.1	Architettura generale	7
2.2	Componenti architetturali	7
2.2.1	Componente Model	7
2.2.2	Componente View	7
2.2.3	Componente Controller	8
3	COMPONENTE MODEL	9
3.1	Package model	9
3.1.1	Entity	9
3.1.2	Label	9
3.1.3	Meaning	9
3.2	Package model.content	9
3.2.1	Content	9
3.2.2	ContentManager	10
3.2.3	Answer	10
3.2.4	Event	10
3.2.5	Message	10
3.2.6	Question	10
3.2.7	Review	10
3.2.8	Talk	10
3.2.9	Thought	10
3.3	Package model.criteria	10
3.3.1	Criterion	10
3.3.2	ListCriterion	11
3.3.3	RangeCriterion	11
3.3.4	SwitchCriterion	11
3.3.5	ValueCriterion	11
3.3.6	CriterionData	11
3.3.7	Emotion	11
3.3.8	Intention	11
3.3.9	PublicationDate	11
3.3.10	Rating	11
3.3.11	Relevance	11
3.3.12	Topic	11
3.3.13	Type	11
3.3.14	User	11
3.4	Package model.filter	11
3.4.1	Filter	12
3.4.2	FilterManager	12
3.4.3	ListFilter	12
3.4.4	RangeFilter	12
3.4.5	SwitchFilter	13
3.4.6	ValueFilter	13
3.5	Package model.provider	13
3.5.1	Provider	13

3.5.2	ProviderRegistry	13
3.5.3	LabelProvider	13
3.5.4	FullTextProvider	13
3.6	Package model.search	14
3.6.1	SearchEntityList	14
3.6.2	SearchManager	14
3.6.3	SearchQuery	14
3.6.4	SearchResult	14
3.6.5	SearchScope	14
3.6.6	SearchTerm	14
3.7	Package model.timeline	14
3.7.1	Timeline	14
4	COMPONENTE VIEW	15
4.1	Package view	15
4.1.1	MainWindow	15
4.2	Package view.content	15
4.2.1	ContentView	15
4.3	Package view.filter	16
4.3.1	FilterContainer	16
4.3.2	FilterView	16
4.3.3	ListFilterView	17
4.3.4	RangeFilterView	17
4.3.5	SwitchFilterView	17
4.3.6	ValueFilterView	17
4.4	Package view.search	17
4.4.1	EntityList	17
4.4.2	EntityView	17
4.4.3	SearchBar	17
4.4.4	SearchScopeSelector	17
4.4.5	LabelEntityList	18
4.5	Package view.timeline	18
4.5.1	TimelineView	18
4.5.2	TimeSlot	18
4.5.3	TimeAxis	18
5	COMPONENTE CONTROLLER	19
5.1	Package controller	19
5.2	Package controller.content	19
5.3	Package controller.filter	19
5.4	Package controller.search	19
5.5	Package controller.timeline	19

INTRODUZIONE

1.1 RIFERIMENTI INFORMATIVI

- Analisi dei requisiti (*analisi_dei_requisiti_1.0* allegata alla presente documentazione);
- Sistema di classificazione (*sistema_di_classificazione_2.0* allegato alla presente documentazione).

ARCHITETTURA

2.1 ARCHITETTURA GENERALE

L'architettura del sistema software rispecchia il design pattern architetturale MVC, che prevede e garantisce la separazione delle tre componenti fondamentali del sistema:

MODEL Racchiude i dati e le informazioni dell'applicazione e definisce le modalità di accesso e fruizione degli stessi da parte delle altre componenti (*Controller* e *View*).

VIEW Rappresenta l'interfaccia grafica mediante la quale vengono visualizzate le informazioni e i dati conservati nel *Model* e l'utente può interagire con il sistema. La rilevazione dell'avvenuta interazione dell'utente è responsabilità di tale componente, mentre la gestione della reazione è demandata al *Controller*.

