

The image features a dark blue background with a complex, abstract graphic of white lines. These lines form a grid that curves and warps, creating a sense of depth and movement. The lines are most dense in the lower right quadrant and become sparser towards the top left. The overall effect is reminiscent of a digital or network structure.

metatomix[®]

Metatomix Semantic Platform

About Metatomix

- Founded in 2000
- Privately held
- Headquarters - Dedham, MA
- Offices in Atlanta, Memphis, San Francisco, and London
- Semantic Technology Leadership
 - Numerous patents, both existing and pending
- Go-to-Market Through Partners
 - EMC Velocity Partner
 - Oracle Developer Partner
 - IBM AIO Partner
- National Labs Technology
 - Acquired technology for text and visual analytics

PARTNERS

EMC²
where information lives[®]

ORACLE[®]

IBM

SYBASE[®]



COGNOS[®]

HCL



UNISYS

agile
TECHNOLOGIES

ciber[®]

INVESTORS

BATTELLE | VENTURES



Dunrath Capital

Fortified Investing

NORTH HILL
VENTURES

Financial Services

Public Sector

SSC – Private Edge
Statement Processor

SSC - NAV Dissemination

SSC – Counterparty
Exposure

ACM for ACI

AirBus - Design to Cost

JIS

JFA / JAC

CI

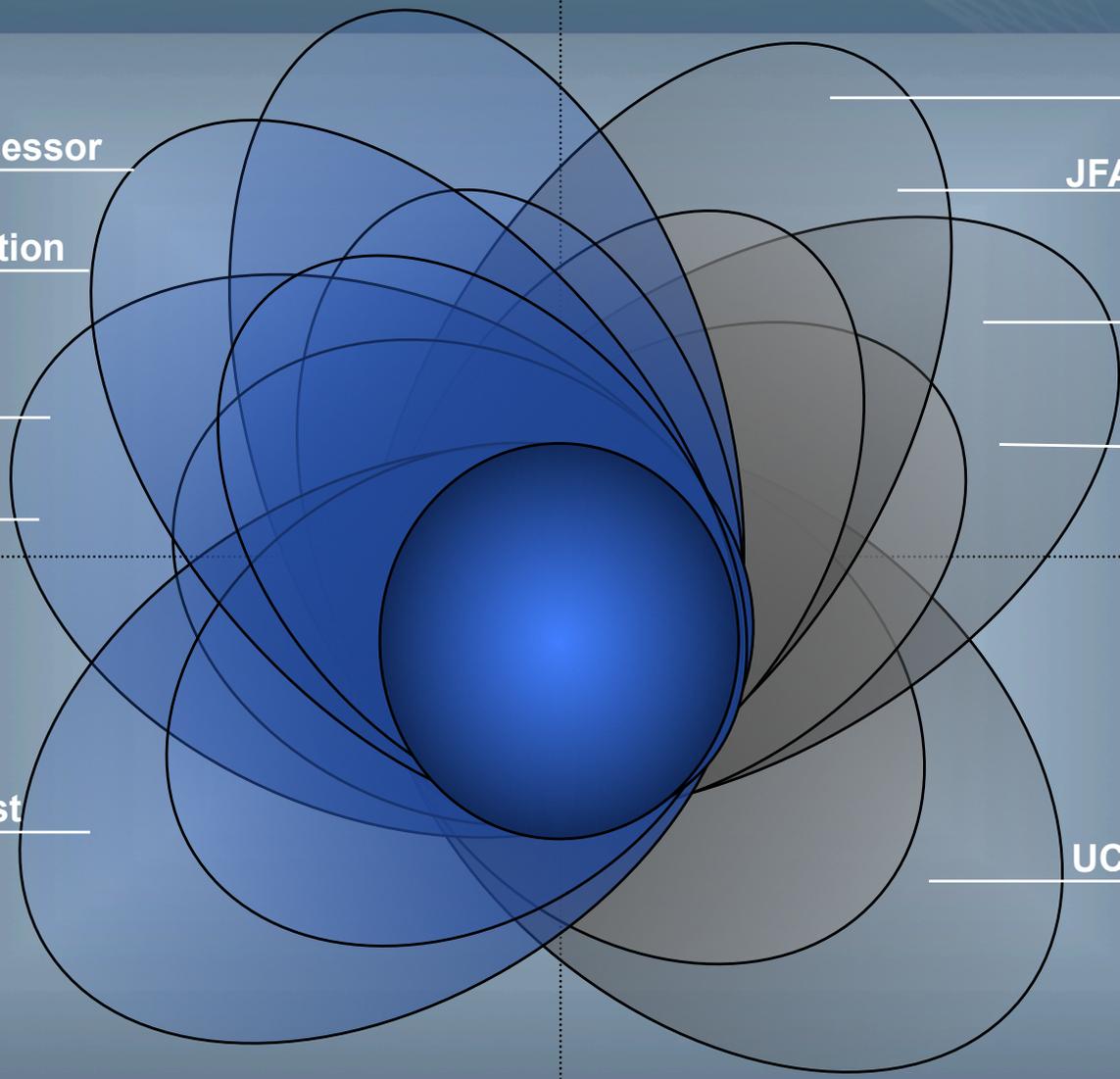
JDX

UCB - Bio Registration

Manufacturing

Life Sciences

metatomix



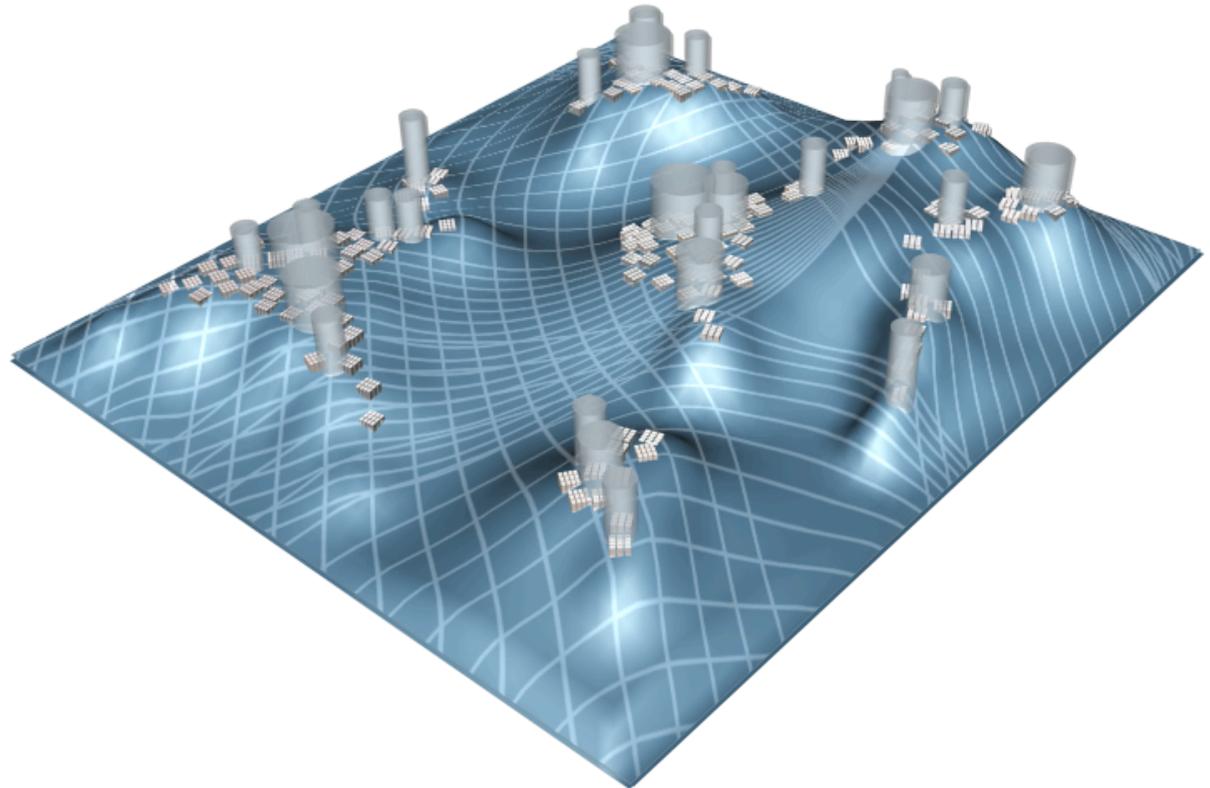
The Metatomix Approach

Metatomix simplifies the way in which an application uses data from ANY data source

Embrace the Distributed Enterprise Landscape

Integrate your Application Specific Data Silos

Leverage Semantic Technology and Rules within our High Performance Runtime Platform to apply a common understanding across applications

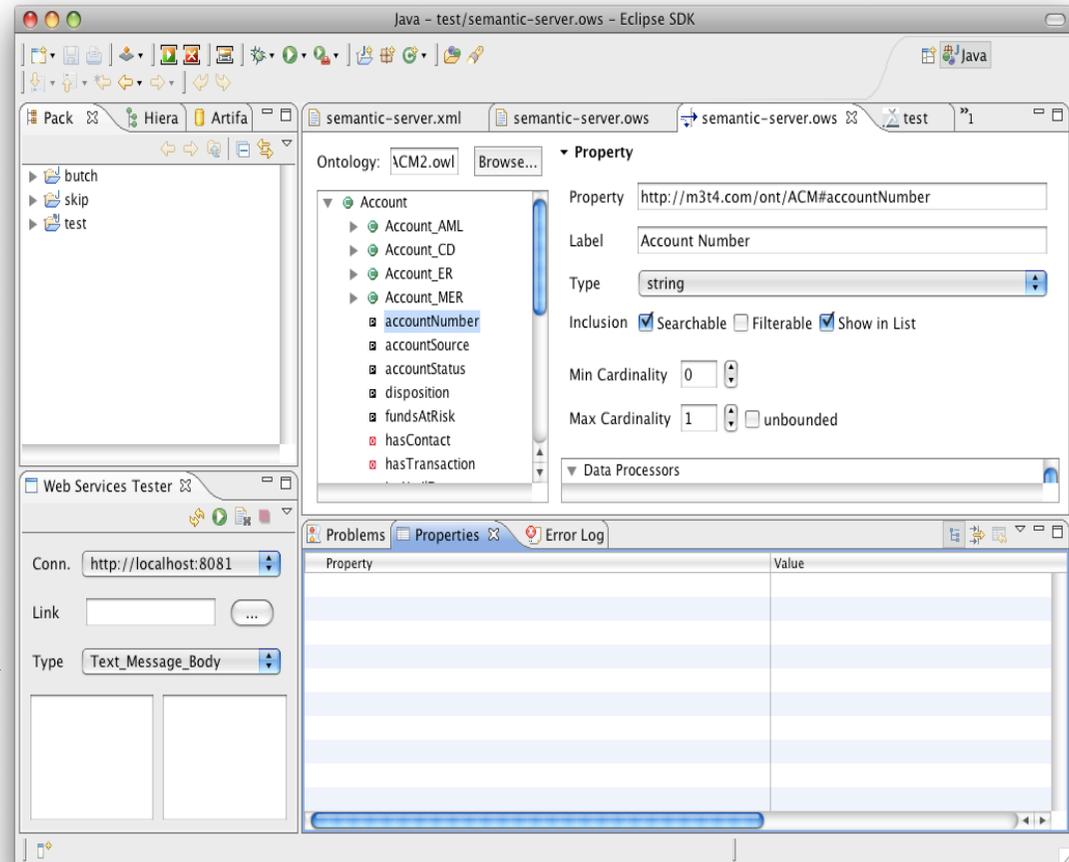
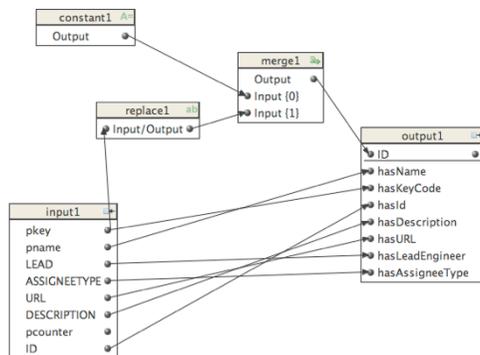


