Vantaggi dell'Astrazione attraverso l'Approccio Orientato agli Oggetti per il Digital Scholarly Editing

Angelo M. Del Grosso, CNR-ILC, angelo.delgrosso@ilc.cnr.it
Federico Boschetti, CNR-ILC, federico.boschetti@ilc.cnr.it
Emiliano Giovannetti, CNR-ILC, emiliano.giovannetti@ilc.cnr.it
Simone Marchi, CNR-ILC, simone.marchi@ilc.cnr.it

1 Introduzione

La comunità delle Digital Humanities (DH) sta diventando sempre più inclusiva, non solo nei confronti di gruppi di ricerca in ambito computazionale, ma anche nei confronti delle comunità che praticano le discipline umanistiche con metodi non digitali, con le quali nel passato c'era un difetto di comunicazione. Grazie a questo dialogo ritrovato, è necessario che l'umanista digitale accolga le esigenze di questa allargata multidisciplinarietà. Le DH sono chiamate quindi ad essere sempre meno autoreferenziali e sempre più attente a definire metodi che producano risultati rilevanti per le discipline tradizionali. La collaborazione fra umanisti e informatici, spesso con l'intermediazione degli umanisti digitali, è ormai un fatto assodato e porta a progetti collaborativi che promuovono l'interoperabilità delle risorse ma non garantiscono generalmente la riusabilità delle componenti software e la personalizzazione delle implementazioni.

Infatti, gli strumenti realizzati all'interno di progetti collaborativi difficilmente riescono a recuperare moduli sviluppati in altri progetti complementari, limitando sensibilmente la possibilità di cooperare senza pesanti adattamenti e ristrutturazioni del software. Per affrontare questo stato di cose, è necessario dunque lo sviluppo di modelli condivisi, astratti e formali finalizzati alla costruzione di strumenti flessibili, estendibili e riusabili rivolti allo studio filologico del testo (fra gli altri cfr. Boschetti e Del Grosso 2015). Proprio in questa direzione stanno andando le grandi iniziative infrastrutturali quali DARIAH[1], DiXIT[2], CLARIN[3], TAPOR[4], DiRT[5], impegnate inizialmente nella catalogazione delle risorse, degli strumenti e alla descrizione dei desiderata (requisiti utente). E' quindi opportuno procedere verso la definizione di un processo community-driven volto alla formalizzazione di tipi di dato astratti (Gabbrielli e Martini 2010; in inglese: *Abstract Data Type*, ADT) in grado di standardizzare i requisiti raccolti.

[1] http://dariah.eu
[2] http://dixit.uni-koeln.de
[3] https://www.clarin.eu
[4] http://www.tapor.ca
[5] http://dirtdirectory.org

2 Contesto

La ricerca di modelli generici per lo studio del testo digitale ha prodotto negli ultimi anni molte riflessioni e dibattiti (si vedano per esempio Robinson 2013b; Sahle 2013; Vanhoutte 2010; Shillingsburg 2006b; Thaller 2006). Per esempio, già nel lavoro "Humanities Computing" McCarty 2005, si evidenziava la necessità di definire sistematicamente, attraverso un processo rigoroso, modelli astratti che fossero condivisi dalla comunità.

A distanza di oltre 10 anni, la discussione sulla necessità di realizzare modelli concettuali per la definizione delle entità di dominio resta ancora una questione aperta (cfr. Pierazzo 2015b, in particolare il capitolo 2 - Modelling Digital Texts; cfr. Schreibman, Siemens e Unsworth 2016) Del resto, l'AIUCD ha tra i suoi obiettivi principali la condivisione delle best practices maturate nelle diverse discipline che hanno come centro di interesse o come interesse secondario il trattamento del testo (Agosti e Tomasi 2014; Tomasi 2015).

