Semantic Overlay Networks

Karl Aberer and Philippe Cudré-Mauroux
School of Computer and Communication Sciences
EPFL -- Switzerland
Overview of the Tutorial

• I. P2P Systems Overview
• II. Query Evaluation in SONs
  – RDFPeers
  – PIER
  – Edutella
• III. Semantic Mediation in SONs (PDMSs)
  – PeerDB
  – Hyperion
  – Piazza
  – GridVine
• IV. Current Research Directions
What this tutorial is about

- Describing a (pertinent) selection of systems managing data in large scale, decentralized overlays networks
  - Focus on architectures and approaches to evaluate / reformulate queries

- It is *not* about
  - A comprehensive list of research projects in the area
    - But we’ll give pointers for that
  - Complete description of each project
    - We focus on a few aspects
  - Performance evaluation of each approach
    - No meaningful comparison metrics at this stage
I. Peer-to-Peer Systems Overview

- Application Perspective: Resource Sharing (e.g. images)
  - no centralized infrastructure
  - global scale information systems
Resource Sharing

• What is shared?

- content
- bandwidth
- storage
- knowledge
- processing
Enabling Resource Sharing

- Searching for Resources
  - Overlay Networks, Routing, Mapping
- Resource Storage
  - Archival storage, replication and coding
- Access to Resources
  - Streaming, Dissemination
- Publishing of Resources
  - Notification, Subscription
- Load Balancing
  - Bandwidth, Storage, Computation
- Trusting into Resources
  - Security and Reputation
- etc.
P2P Systems

- System Perspective: Self-Organized Systems
  - no centralized control
  - dynamic behavior
What is Self-Organization?

• Informal characterization (physics, biology,... and CS)
  – distribution of control (= decentralization)
  – local interactions, information and decisions
    (= autonomy)
  – emergence of global structures
  – failure resilience and scalability

• Formal characterization
  – system evolution $f_T: S \rightarrow S$, state space $S$
  – stochastic process (lack of knowledge, randomization)
    \[ P(s_j, t+1) = \sum_i M_{ij} P(s_i, t), P(s_i| s_j) = M_{ij} \in [0,1] \]
  – emergent structures correspond to equilibrium states
  – no entity knows all of $S$
Examples of Self-Organizing Processes

- **Evolution of Network Structure**
  - Powerlaw graphs: Preferential attachment + growing network [Barabasi, 1999]
  - Small-World Graphs: FreeNet Evolution

- **Stability of Network**
  - Analysis of maintenance strategies
  - Markovian Models, Master Equations

- **Resource Allocation**
  - game-theoretic and economic modelling

- **Probabilistic Reasoning**
  - Belief propagation for semantic integration (see later)
Efficiently Searching Resources (Data)

• Find images taken last week in Trondheim!
Overlay Networks

• Form a logical network in top of the physical network (e.g. TCP/IP)
  – originally designed for resource location (search)
  – today other applications (e.g. dissemination)
• Each peer connects to a few other peers
  – locality, scalability
• Different organizational principles and routing strategies
  – unstructured overlay networks
  – structured overlay networks
  – hierarchical overlay networks
Unstructured Overlay Networks

- Popular example: Gnutella
- Peers connect to few random neighbors
- Searches are flooded in the network

Example: $C=3, \text{TTL}=2$
Structured Overlay Networks

- Popular examples: Chord, Pastry, P-Grid, ...
- Based on embedding a graph into an identifier space (nodes = peers)
- Peers connect to few neighbors carefully selected according to their distance
- Searches are performed by greedy routing
- Variations of Kleinberg's small world graphs:
  \[ P[u \rightarrow v] \sim d(u, v)^{-r} \]
Conceptual Model for Structured Overlay Networks