Result: A semantic real-time virtual integration layer that unifies the applications across your enterprise landscape and allows you to see relationships amid 1000's of seemingly unrelated events

m3t4.studio Tools

Eclipse Based Plugins

- Ontology Editor
- RDF Editor
- Rules / Policy Editors
- Mapping Tool
 - Data Sources <-> RDF
- Semantic Editing Framework



RDB2RDF Specific Tools

→ Ontology Editor Import Tool

- Import objects from database
- Add database objects to an owl ontology

→ Eclipse Mapping Tool

- Visual editor in eclipse
- Creates a mapping between a database and RDF

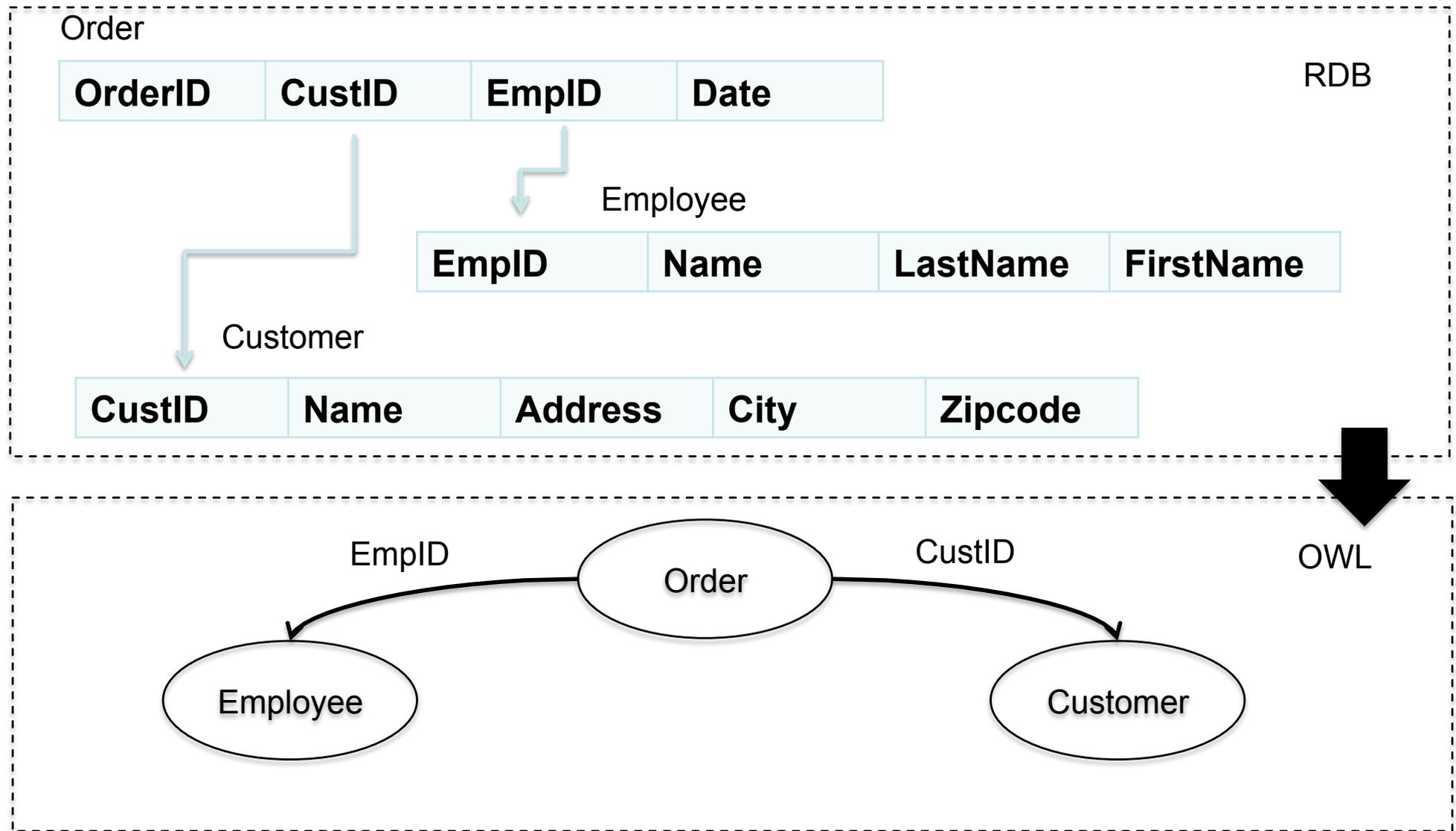
→ Mapping in Mtx Discovery

- Visual editor
- CSV -> RDF

Ontology Editor Import Tool

- Allows for seeding an ontology with a representative relational schema
- Currently reads XML, XSD and Database schemas
- Recognizes relationships among tables and creates the same relationship in the ontology
- Tables translate to classes
- Columns translate to properties
- FK relationships translate into ObjectType Properties

Seeding the Ontology



Connect to the Source Schema

The screenshot shows the Eclipse IDE with the 'Import Ontology Objects' wizard open. The wizard is in the 'Database Import' step, which is used to build ontology information based on database metadata. The wizard has several options checked, including 'Enable smart capitalization', 'Apply capitalization to all text in object names not just first letter', and 'Force separated words to be joined'. The connection parameters are set to 'mysql' driver, 'jdbc:mysql://10.1.2.28:3306/nwdb' URL, 'windows' user, and a masked password. A 'Test Connection...' button is visible. The background shows the Eclipse Package Explorer with a project named 'mapping-sample' containing an 'artifacts' folder with files like 'nw-ontology.owl'.