CONTROLLER Incorpora la logica di controllo dell'applicazione, inizializzando il sistema e traducendo l'interazione dell'utente con l'interfaccia grafica (*View*) in operazioni sui dati (*Model*).

2.2 COMPONENTI ARCHITETTURALI

2.2.1 Componente Model

La componente *model* conserva tutte i tipi di informazioni connessi alla ricerca:

- gli ambiti (etichette, frasi);
- i criteri (termini di ricerca, entità);
- i filtri (argomento, data di pubblicazione, emozioni, ...);
- i risultati (contenuti informativi).

2.2.2 Componente View

La componente *view* rappresenta l'interfaccia grafica mediante la quale l'utente interagisce con il sistema per effettuare una ricerca, raffinarne i criteri o consultarne i risultati.

LIVELLI Essa è organizzata in quattro livelli (o strati) distinti, che includono rispettivamente:

- gli strumenti e le informazioni connesse alla ricerca (barra di ricerca, etichette, entità, ...);
- i filtri di ricerca;
- i risultati della ricerca (proprietà e relazioni dei contenuti informativi);
- la cronologia (organizzazione temporale dei contenuti).

2.2.3 *Componente Controller*

La componente *controller* gestisce l'interazione dell'utente con l'interfaccia grafica e le operazioni connesse alla ricerca e alla visualizzazione dei contenuti informativi, tra cui:

- la configurazione dei criteri di ricerca (selezione di un'accezione di un'etichetta, gestione delle entità);
- il reperimento dei contenuti informativi corrispondenti ai criteri di ricerca;
- la gestione dei filtri di ricerca;
- l'aggiornamento dei risultati di ricerca visualizzati a fronte di modifiche alle entità e ai filtri di ricerca;
- ...

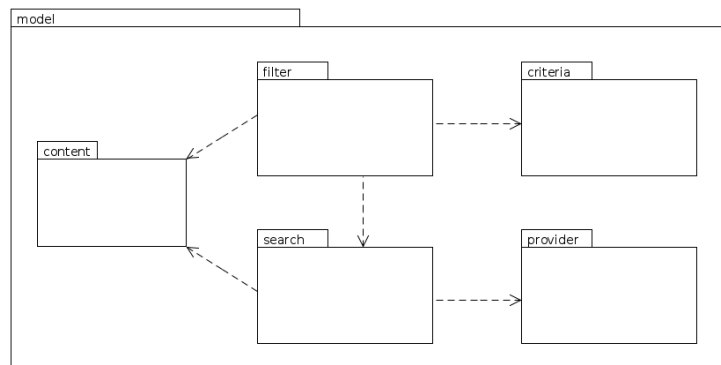


Figura 1: Diagramma del package *model*

Questo capitolo illustra i componenti del *Model*, per ciascuno dei quali è indicato il nome della classe accompagnata da un'identificazione sintetica, separati dal carattere '|' (separatore verticale).

3.1 PACKAGE MODEL

3.1.1 *Entity* | *Entità del dominio*

La classe modella le entità del dominio della piattaforma, a ciascuna delle quali è associata un'etichetta - in una specifica accezione - che la identifica univocamente nell'ambito della piattaforma (*Label*).

3.1.2 *Label* | *Etichetta del dizionario*

La classe modella le etichette del dizionario della piattaforma, ciascuna delle quali può possedere molteplici accezioni (*Meaning*).

3.1.3 *Meaning* | *Accezione di un'etichetta*

La classe modella un'accezione di un'etichetta (*Label*), cui è associata un'entità del dominio (*Entity*).

3.2 PACKAGE MODEL.CONTENT

3.2.1 *Content* | *Contenuto informativo*

La classe astratta rappresenta un generico contenuto informativo pubblicato dagli utenti nella piattaforma, caratterizzato dalle seguenti proprietà:

- autore (*User*);

- data di pubblicazione (*PublicationDate*);
- titolo.