- **Six key design aspects**
  - Choice of an identifier space \((l,d)\)
  - Mapping of peers \((F_P)\) and resources \((F_R)\) to the identifier space
  - Management of the identifier space by the peers \((M)\)
  - Graph embedding (structure of the logical network) \(G=(P,E)\) (N - Neighborhood relationship)
  - Routing strategy \((R)\)
  - Maintenance strategy

Set of resources \(R\)

Group of peers \(P\)
Hierarchical Overlay Networks

- Popular Example: Napster, Kaaza
- Superpeers form a structured or unstructured overlay network
- Normal peers attach as clients to superpeers
Beyond Keyword Search

⇒ searching semantically richer objects in overlay networks

<es:cDate> 05/08/2004 </es:cDate>

<xap:CreateDate>2001-12-19T18:49:03Z</xap:CreateDate>
<xap:ModifyDate>2001-12-19T20:09:28Z</xap:ModifyDate>

<myRDF:Date>Jan 1, 2005</myRDF:Date>
Managing Heterogeneous Data

- Support of structured data at peers: schemas
- Structured querying in peer-to-peer system
- Relate different schemas representing semantically similar information

Date?

- `<es:cDate>05/08/2004</es:cDate>`
- `<myRDF:Date>Jan 1, 2005</myRDF:Date>`

©2005, Karl Aberer and Philippe Cudré-Mauroux
II. Query Evaluation in SONs

Beyond keyword search

→ searching complex structured data in overlay networks
Standard RDMS over overlay networks

- Strictly speaking impossible

- **CAP theorem**: pick at most two of the following:
  1. Consistency
  2. Availability
  3. Tolerance to network Partitions

- **Practical compromises**:
  - Relaxing ACID properties
    - **Soft-states**: states that expire if not refreshed within a predetermined, but configurable amount of time

Distributed Hash Table Lookup

- DHT lookups designed for binary relations (key, content)
- Structured data (e.g., RDF statements) can sometimes be encoded in simple, rigid models

Index attributes to resolve queries as distributed table lookups

\[ t = (\text{<info: rdfpeers}> \quad \text{<dc: creator}> \quad \text{<info: MinCai>}) \]

Key 1  
Key 2  
Key 3
**RDFPeers**: A distributed RDF repository

**Who?**
- U.S.C. *(Information Sciences Institute)*

**Overlay structure**
- DHT *(MAAN [Chord]*)

**Data model**
- RDF

**Queries**
- RDQL

**Query evaluation**
- Distributed *(iterative lookup)*
RDFPeers Architecture
Index Creation (1)

Triple $t = \langle \text{info:.rdfpeers} \rangle \langle \text{dc:creator} \rangle \langle \text{info:mincai} \rangle$

- **Soft-states**
  - Each triple has an expiration time
- **Locality-preserving hash-function**
  - Range searches
Index Creation (2)
Query Evaluation

• Iterative, distributed table lookup

1) Get(foaf:Person)

2) Results = $\Pi_{\text{subject}} \sigma_{\text{object}=\text{foaf:pPerson}} (R)$

3) Get(“John”)

4) Results = Results $\cap \Pi_{\text{subject}} \sigma_{\text{object}=\text{“John”}} (R)$
Want more? Distributed RDF Notifications

- **Pub/Sub** system on top of RDFPeers
- **Subscription** = triple pattern with at least one constant term
  - Routed to the peer $P$ responsible of the term
  - $P$ keeps a local list of subscriptions
  - Fires notifications as soon as a triple matching the pattern gets indexed
- **Extensions** for disjunctive and range subscriptions
References


DHT-Based RDMS

• Traditional DHTs only support keyword lookups
• Traditional RDMS do no scale gracefully with the number of nodes

Scaling-up RDMS over a DHT
- Distributing storage load
- Distributing query load
⇒ Relaxing ACID properties
The PIER Project

Who?
- U.C. Berkeley

Overlay structure
- DHT (currently Bamboo and Chord)