3 Approccio metodologico

Le esperienze maturate in seno a progetti nei quali siamo stati coinvolti negli ultimi anni hanno messo in evidenza che astrazioni in grado di rappresentare risorse di natura diversa, come ad esempio, nel caso dell'allineamento, singole parole, parole e immagini, unità di testo a diverse granularità, consentono di sviluppare sistemi più flessibili e riusabili. Tra i nostri progetti citiamo: l'ERC "Greek into Arabic[6]" (testi greci e arabi in parallelo); il PRIN "F. de Saussure[7]" (testo e immagini affiancate); il progetto di Traduzione del Talmud Babilonese[8] (segmenti di testo in lingua originale con la relativa traduzione italiana); il progetto "Clavius on the Web[9]" (testo e immagini allineati); il CoPhiProofreader[10] (allineamento dei risultati del processo di OCR); il software Euporia[11] (testi con traduzione a fronte da annotare).

L'approccio qui proposto prevede l'adozione di un processo rigoroso e orientato agli oggetti (in inglese: *Object Oriented Analysis and Design & Object Oriented Programming*, OOAD&OOP) mutuato da pratiche di ingegneria del software ormai consolidate (Dathan e Ramnath 2015). Questo prevede, inoltre, il coinvolgimento della comunità interessata allo studio filologico del testo digitale fin dalle prime fasi della sua attuazione. Ci proponiamo, infatti, di seguire un processo community-driven (Vernon 2013) e user-centered (Gibbs e Owens 2012) che tenga in particolare considerazione le esigenze degli utenti finali.

Obiettivo della metodologia è quello di definire tipi di dato astratti in grado di cogliere le caratteristiche essenziali del dominio di interesse. L'approccio si articola nei seguenti passi:

1. individuazione dei requisiti utente tramite la definizione di *user stories* (Cohn 2004) (ad esempio: "*come* editore critico, *voglio* ricercare tutte le varianti di una lezione di un testo *al fine di* confrontare i vari testimoni");

[6] http://www.greekintoarabic.eu, https://github.com/literarycomputinglab/G2A_Wapp
[7] http://webilc.ilc.cnr.it/viewpage.php/sez=ricerca/id=917/vers=ita
[8] https://www.talmud.it
[9] http://claviusontheweb.it
[10] https://github.com/CoPhi/cophiproofreader
[11] https://github.com/CoPhi/EUporiaJsf

2. identificazione delle entità di dominio (Del Grosso et al. 2016) (ad esempio: l'entità *Source* denota il concetto di fonte primaria, mentre l'entità *Content* denota il contenuto informativo della fonte stessa);

3. definizione degli ADT (Boschetti et al. 2014) attraverso l'identificazione a) dei modelli astratti (ad esempio *Document* per la rappresentazione digitale di un documento) e b) delle relative operazioni (ad esempio aggiunta di una annotazione linguistica ad una porzione di un documento).

Una volta definiti gli ADT sarà possibile realizzare, a partire da essi, dei componenti software, da intendersi come unità uniformi di servizi (per esempio analisi linguistica, annotazione del testo, ecc.) invocabili tramite interfacce di programmazione (API). La realizzazione di interfacce grafiche (GUI) completerà il processo di sviluppo.

Come si è detto all'inizio di questa sezione, ci siamo occupati di vari tipi di allineamento (testo a diversi livelli di granularità, testo e immagine, ecc.) e per questo abbiamo cercato di generalizzare l'approccio proponendo un tipo di dato astratto schematizzato in Tab. 1.

Grazie a questa formalizzazione siamo in grado di standardizzare l'uso del servizio e garantire l'allineamento di liste di oggetti di tipo diverso. La flessibilità di tale soluzione risiede nella personalizzazione della strategia di allineamento. Ad esempio se si allineano stringhe la strategia farà uso di meccanismi basati sull'*edit distance*, mentre se si allineano oggetti più complessi la strategia valuta la distanza nello spazio vettoriale delle sue *features*.

A questo proposito stiamo lavorando anche alla definizione di modelli per la rappresentazione del testo e delle relative risorse contestuali (lessici, terminologie e ontologie) e alla realizzazione di componenti software per il loro trattamento. Nel caso specifico del problema dell'allineamento, potremo, infatti, avvalerci di nuove *features* di natura lessico-semantica. Sarà possibile valutare, ad esempio, l'allineamento tra due parole sulla base di una similarità semantica rappresentata attraverso una relazione di sinonimia specificata tra i rispettivi sensi lessicali (nel nostro caso codificati attraverso il modello *lemon*[12]).

