Towards a Distributed Architecture for Value Added Services to Web Community Digital Libraries

Maurizio Marchese
Aliaksei Yanchuk
Fausto Giunchiglia

University of Trento, Italy

P2PKM 2005 Workshop,
San Diego, USA
The target problem

- Managing services to scientific/scholar web communities
  - examples: Citeseer, Google Scholar, DBLP, Bibster..
  - manage, navigate and exploit the ever increasing quantity of scientific/scholar electronic information resources
  - semantic content enrichment (i.e. metadata extraction and annotation) adds value to services
  - sources distribution evolving over time

- Next generation Citeseer-like systems:
  - scalability, open communities of practice,
  - extension to value added community services
Centralized vs. communities service

- In current Web Repositories (CiteSeer, Google scholar), scalability is solved through centralization:
  - a one-size-fits-all-users service
  - the number of information providers is low compared to the number of information consumers

- Centralization in web communities is undesirable:
  - technically (access bottlenecks)
  - strategically (society at large)
  - the number of information providers will increase as the barriers to build, join and maintain open collaborative environments will be lowered
Presentation Outline

- Current platforms for scientific communities support

- Proposed CiteSeer.EU (CS.EU) Architecture:
  - Concepts
  - Design challenges
  - Proposed solution

- Support for querying functionalities

- Conclusions and future work
Current platforms for scientific communities support

- Current systems are centralized and support a number of services:

  **data/content level**
  - crawl and collect documents from the web,
  - convert them in different electronic formats,

  **knowledge level**
  - extraction: title, authors, citations
  - navigate citations and visualize context
  - ranking of papers and authors (scientometrics)
  - indexes also of commercial digital libraries
Proposed CiteSeer.EU Architecture

- **Concepts:**
  - **Platform:** service-oriented infrastructure
    - Data delivery and security services as baseline
    - All other services as extensions services
  - **Site:** installed Platform instance
  - **Community:** unions of a number of sites with common goals (typically scientific domain, or research objective)
  - **Community Groups**
CS.EU: Enabling Community

- the initial community configures CS.EU software to enable distributed community with initially single community participant.
CS.EU: Second participant Certification and Join
Internal Expansion
Unlimited scalability
And it would feel this:
P2P - Style

- CiteSeer.EU is not 100% P2P
  - the community is very rigid, and serious security mechanisms on the community level ensure that no random site would access it.

- P2P-style comes on the level of individual users:
  - CS.EU offers “firewalled” P2P functionality between the users with incorporated data confidentiality mechanisms.
Main Design Challenges

- Design should be capable of covering full spectrum of core web communities services
- Unconstrained scalability options
- Unconstrained functionality enhancement options
Main Technical Challenges

- Support virtually any communication protocol
- Be flexible enough to provide shrink-to-fit deployment capabilities
- Balance unrestricted access and accountability
- Last, but not least:
  - Balance needs for security with extensible, open environment dynamism
  - Trusted community user authentication and authorization without centralized authority
And this is how we are doing it:

- Use **Extended SOA** as baseline architecture
- Bare architecture provides:
  - Address Space Service
  - Data Delivery Service and Network
  - Security Service
  - Infrastructural Service and Gateway Service
Address Space Service

● **Present State**: Synthetic primary identification key
  - present approach to indexing data sets is to associate a **synthetic** primary identification key with a tuple, and use the secondary (auxiliary) part to determine the value of the primary key.
  - to search, meaningful properties of the tuple have to be processed (indexes) to allow resolution from their values to the primary key.
  - Example: in the Internet the URLs may be considered a primary key, since the URL does not provide details of the content. The search engines employ full-text indexing to index text and allow finding the URLs.
Address Space Service

● CiteSeer.EU proposal:
  compound indexes scheme:
  - Extended notion of the primary key to allow this key to be *computable*, rather than be a *synthetic*;
  - Allow enough leeway in the architecture to provide unlimited options to build further specific search engines.
Data Piece Address

- Address $A(D_i)$ of the $i^{th}$ data piece $D_i$ comprises two parts:

$$A(D_i) = <I_i, \Psi_j>$$

- $I_i$: set of individual atomic indexes $(i_1, i_2, ..., i_n)$
- $\Psi_j$: $j^{th}$ supplier identification, i.e. a pseudo-address that points out to a resolvable service endpoint capable to supply a data stream for the data piece identification.
The Address Space Service’ primary goal is to provide the extended primary key.

- The set I_i is a superposition of atomic indexes, provided by the supplier of the data (manually or automatically).