3.2.2 *ContentManager* | *Gestione dei contenuti*

Interfaccia del package *model.content* per la gestione dei contenuti (*Content*) istanziati.

3.2.3 *Answer* | *Risposta*

DIPENDENZE Estende la classe *Content*.

3.2.4 *Event* | *Evento*

DIPENDENZE Estende la classe *Content*.

3.2.5 *Message* | *Comunicazione privata*

DIPENDENZE Estende la classe *Content*.

3.2.6 *Question* | *Domanda*

DIPENDENZE Estende la classe *Content*.

3.2.7 *Review* | *Recensione*

DIPENDENZE Estende la classe *Content*.

3.2.8 *Talk* | *Discorso*

DIPENDENZE Estende la classe *Content*.

3.2.9 *Thought* | *Pensiero*

DIPENDENZE Estende la classe *Content*.

3.3 PACKAGE MODEL.CRITERIA

Ciascun contenuto può essere classificato in accordo a differenti criteri, ciascuno dei quali ne prende in esame una differente proprietà, come l'autore, la data di pubblicazione, il tipo, le emozioni associate,

Ciascuna proprietà può essere propria del contenuto o funzionale al contesto di ricerca (attinenza, ...).

3.3.1 *Criterion* | *Criterio di classificazione*

Interfaccia dei (tipi di) criteri di classificazione.

3.3.2 *ListCriterion* | Criterio basato su una lista di valori

La classe *ListCriterion* viene utilizzata in combinazione con un filtro *ListFilter*, fornendo i valori e i parametri di configurazione iniziale.

3.3.3 *RangeCriterion* | Criterio basati su un intervallo di valori

La classe *RangeCriterion* viene utilizzata in combinazione con un filtro *RangeFilter*, fornendo i valori e i parametri di configurazione iniziale.

3.3.4 *SwitchCriterion* | Criterio a doppio stato

La classe *SwitchCriterion* viene utilizzata in combinazione con un filtro *SwitchFilter*, fornendo i valori e i parametri di configurazione iniziale.

3.3.5 *ValueCriterion* | Filtro basato su una soglia di valore

La classe *ValueCriterion* viene utilizzata in combinazione con un filtro *ValueFilter*, fornendo i valori e i parametri di configurazione iniziale.

3.3.6 *CriterionData* | Sorgente dati per un criterio di classificazione

Classe astratta dei componenti che forniscono liste di valori per i criteri di classificazione.

3.3.7 *Emotion* | Emozione

3.3.8 *Intention* | Intenzione

3.3.9 *PublicationDate* | Data di pubblicazione

3.3.10 *Rating* | Giudizio

3.3.11 *Relevance* | Attinenza

3.3.12 *Topic* | Argomento

3.3.13 *Type* | Tipo di contenuto

3.3.14 *User* | Utente

La classe rappresenta un utente della piattaforma, potenziale autore di contenuti.

3.4 PACKAGE MODEL.FILTER

Ciascun filtro è associato ad una proprietà dei risultati di ricerca (*Criterion*) e partiziona automaticamente l'insieme dei possibili valori in due sottoinsiemi: ammessi o bloccati. Nella configurazione iniziale, tutti i valori possibili di una proprietà sono ammessi, mentre l'utente può intervenire secondo modalità differenti per alterare tale partizionamento (vedi sezione 4.3).

Il valore che ciascun contenuto (da intendersi come risultato di una ricerca) ha rispetto ad una proprietà associata ad un filtro può quindi appartenere ad uno dei due sottoinsiemi: il contenuto viene mostrato solo se tutti i valori delle proprietà in questione risultano ammessi.

3.4.1 *Filter* | *Filtro di ricerca*

Tale componente rappresenta l'interfaccia dei filtri per il raffinamento dei risultati di una ricerca e viene implementata dalle classi che modellano le tipologie standard di filtri di ricerca (*ListFilter*, *RangeFilter*, *SwitchFilter* e *ValueFilter*).