Data model
- Relational

Queries
- Relational, with joins and aggregation

Query evaluation
- Distributed (based on query plans)
PIER Architecture

• Peer-to-peer Information Exchange and Retrieval
• Relational query processing system built on top of a DHT
• Query processing and storage are decoupled

→Sacrificing strong consistency semantics
  • Best-Effort
Main Index Creation: DHT Index

- Indexing tuples in the DHT (equality-predicate index)
  - Relation R1: \{35, abc.mp3, classical, 1837,...\}
  - Index on 3rd/4th attributes:
    - \textbf{hash key}=\{R1.classical.1837,35\}

- Soft-state storage model
  - Publishers periodically extend the lifetime of published objects

- No system metadata
  - All tuples are self-describing
Two Other Indexes

• Multicast index
  – Multicast tree created over the DHT

• Range index
  – Prefix hash tree created over the DHT
Query Evaluation

- Queries are expressed in an algebraic dataflow language
  - A query plan has to be provided

- PIER processes queries using three indexes
  - DHT index for equality predicates
  - Multicast index for query dissemination
  - Range index for predicates with ranges
Symmetric hash join

- Equi-join on two tables \( R(A, B) \) and \( S(C, B) \)

1. Disseminate query to all nodes (multicast tree)
   - Find peers storing tuples from \( R \) or \( S \)
2. Peers storing tuples from \( R \) and \( S \) hash and insert the tuples based on the join attribute
   - Tuples inserted into the DHT with a temporary namespace
3. Nodes receiving tuples from \( R \) and \( S \) can create the join tuples
4. Output tuples are sent back to the originator of the query

\[
1) \ R(A, B) \bowtie S(C, B) \\
2) \ R(ai, bj) \implies \text{put(hash(TempSpace bj), (ai, bj))} \\
3) \ S(ck, bj) \implies \text{put(hash(TempSpace bj), (ck, bj))} \\
4) \ R(ai, bj) \bowtie S(ck, bj)
\]
Want more? Join variants in PIER

• **Skip rehashing**
  – When one of the tables is already hashed on the join attribute in the equality-predicate index

• **Symmetric semi-join rewrite**
  – Tuples are projected on the join attribute before being rehashed

• **Bloom filter rewrite**
  – Each node creates a local Bloom filter and sends it to a temporary namespace
  – Local Bloom filters are OR-ed and multicast to nodes storing the other relations
  – Followed by a symmetric hash join, but only the tuples matching the filter are rehashed
References

- J. M. Hellerstein: Toward network data independence. SIGMOD Record 32(3), 2003


Routing Indices

- Flooding an overlay network with a query can be inefficient
- Disseminating a query often boils down to computing a multicast tree for a portion of the peers

Storing semantic routing information at various granularities directly at the peers

- Schema level
- Attribute level
- Value level
The Edutella Project

Who?
- *U. of Hannover (mainly)*

Overlay structure
- *Super-Peer (HyperCup)*

Data model
- *RDF/S*

Queries
- *Triple patterns (or TRIPLE)*

Query evaluation
- *Distributed (based on routing indices)*
Edutella Architecture

- An RDF-based infrastructure for P2P applications
- End-peers store resources annotated with RDF/S
- Super-peer architecture
  - HyperCup super-peer topology
  - Routing based on indices
  - Two-phase routing
    - Super-peer to super-peer
    - Super-peer to peer
Index construction: SP/P routing indices

- Registration: end-peers send a summary of local resources to their super-peer
  - Schema names used in annotations
  - Property names used in annotations
  - Types of properties (ranges) used in annotations
  - Values of properties used in annotations
- Not all levels have to be used
- Super-peers aggregate information received from their peers and create a local index
- Registration is periodic
  - Soft-states
Index Construction: SP/SP routing indices

- Super-peers propagate the SP/S indices to other super-peers with spanning trees

- Super-peers aggregate the information in SP/SP indices
  - Use of semantic hierarchies