4 Conclusioni

Troppo spesso gli strumenti sviluppati nell'ambito delle DH si limitano a risolvere problemi specifici (tipicamente attraverso interfacce grafiche) senza particolare attenzione alla riusabilità dei modelli e dei moduli software; una possibile spiegazione si ritrova nella effettiva carenza di astrazioni delle entità di dominio e delle librerie software che possano manipolarle.

Come descritto in questo contributo, si intende proporre alla comunità delle DH un approccio ingegneristico, orientato agli oggetti, che possa contribuire a uno sviluppo più sistematico e condiviso di entità software da trattare come vere e proprie risorse (al pari di quelle testuali / documentali) da arricchire, estendere e condividere.

Un lavoro di questo tipo, per definizione, richiede il coinvolgimento dell'intera comunità, alla quale si chiede di specificare, in modo chiaro e puntuale, i requisiti d'uso fondamentali e condivisi dai quali partire per realizzare componenti software che siano utilizzabili (previ adattamenti minimi) da un numero maggiore possibile di utenti.

[12] http://lemon-model.net

public class **AlignmentTable**		
	AlignementTable()	Costruttore
void	**addList**(*List*<S> list)	Aggiunta di una lista di elementi da allineare con gli elementi di altre liste
void	**alignAll**(*Strategy*<S,T> strategy)	Processo di allineamento con strategia personalizzata
Record<T>	**getRecord**(int idx)	Ritorna l'allineamento alla posizione specificata da idx
Element<T>	**getElement**(int recordIdx, int idx)	Ritorna l'elemento della tabella specificato dalla posizione *idx* del Record specificato da *recordIdx*
List<*Element*<T>>	**getColumn**(int idx)	Ritorna la lista allineata alla colonna specificata da idx
boolean	**isAligned**()	Verifica se le liste sono state allineate
...	...	Iteratori...

Tabella 1: AlignmentTable Abstract Data Type. L'ADT specifica le operazioni (il comportamento) che possono essere effettuate sugli oggetti che lo istanziano. La rappresentazione dello stato è nascosto all'utente dell'ADT.

Operativamente, si intende proporre la costituzione di uno Special Interest Group aperto a tutti coloro che siano disposti a confrontarsi nella raccolta dei requisiti degli utenti, delle entità di dominio, degli ADT e delle API.

La metodologia di lavoro prevede l'utilizzo di un processo iterativo (es. Agile use case driven e Domain-driven design) nel quale i requisiti potranno evolvere in accordo ai contributi provenienti dalla comunità fino a stabilizzarsi in una forma che sia la più largamente condivisa: sarà possibile far fronte alla loro natura intrinsecamente instabile proprio attraverso l'adozione del paradigma di analisi, progettazione e sviluppo orientato agli oggetti.

Questo processo si potrà concretizzare, ad esempio, attraverso la costituzione di un portale pubblico comprensivo di forum, wiki e chat, che fungerà da hub per la raccolta dei contributi provenienti dai gruppi italiani impegnati nelle DH interessati a partecipare a questa iniziativa.

Il risultato atteso sarà la pubblicazione di documenti di specifica i quali guideranno lo sviluppo di strumenti software, flessibili ed estendibili, in grado di soddisfare i bisogni propri della nostra comunità.

Il convegno dell'AIUCD, coerentemente alla vocazione fortemente inclusiva e collaborativa esplicitata nella *call for papers*, potrà fornirci l'opportunità di lanciare pubblicamente questa iniziativa.

Bibliografia

Agosti, M., e F. Tomasi, cur. 2014. *Collaborative Research Practices and Shared Infrastructures for Humanities Computing*. Proceedings of revised papers of the Annual Conference of the Associazione per l'Informatica Umanistica e la Cultura Digitale (AIUCD). 11-12 December 2013 Padova: Cooperativa Libraria Editrice Università Di Padova (CLEUP). ISBN: 9788867872602.

Boschetti, F., e A. M. Del Grosso. 2015. «TeiCoPhiLib: A Library of Components for the Domain of Collaborative Philology». A cura di A. Ciula e F. Ciotti. [on line journal], *Journal of the Text Encoding Initiative*, n. 8 (). ISSN: 2162-5603. doi:10.4000/jtei.1285. http://jtei.revues.org/1285.