Ciascuna istanza di una sottoclasse concreta è associata ai risultati di una ricerca specifica (*SearchResult*), su cui si applica il filtro corrispondente.

3.4.2 *FilterManager* | *Gestione dei filtri*

Tale componente rappresenta l'interfaccia del package *model.filter*, utile a esporre le funzionalità per l'istanziamento e la gestione dei filtri.

3.4.3 *ListFilter* | *Filtro a lista di valori*

Il componente modella un filtro basato su una lista di possibili valori, ciascuno dei quali può essere autorizzato o bloccato dall'utente.

DIPENDENZE: Ogni istanza della classe *ListFilter* è associata ad un criterio *ListCriterion*.

3.4.4 *RangeFilter* | *Filtro ad intervallo di valori*

Il componente modella un filtro basato su un intervallo di valori ordinati (numeri, date, ...). Siano *inf* e *sup* rispettivamente l'estremo inferiore e superiore dell'intervallo: risultano dunque ammessi tutti e soli i valori *v* tali che $v \in [inf, sup]$.

All'utente è consentito scegliere i valori desiderati di *inf* e *sup* tali che:

- *sup* sia un valore valido;
- *inf* sia un valore valido;
- $inf \leq sup$;
- se è definito un valore attuale *current*, allora deve valere $min \leq current \leq max$.

Se l'insieme dei valori ordinati prevede un minimo *min* e/o un massimo *max* si aggiungono le seguenti condizioni:

- $sup \leq max$;
- $inf \geq min$.

DIPENDENZE: Ogni istanza della classe *RangeFilter* è associata ad un criterio *RangeCriterion*.

3.4.5 *SwitchFilter* | Filtro a doppio stato

Il componente modella un filtro il cui partizionamento è predefinito e invariabile e che l'utente può solamente abilitare o disattivare.

DIPENDENZE: Ogni istanza della classe *SwitchFilter* è associata ad un criterio *SwitchCriterion*.

3.4.6 *ValueFilter* | Filtro a soglia di valore

Il componente modella un filtro basato su una soglia di valore. Sia *value* il valore scelto: in tal caso risultano ammessi tutti e soli i valori x validi tali che $x \geq value$.

Se l'insieme prevede un minimo *min* e/o un massimo *max* dev'essere soddisfatta anche la condizione $min \leq x \leq max$.

DIPENDENZE: Ogni istanza della classe *ValueFilter* è associata ad un criterio *ValueCriterion*.

3.5 PACKAGE MODEL.PROVIDER

Le classi del package *model.provider* forniscono le funzionalità necessarie ad interrogare la base di dati per reperire i contenuti informativi corrispondenti ai criteri di ricerca immessi e all'ambito specificato.

Dal momento che tali operazioni verranno affidate a componenti terze, tuttora oggetto di analisi e valutazione, le scelte progettuali illustrate di seguito tengono conto di tale incertezza e sono pertanto da considerarsi non esaustive o definitive.

3.5.1 *Provider* | Motore di ricerca

La componente rappresenta l'interfaccia standard implementata dai fornitori di ricerca (*LabelProvider*, *FullTextProvider*).

3.5.2 *ProviderRegistry* | Gestione dei fornitori di ricerca

La componente gestisce le istanze delle classi *LabelProvider* e *FullTextProvider*.

3.5.3 *LabelProvider* | Ricerca delle etichette

La classe fornisce le funzionalità necessaria a cercare dei termini di ricerca tra le etichette assegnate ai contenuti, considerandone le specifiche accezioni.

DIPENDENZE Implementa l'interfaccia *Provider* e le sue istanze vengono gestite da *ProviderRegistry*.

3.5.4 *FullTextProvider* | Ricerca dei contenuti informativi

La classe permette di cercare la presenza dei termini di ricerca nel titolo, nel corpo o in altri campi di un contenuto informativo.