<table>
<thead>
<tr>
<th>Granularity</th>
<th>Index of A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schema</td>
<td>dc</td>
</tr>
<tr>
<td></td>
<td>lom</td>
</tr>
<tr>
<td>Property</td>
<td>dc:subject</td>
</tr>
<tr>
<td></td>
<td>dc:language</td>
</tr>
<tr>
<td></td>
<td>lom:context</td>
</tr>
<tr>
<td>Property</td>
<td>ccs:nets</td>
</tr>
<tr>
<td>Value Range</td>
<td>ccs:soft</td>
</tr>
<tr>
<td></td>
<td>eng</td>
</tr>
<tr>
<td>Property</td>
<td>lom:context</td>
</tr>
<tr>
<td>Value</td>
<td>“undergrad”</td>
</tr>
<tr>
<td></td>
<td>“de”</td>
</tr>
</tbody>
</table>
Query Evaluation

Q: (? , dc:language, “de”)
   (? , lom:context, “undergrad”)
   (? , dc:subject, ccs:softwareengineering)
Want More? Decentralized Ranking

- Number of results returned grow with the size of the network
- Decentralized top-k ranking
  - New weight operator to specify which predicate is important
  - Aggregation of top-k in three stages
    - End-peer
    - Super-peer
    - Query originator
References


III. Semantic Mediation in SONs

- What if (some) peers use different schemas to store semantically related data?
  - Need ways to relate schemas in decentralized settings

⇒ unstructured overlay network at the semantic layer
⇒ Peer Data Management Systems (PDMS)
Semantic Mediation Layer

Overlay Layer

“Physical” layer

Correlated / Uncorrelated
Source Descriptions

• Heterogeneous schemas can share semantically equivalent attributes
• On the web, users are willing to annotate resources or filter results manually

Let users annotate their schemas
  – Search & Match similar annotations
  – Use IR methods to rank matches
  – Let users filter out results
PeerDB

Who?
  - National U. of Singapore

Overlay structure
  - Unstructured (BestPeer)

Data model
  - Relational

Mappings
  - Keywords

Query reformulation
  - Distributed

Query evaluation
  - Distributed
PeerDB architecture
Index Construction

- Peers store **keywords** related to local relations / attributes

<table>
<thead>
<tr>
<th>Names</th>
<th>Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kinases</td>
<td>protein, human</td>
</tr>
<tr>
<td>SeqID</td>
<td>key, identifier, ID</td>
</tr>
<tr>
<td>length</td>
<td>length</td>
</tr>
<tr>
<td>proteinSeq</td>
<td>sequence, protein sequence</td>
</tr>
</tbody>
</table>

Attribute names

Provided by experts
Query Reformulation (1)

- Local query $Q(R,A)$
  - $R$: set of local relations
  - $A$: set of local attributes

- Agents carrying the query are sent to neighbors

- Relations $D$ from neighboring peers are ranked w.r.t. a matching function $Match(Q,D)$
  - Higher matching values if $R$’s keywords can be matched to relation names / keywords of the neighbor
  - Higher matching values if $A$’s keywords can be matched to attributes names / keywords of the neighbor
Query Reformulation (2)

- Promising relations with $Match(Q,D) > threshold$ are returned to the user
  - User filters out false positives manually at the relation level

- At the neighbor, the agent reformulates the query with local synonyms for R, A
  - Attributes might be dropped if no synonym is found
  - Results are returned to the query originator

- Query is forwarded iteratively in this manner with a certain TTL
Want More? Network Reconfiguration

- Result performance depends on the semantic clustering of the network
- PeerDB network is reconfigurable according to three strategies:
  - MaxCount
    - Choose as direct neighbors the peers which have returned the most answers (tuples / bytes)
  - MinHops
    - Choose as direct neighbors those peers which returned answers from the furthest locations
  - TempLoc
    - Choose as direct neighbors those peers that have recently provided answers.
References


Mapping Tables

- Semantically equivalent data values can often be mapped easily one onto the other