✓ Choose source type
✓ Provide connection parameters

Select the Tables and Columns

- ✓ Select the Tables and Columns you wish to include to seed your ontology
- ✓ Relationships between tables are recognized and the appropriate owl statements are created

Import Ontology Objects

Database Import
Select desired objects. Adjust object types if necessary

Prop/Class
Select all
Deselect all

Nwcustomers
 address
 city
 companyname
 contactname
 contacttitle
 country
 customerid
 fax
 phone
 postalcode
 region
 Nwemployees
 address
 birthdate
 city
 country
 employeeid

< Back Next > Finish Cancel

Augmenting the Ontology

The screenshot displays the Metatomix ontology editor interface. On the left, the 'Classes' panel shows a hierarchy starting with 'owl:Thing' and including subclasses like 'Nwcustomers', 'Nwemployees', 'Nworderdetails', 'Nwords', 'Nwproducts', 'Nwshippers', and 'Nwsuppliers'. A 'New Ontology Property' dialog box is open, showing options to add properties to 'Nwcustomers' with a 'Super Property' of 'Root' and a 'Namespace' of 'http://m3t4.com/ont/#'. Below this, the 'Properties' panel shows a list of properties including 'address', 'birthdate', 'categoryid', 'city', 'companyname', 'contactFirstName', 'contactLastName', 'contactname', 'contacttitle', 'country', 'customerid', 'discontinued', and 'discount'. The 'contactLastName' property is selected, and its details are shown in the 'Property Details' panel, including its name, comment, type (Datatype), and various flags like Functional, Symmetric, Inverse Functional, and Transitive.

- ✓ Ontology seeded
- ✓ Add properties
 - ✓ contactLastName
 - ✓ contactFirstName

The New Ontology

Viewer

View Details

Currently displaying 27 nodes. Add nodes to the view using the drop-down below.