DIPENDENZE Implementa l'interfaccia *Provider* e le sue istanze vengono gestite da *ProviderRegistry*.

3.6 PACKAGE MODEL.SEARCH

3.6.1 *SearchEntityList* | Lista delle entità cercate

3.6.2 *SearchManager* | Gestione della ricerca

3.6.3 *SearchQuery* | Query di ricerca

Tale componente modella la query di ricerca, ossia la stringa inserita dall'utente nella barra di ricerca (*SearchBar*) e contenente una lista di termini o espressioni separati da virgola.

La classe fornisce i metodi per effettuare l'analisi sintattica della stringa al fine di estrapolare i termini (*SearchTerm*) da cercare e stabilire quali di essi siano etichette o frasi.

DIPENDENZE A ciascuna istanza della classe sono associati dei termini di ricerca (*SearchTerm*).

3.6.4 *SearchResult* | Risultati della ricerca

DIPENDENZE Ciascuna istanza della classe gestisce un insieme di contenuti (*Content*) corrispondenti ai criteri di ricerca.

3.6.5 *SearchScope* | Ambito di ricerca

La classe rappresenta il generico ambito di ricerca, che l'utente può selezionare mediante il corrispondente selettore (*SearchScopeSelector*).

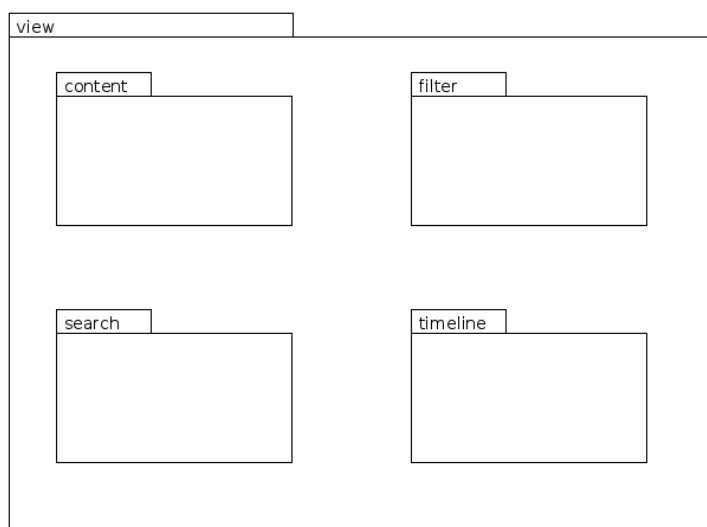
3.6.6 *SearchTerm* | Termine di ricerca

La componente rappresenta un'interfaccia comune a tutti i tipi di termini di ricerca, ossia frasi e etichette.

Ciascun termine di ricerca viene memorizzato all'interno di un'istanza di *SearchQuery* e viene passato al *SearchManager* per consentire l'esecuzione della ricerca.

3.7 PACKAGE MODEL.TIMELINE

3.7.1 *Timeline* | Linea del tempo

Figura 2: Diagramma del package *view*

Questo capitolo illustra i componenti del *View*, per ciascuno dei quali è indicato il nome della classe accompagnata da un'identificazione sintetica, separati dal carattere *'|'* (separatore verticale).

4.1 PACKAGE VIEW

4.1.1 *MainWindow* | Finestra principale

La finestra principale rappresenta il contenitore all'interno del quale vengono inseriti e opportunamente collocati tutti gli elementi grafici dell'interfaccia, tra cui gli strumenti di ricerca, i filtri e i contenuti.

4.2 PACKAGE VIEW.CONTENT

4.2.1 *ContentView* | Contenuto informativo

Si tratta del componente grafico deputato a rappresentare graficamente un contenuto informativo, risultato di una ricerca effettuata dall'utente, e le relative informazioni, in formato grafico o testuale e distinte in essenziali o aggiuntive.