Specification of P2P mappings at the data value level
- Reformulate queries based on these mapping tables

<table>
<thead>
<tr>
<th>GDB_id</th>
<th>SwissProt_id</th>
</tr>
</thead>
<tbody>
<tr>
<td>GDB:120231</td>
<td>Q14930</td>
</tr>
<tr>
<td>GDB:120231</td>
<td>Q9UMK3</td>
</tr>
</tbody>
</table>

Ids from the GDB relation at Peer P1

Semantically equivalent Ids from SwissProt relation at peer P2
The Hyperion Project

Who?
- U. of Toronto
- U. of Ottawa
- U. of Edinburgh
- U. of Trento

Overlay structure
- Unstructured

Data model
- Relational

Queries
- S+J algebra with projection

Query reformulation
- Distributed

Query evaluation
- Distributed
Hyperion: Architecture

P2P Layer

P2P API

RM

QM

AM

Peer DB

Local DB

Mapping Tables

Mapping Expressions

query

answer

local

global

acquaintances
Creating Mapping Tables

- Initially created by domain experts
- Mapping tables semantics:
  - **Closed-open-world semantics**
    - Partial knowledge
  - **Closed-closed-world semantics**
    - Complete information
- Common associations, e.g., identity mappings, can be expressed with unbound variables
- Efficient algorithm to infer new mappings or check consistency of a set of mappings along a path
Query Reformulation

- Query posed over local relations only
  - $S+J$ algebra with projection

- Iterative distributed reformulations
  - Network flooding (on acquaintance links)

- Local algorithm ensures **sound and complete** reformulation of query $q_1$ at $P_1$ to query $q_2$ at $P_2$
  - Soundness: only values that can be related to those retrieved at $P_1$ are retrieved at $P_2$
  - Completeness: retrieving all possible sound values
Query Reformulation with multiple tables

- Transform the query in its equivalent disjunctive normal form and pick the relevant tables only

<table>
<thead>
<tr>
<th>keyword</th>
<th>kw</th>
<th>article_id</th>
<th>paper_id</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPH</td>
<td>APH</td>
<td>20185348</td>
<td>10719179</td>
</tr>
<tr>
<td>OPH</td>
<td>AARE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NGF receptor</td>
<td>p75 ICD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G9 sialidase</td>
<td>Sialidase 1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Q$_9$: \textit{select} * \\
\textit{from} MedLine \\
\textit{where} keyword = “OPH” AND year = “1999”

Q$_{10}$: \textit{select} * \\
\textit{from} PubMed \\
\textit{where} (kw = “APH” OR kw = “AARE”) AND py = “1999”
Want More? Distributed E.C.A. Rules

- When views between schemas are defined, Consistency can also be ensured via a distributed rule system
  - Event-Condition-Action rule language and execution engine
  - Events, conditions and actions refer to multiple peers

\[ AA_{-}Passenger(p, n) \supseteq BA_{-}Passenger(p, n) \]

```
create trigger passengerInsertion
after insert on BA_Passenger
    referencing new as NewPass
for each row
begin
    insert into AA_Passenger values NewPass
    in Alpha-Air DB;
end
```
References


Extending Data Integration Techniques

- Centralized data integration techniques take advantage of views to reformulate queries in efficient ways.
The Piazza Project

Who?
  - *U. of Washington*

Overlay structure
  - *Unstructured*

Data model
  - *Relational (+XML)*

Queries
  - *Relational*

Query reformulation
  - *Centralized*

Query evaluation
  - *Distributed*
An example of semantic topology

Peer to local DB mapping (Storage Description)

P2P schema mapping (Peer Description)
Creating Mappings in Piazza

- Mappings = views over the relations
  - Cf. classical data integration

- Supported mappings:
  - Definitions (GAV-like)
    
    \[
    9DC : SkilledPerson(PID, "EMT") :-
    \]
    
    \[
    FS : Schedule(PID, vid),
    FS : 1stResponse(vid, s, l, d),
    FS : Skills(PID, "medical")
    \]