Add Node: Nworders

Viewer

View Details

Currently displaying 27 nodes. Add nodes to the view using the drop-down below.

Add Node: Nworders

Viewer

View Details

Currently displaying 27 nodes. Add nodes to the view using the drop-down below.

Add Node: Nworders

Overview Imports Metadata Classes Properties Individuals Source Viewer SPARQL

Overview Imports Metadata Classes Properties Individuals Source Viewer SPARQL

Overview Imports Metadata Classes Properties Individuals Source Viewer SPARQL

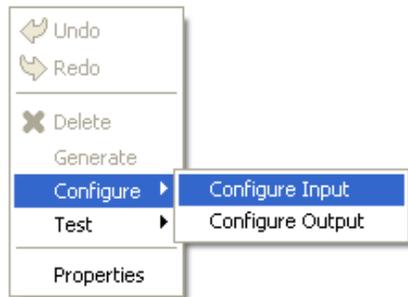
Overview Imports Metadata Classes Properties Individuals Source Viewer SPARQL

✓ Graph view of the ontology

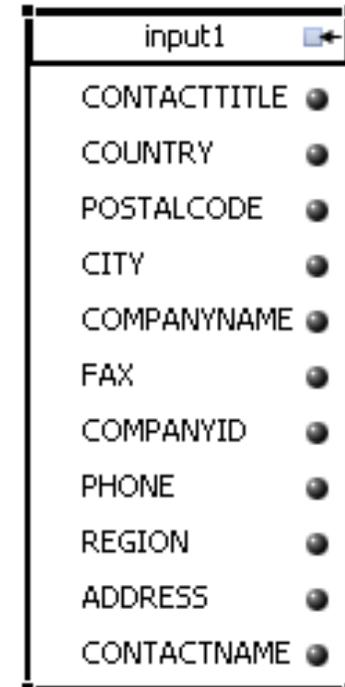
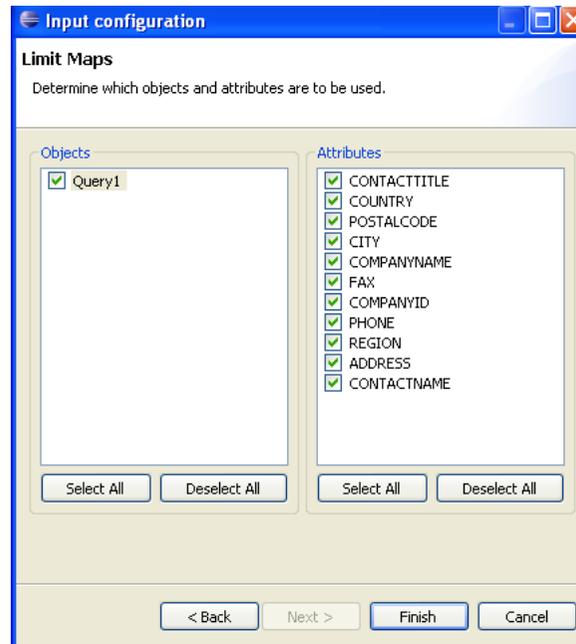
Eclipse-based Mapping Tool

- Allows mapping from relational schemas to an ontology in a click-and-drag paradigm
- Generates xml-based mapping file – executed at runtime for real-time RDF generation
- Test mappings before deployment
- Multiple functions available to format URI's, split strings, merge fields, etc.