Le informazioni essenziali sono utili per una immediato e preciso inquadramento del contenuto informativo da parte dell'utente al fine di stabilirne la rilevanza soggettiva.

*Informazioni
essenziali*

AUTORE L'autore del contenuto è l'utente (*User*) che lo ha pubblicato all'interno della piattaforma e viene rappresentato testualmente mediante il suo *nome utente* o *proprio*.

ATTINENZA Il grado di attinenza di un contenuto rispetto ai criteri di ricerca corrisponde - in percentuale - al rapporto tra le entità assegnate e quelle cercate. Tale informazione viene rappresentata graficamente variando proporzionalmente la *dimensione* dell'elemento grafico.

DATA DI PUBBLICAZIONE La data di pubblicazione del contenuto viene indicata testualmente.

TIPO Il tipo di contenuto (*Content*) viene rappresentato graficamente mediante la forma¹ specifica dell'elemento grafico e può essere:

TITOLO Il titolo assegnato al contenuto rappresenta un'informazione chiave e viene rappresentata in formato testuale.

Le informazioni aggiuntive forniscono all'utente dettagli utili per approfondire l'esame di un contenuto.

*informazioni
aggiuntive*

ARGOMENTO Ciascun contenuto può riferirsi al più ad un argomento, rappresentabile in maniera grafica (colore o simbolo) o testuale.

EMOZIONI A ciascun contenuto possono essere associate diverse emozioni, che esprimono lo stato d'animo dell'autore al momento della sua redazione o pubblicazione. La lista delle emozioni associate ad un contenuto viene visualizzata in formato testuale.

ENTITÀ A ciascun contenuto possono essere assegnate delle etichette, che riferiscono altrettante entità del dominio da visualizzare in formato testuale (mediante il rispettivo nome).

INTENZIONI A ciascun contenuto possono essere associate delle intenzioni, che possano fornire delle linee guida interpretative ai lettori. La lista delle intenzioni associate ad un contenuto viene visualizzata in formato testuale.

4.3 PACKAGE VIEW.FILTER

4.3.1 *FilterContainer* | Livello filtri

Il *FilterContainer* raccoglie e mantiene separati in un livello distinto gli elementi grafici di tipo *FilterView*, che forniscono all'utente gli strumenti per impostare i filtri di ricerca.

4.3.2 *FilterView* | Filtro di ricerca

Si tratta dell'interfaccia delle componenti grafiche, che rappresentano le tipologie standard dei filtri per il raffinamento dei risultati di una ricerca, ossia (*ListFilterView*, *RangeFilterView*, *SwitchFilterView* e *ValueFilterView*).

¹ Le forme utilizzate sono elementari per garantire l'immediata riconoscibilità da parte di qualsiasi tipo di utente, evitando l'impiego di forme potenzialmente ambigue o ignote a seconda del suo profilo sociale, culturale, geografico, ...

4.3.3 *ListFilterView* | Filtro con lista di valori

Il componente grafico - associato alla classe *ListFilter* - visualizza le liste dei valori ammissibili e bloccati e consente all'utente di:

- spostare un valore da una lista all'altra;
- azzerare il filtro, spostando automaticamente tutti i valori nella lista degli ammissibili.

4.3.4 *RangeFilterView* | Filtro con intervallo di valori

Il componente grafico - associato alla classe *RangeFilter* - permette all'utente di specificare l'estremo inferiore e superiore dell'intervallo dei valori ammissibili secondo le regole definite nella suddetta classe.

4.3.5 *SwitchFilterView* | Filtro a doppio stato

Il componente grafico - associato alla classe *SwitchFilter* - consente di abilitare o disattivare il filtro corrispondente o di visualizzarne lo stato corrente.

4.3.6 *ValueFilterView* | Filtro con soglia di valore

Il componente grafico - associato alla classe *ValueFilter* - consente all'utente di modificare il valore della soglia associata al filtro corrispondente.