Difference with classical Web publishing:

- At present, the user places the paper on the Internet, and allows the search engine to perform indexes.
- CiteSeer.EU community orientation: the indexer would know enough about the motivation either of the provider or of the searcher to provide enough additional data, along with the synthetic primary key.
“structuricity” of the primary key

- **CSEU’s PK is a hybrid between synthetic primary key and structured description of the entity.**

- Searching use case:
  - the searcher makes assumption on important properties of the data piece it intends to find – i.e. key words, list of cited papers, frequency of certain words, distance between key words, and so forth.
  - this assumption is captured in a primary key template that is passed to the address space service.
  - the address space service returns the addresses that actually match the template.

- Providing structure to the peer sites allow the sites to use the structured information into their own indexing systems
CSEU Content Identification Concept: PK + metadata

- $I_i$ is superposition of atomic indexes (ids)
- Technical and descriptive roles
- At least one atomic index is used as synthetic PK to facilitate database queries
- Superposition allows **scoping** of the identification for user
Example: a typical I$_{i}$ scoping
Scoping Example: Content and Citations only
Example: citation indexing service

- The citations are introduced as atomic indexes when a paper is introduced into the environment
  - All citations are captured automatically or semi-automatically by the software. The citations database will be propagated to all Sites of the community.
  - If particular Site would wish to introduce additional services to the community, it does so by extending the primary key to contain additional atomic indexes pertaining to the citation.
- A searcher may select the scope, that can apply to
  - generally accepted citations database;
  - refined by particular site database.
Power of “pseudo-address”

- Pseudo-address is *what we have and who can service it*.
- It codifies the sufficient information to perform discovery of service endpoint, establish communication and request data stream.
- Identifies useful *content*.
  - Content may be single document, could be collection of related documents
  - Caller chooses most optimal service that would provide it.
Data Delivery Service and Data Delivery Network

- **Data Delivery Service**: takes the pseudo-address and data piece ID and knows how to reach the data streams

- **Data Delivery Network**: distributed persistence with Address Space and Data Delivery Services:
  - Distributed address space that stores meta-data information about available data sources
  - Pseudo-address resolution and delivery services
  - P2P routing and authenticity mechanisms
Security Service

- Blend Public Key Infrastructure (PKI) with P2P-style inter-site communications
- Establish comprehensive security role delegation model
- Employ data integrity and confidentiality techniques (signing/encryption), maintain security audit trail
P2P Integrity Challenge

- In the P2P environment, it’s hard to control the distribution. This is both bad and good, depending on what the focus is.

- In our approach, when a source gives data to a receiver, it should be capable of including the distribution scope.
... and CS.EU solution:

- sign and encrypt the data with transfer-time certificates.
Difference with traditional P2P

- User never has direct access to other users – no direct technical connection;
- Exchange operations are done via security / infrastructure services, that implement “data firewall” for outgoing data pieces if necessary.
- User cannot get data / services he isn’t certified to.
To search the data in CS.EU, we support three possible scenarios:

- knowledge of $A(D_i)$ is available prior to query time, and feed it directly to the Data Delivery Network services;
- construction of a template of the address is done at query time and used to perform the lookup query
- extension services are called to perform advanced queries
Template querying

- Implements query-by-example semantics.
  - query is actually a template, thus technically query can be made structural
  - Caller is allowed to make assumptions about structure and properties of the data returned, that may not have relation to the search term
  - Possibility to search based on important content of the document, i.e. text relevance, keywords, citations…
  - It is possible to build power target-specific indexing / searching mechanisms
Template querying implementation

- The template $T^* = < I^* ; Y^* >$ is a pair of two sets $I^*$ and $Y^*$, where $I^*$ is the set of example indexes the caller assumes the $D_i$ would have, and $Y^*$ is the set of example pseudo-addresses the caller assumes the data pieces may be served by.

- The matching function $M$ is a function that checks if template $T^*$ is actually reflected on the data piece address $A(D_i)$
  - for instance ALL example indexes AND pseudo addresses MUST be reflected on the indexes and pseudo-addresses of a particular data piece address.
  - Less restricting criteria can also be applied.
Conclusions and Future Work

● Conclusions
  – Proposal for a distributed architecture for the management and use of digital content in web communities
  – Architectural framework based on
    ● P2P-style architecture to bring the dynamism and decentralize control from P2P system.
    ● xSOA for extensibility
    ● Core concepts: computable compounded indexes and pseudo-addresses

● Future Work
  – implementing the platform
  – exploring properties of compounded indexes
  – exploring appropriate matching function implementation