  - Inclusions (LAV-like)
    
    \[
    LH : CritBed(bed, hosp, room, PID, status) \subseteq
    \]
    
    \[
    H : CritBed(bed, hosp, room),
    H : Patient(PID, bed, status)
    \]
Posing queries in Piazza

• Local query iteratively reformulated using the mappings

• Reformulation algorithm
  – Input: a set of mappings and a conjunctive query expression \( Q \) (evt. with comparison predicates)
  – Output: a query expression \( Q' \) that only refers to stored relations at the peers

• Reformulation is centralized
Query reformulation in Piazza

- Constructing a rule-goal tree:

Reformulated query:

\[ Q'(r1,r2) : \text{ProjMember}(r1,p), \text{ProjMember}(r2,p), \text{CoAuthor}(r1,r2) \cup \text{ProjMember}(r1,p), \text{ProjMember}(r2,p), \text{CoAuthor}(r2,r1) \]
More? Piazza & XML

- Piazza also considers query reformulation for semi-structured XML documents
- Mappings expressed with a subset of XQuery
  - Composition of XML mappings
- Containment of XML queries
References


Semantic Gossiping (Chatty Web)

- Schemas might only partially overlap
- Mappings can be faulty
  - Heterogeneity of conceptualizations
  - Inexpressive mapping language
  - (Semi-) automatic mapping creation

Self-organization principles at the semantic mediation layer
  - Detect inconsistent mappings
  - Per-hop semantic forwarding
    - Syntactic criteria
    - Semantic criteria
GridVine

Who?
  - EPFL

Overlay structure
  - DHT (P-Grid)

Data model
  - RDF (annotations) RDFS (schemas) OWL (mappings)

Queries
  - RDQL

Query reformulation
  - Distributed

Query evaluation
  - Distributed
GridVine Architecture

- Data / Schemas / Mappings are all indexed

⇒ Decoupling
Deriving Routing Indices (semantic layer)

- Automatically deriving quality measures from the mapping network to direct reformulation
  - Cycle / parallel paths / results analysis
Example: Cycle Analysis

- What happened to an attribute $A_i$ present in the original query?
  - $(T_{1\rightarrow\ldots\rightarrow n\rightarrow 1})$ (Creator) = (Creator) ✓
  - $(T_{1\rightarrow\ldots\rightarrow n\rightarrow 1})$ (Creator) = (Subject) ✗
  - $(T_{1\rightarrow\ldots\rightarrow n\rightarrow 1})$ ($A_i$) = ∅
Example: Cycle Analysis

- In absence of additional knowledge:
  - “Foreign” links have probability of being wrong $\varepsilon_{cyc}$
  - Errors could be “accidentally” corrected with prob $\delta_{cyc}$
- Probability of receiving positive feedback (assuming $A \rightarrow B$ is correct) is $(1-\varepsilon_{cyc})^5 + (1-(1-\varepsilon_{cyc})^5)$ $\delta_{cyc} = \text{pro}^+(5, \varepsilon_{cyc}, \delta_{cyc})$
Example: Cycle Analysis

- Likelihood of receiving series positive and negative cycle feedback $c_1, \ldots, c_k$:

$$l(c_1, \ldots, c_k) =$$

$$(1 - \varepsilon_s) \prod_{c_i \in C^+} \text{pro}^+(|c_i|, \varepsilon_{\text{cyc}}, \delta_{\text{cyc}}) \prod_{c_i \in C^-} (1 - \text{pro}^-(|c_i|, \varepsilon_{\text{cyc}}, \delta_{\text{cyc}})) + \varepsilon_s \prod_{c_i \in C^+} \text{pro}^-(|c_i|, \varepsilon_{\text{cyc}}, \delta_{\text{cyc}}) \prod_{c_i \in C^-} (1 - \text{pro}^+(|c_i|, \varepsilon_{\text{cyc}}, \delta_{\text{cyc}}))$$
Which Link to Trust?