4.4 PACKAGE VIEW.SEARCH

4.4.1 *EntityList* | Elenco delle entità cercate

Al termine della ricerca, tale componente visualizza le entità (*EntityView*) riferite dalle etichette e cercate tra i contenuti.

4.4.2 *EntityView* | Entità

Tale componente grafico visualizza le informazioni essenziali associate ad un'entità e consente all'utente di:

- visualizzare la lista delle entità padre;
- visualizzare la lista delle entità figlie;
- sostituire l'entità corrente con un padre o un figlio.
- eliminare l'entità corrente.

4.4.3 *SearchBar* | Barra di ricerca

La barra di ricerca rappresenta il componente grafico (campo di testo) mediante il quale l'utente può inserire i termini di ricerca.

4.4.4 *SearchScopeSelector* | Selettore dell'ambito di ricerca

Il selettore dell'ambito di ricerca è un componente grafico (menu a tendina) che permette all'utente di circoscrivere la ricerca ad informazioni specifiche:

TUTTO

Estende la ricerca a tutti i tipi di informazioni indicizzate o ricercabili all'interno della piattaforma.

ETICHETTE

Limita la ricerca alle sole etichette assegnate ai contenuti.

FRASI

Limita la ricerca alle informazioni presenti in un contenuto (titolo, corpo, ...).

4.4.5 *LabelEntityList* | *Accezioni di un'etichetta*

Ove il sistema individui tra i termini cercati delle etichette aventi accezioni multiple, tale componente grafico provvedere a mostrare - per ciascuna etichetta - la lista delle entità (*EntityView*) riferite e consente all'utente di selezionare quella rispetto cui intenda procedere con la ricerca.

Al termine della ricerca, l'entità selezionata viene mostrata nel componente *EntityList*.

4.5 PACKAGE VIEW.TIMELINE

4.5.1 *TimelineView* | *Cronologia dei contenuti*4.5.2 *TimeSlot* | *Unità temporale*4.5.3 *TimeAxis* | *Asse temporale*

COMPONENTE CONTROLLER

Questo capitolo illustra i componenti del *Controller*, per ciascuno dei quali è indicato il nome della classe accompagnata da un'identificazione sintetica, separati dal carattere 'l' (separatore verticale).

5.1 PACKAGE CONTROLLER

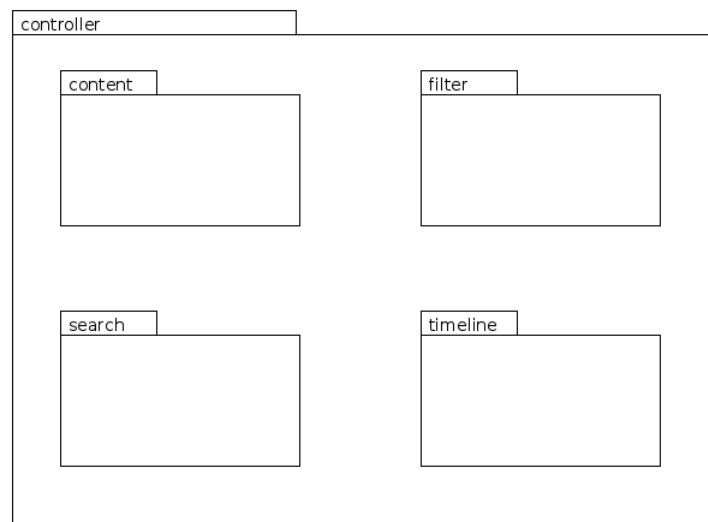


Figura 3: Diagramma del package *controller*

5.2 PACKAGE CONTROLLER.CONTENT

5.3 PACKAGE CONTROLLER.FILTER

5.4 PACKAGE CONTROLLER.SEARCH

5.5 PACKAGE CONTROLLER.TIMELINE