Boschetti, F., et al. 2014. «A top-down approach to the design of components for the philological domain». In *Book of abstract of Digital Humanities Conference*, a cura di M. Terras, 109–111. Lousanne. http://dharchive.org/paper/DH2014/Paper-673.xml.

Cohn, M. 2004. *User Stories Applied: For Agile Software Development*. Agile Software Development. Crawfordsville, Indiana: Addison-Wesley Professional. ISBN: 9780321205681.

Dathan, B., e S. Ramnath. 2015. *Object-Oriented Analysis, Design and Implementation*. Second. Undergraduate Topics in Computer Science, 1863-7310. 10.1007/978-3-319-24280-4. Springer International Publishing. ISBN: 9783319242804.

Del Grosso, A. M., et al. 2016. «Defining the Core Entities of an Environment for Textual Processing in Literary Computing». In *Digital Humanities 2016: Conference Abstracts*, a cura di M. Eder e J. Rybicki, 771–775. Kraków: Jagiellonian University & Pedagogical University. ISBN: 9788394276034. http://dh2016.adho.org/abstracts/425.

Gabbrielli, M., e S. Martini. 2010. *Programming Languages: Principles and Paradigms*. First. Undergraduate Topics in Computer Science, 1863-7310. DOI: 10.1007/978-1-84882-914-5. Springer London. ISBN: 9781848829138.

Gibbs, F., e T. Owens. 2012. «Building Better Digital Humanities Tools: Toward broader audiences and user-centered designs». A cura di M. Carassai e E. Takehana. *Digital Humanities Quarterly* 6 (2). ISSN: 1938-4122. http://www.digitalhumanities.org/dhq/vol/6/2/000136/000136.html.

McCarty, W. 2005. *Humanities Computing*. London: Palgrave Macmillan. ISBN: 9781403935045.

Pierazzo, E. 2015b. *Digital Scholarly Editing : Theories, Models and Methods*. Digital Research in the Arts and Humanities. Farnham Surrey: Ashgate. ISBN: 9781472412119.

Robinson, P. 2013b. «Towards a Theory of Digital Editions». A cura di W. Van Mierlo e A. Fachard. ISBN13: 9789042036321, *Variants: The Journal of the European Society for Textual Scholarship*, Variants, n. 10:105–131. ISSN: 1573-3084.

Sahle, P. 2013. *Digitale Editionsformen: Textbegriffe und Recodierung (Teil 3). Zum Umgang mit der Überlieferung unter den Bedingungen des Medienwandels*. Vol. 9. Schriften des Instituts für Dokumentologie und Editorik. Books on Demand. ISBN: 9783848239665.

Schreibman, S., R. Siemens e J. Unsworth, cur. 2016. *A new companion to digital humanities*. Vol. 93. Blackwell companions to literature and culture. Chichester, West Sussex: John Wiley & Sons. ISBN: 9781118680599.

Shillingsburg, P. L. 2006b. *From Gutenberg to Google: Electronic Representations of Literary Texts*. New York, USA: Cambridge University Press. ISBN: 9780521683470.

Thaller, M. 2006. «Waiting for the Next Wave: Humanities Computing in 2006». In *Literatures, Languages and Cultural Heritage in a digital world*. King's College London. http://legacy.cch.kcl.ac.uk/clip2006/content/abstracts/paper04.html.

Tomasi, F., cur. 2015. *Humanities and their Methods in the Digital Ecosystem*. Proceedings of revised papers of the Annual Conference of the Associazione per l'Informatica Umanistica e la Cultura Digitale (AIUCD). 18-19 Settembre 2014, Bologna: Association for Computing Machinery (ACM), New York, NY, USA. ISBN: 9781450332958. http://dl.acm.org/citation.cfm?id=2802612.

Vanhoutte, E. 2010. «Defining Electronic Editions: A Historical and Functional Perspective». In *Text and Genre in Reconstruction: Effects of Digitalization on Ideas, Behaviours, Products and Institutions*, a cura di W. McCarty, 119–144. Digital Humanities. Open Book Publishers. ISBN: 9781906924263.

Vernon, V. 2013. *Implementing domain-driven design*. First. Westford, Massachusetts (US): Addison-Wesley Professional. ISBN: 9780321834577.