- Without other information on $\varepsilon_{\text{cyc}}$ and $\delta_{\text{cyc}}$, likelihood of our link being correct or not:
  $$p^+ = \lim_{\varepsilon \to 0} \int_{\delta_{\text{cyc}}} \int_{\varepsilon_{\text{cyc}}} \frac{1}{(c_1, \ldots, c_k)} \, d\varepsilon_{\text{cyc}} \, d\delta_{\text{cyc}}$$
  $$p^- = \lim_{\varepsilon \to 1} \int_{\delta_{\text{cyc}}} \int_{\varepsilon_{\text{cyc}}} \frac{1}{(c_1, \ldots, c_k)} \, d\varepsilon_{\text{cyc}} \, d\delta_{\text{cyc}}$$

$$\Rightarrow \gamma = \frac{p^+}{(p^++p^-)}$$
Reformulating query: Semantic Gossiping

- Selectively forward queries at the semantic mediation layer
  - Syntactic thresholds
    - Lost predicates
  - Semantic thresholds
    - Results analysis
    - Cycles analysis
- Drop/Repair faulty mappings
  - Self-organized semantic layer
Decentralized Query Resolution: Overview

Query

SELECT ?picture
WHERE {?picture <rdf:type> <NewYearPic:NewYearPicClass>}
  {?picture <NewYearPic:Location> ?loc}
AND ?loc =~ /Lausanne/

NewYearPic schema

Translation Link

1

2

3

P-Grid

DSC000045.JPG Annotations

4

©2005, Karl Aberer and Philippe Cudré-Mauroux
Want more? Belief Propagation in SONs

- Inferring **global** semantic quality values from a decentralized message-passing process
References


IV. Current Research Directions
Emergent Semantics

- Semantic Overlay Networks can be viewed as highly dynamic systems (churn, autonomy)
- Semantic agreements can be understood as emergent phenomena in complex systems

⇒ Principles
- mutual agreements for meaningful exchanges
- agreements are dynamic, approximate and self-referential
- global interoperability results from the aggregation of local agreements by self-organization

SON Graph Analysis

• Networks resulting from self-organization processes
  – powerlaw graphs, small world graphs
• Structure important for algorithm design
  – distribution, connectivity, redundancy

⇒ Analysis and Modeling of SON from a graph-theoretic perspective

Information Retrieval and SONs

- Combination of structural, link-based and content-based search
- Precision of query answers drops with semantic mediation

⇒ IR techniques to optimize precision/recall in SONs
  - Distributed ranking algorithms
  - Content-based search with DHTs
  - Peer selection using content synopsis


Corpus-Based Information Management

- Very large scale, dynamic environments require on-the-fly data integration
- Automated schema alignment techniques may perform poorly
  - Lack of evidence

⇒ Using a preexisting corpus of schema and mapping to guide the process
  - Mapping reuse
  - Statistics offer clues about semantics of structures

J. Madhavan, P. A. Bernstein, A.i Doan and A. Y. Halevy: Corpus-based Schema Matching. ICDE 2005
Declarative Overlay Networks

- Overlay networks are very hard to design, build, deploy and update

→ Using declarative language not only to query, but also to express overlays
  - Logical description of overlay networks
  - Executed on a dataflow architecture to construct routing data structures and perform resource discovery

Internet-Scale Services

- Many infrastructures tackle today data management at Internet scale
  - Semantic Web
  - Web Services
  - Grid Computing
  - Dissemination Services

$\Rightarrow$ SONs as a generic infrastructure for very large-scale data processing
Further References

- Length limits constrained the number of approaches we could discuss...

⇒ http://lsirwww.epfl.ch/SON

For a more complete list of research projects in the area of Semantic Overlay Networks