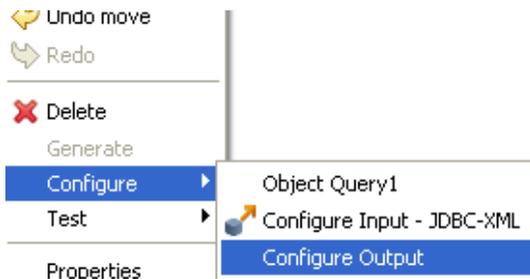
Configure the Input



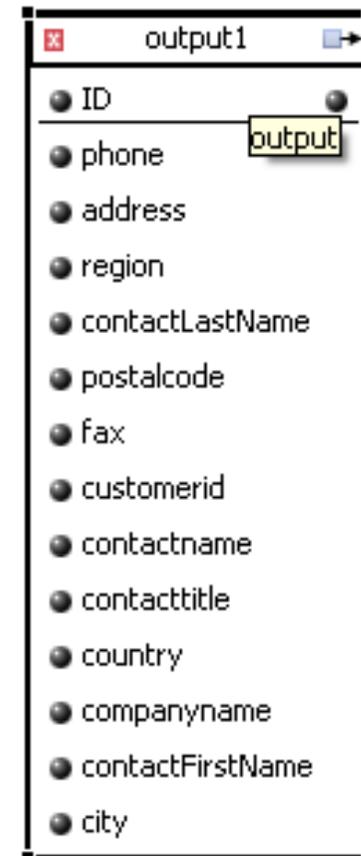
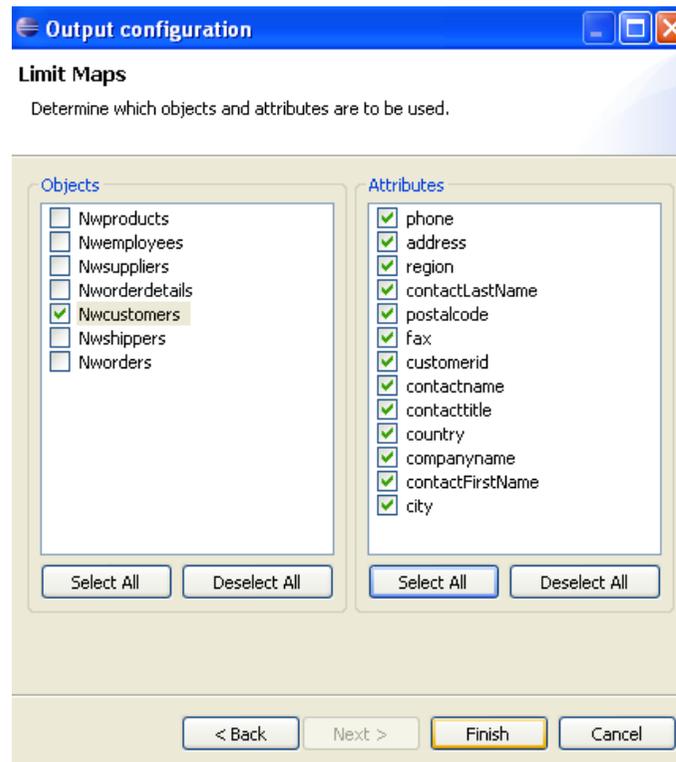
- ✓ To create a mapping, first create an input
- ✓ Select the columns you want to map



Configure the Output



- ✓ Next, create an output, the ontology
- ✓ Select the entities and properties involved in the mapping



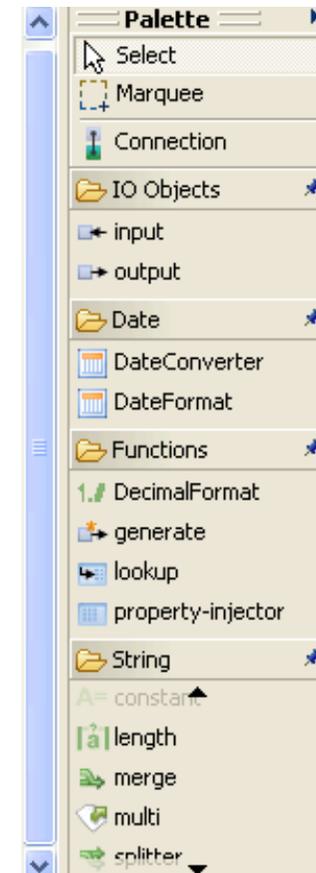
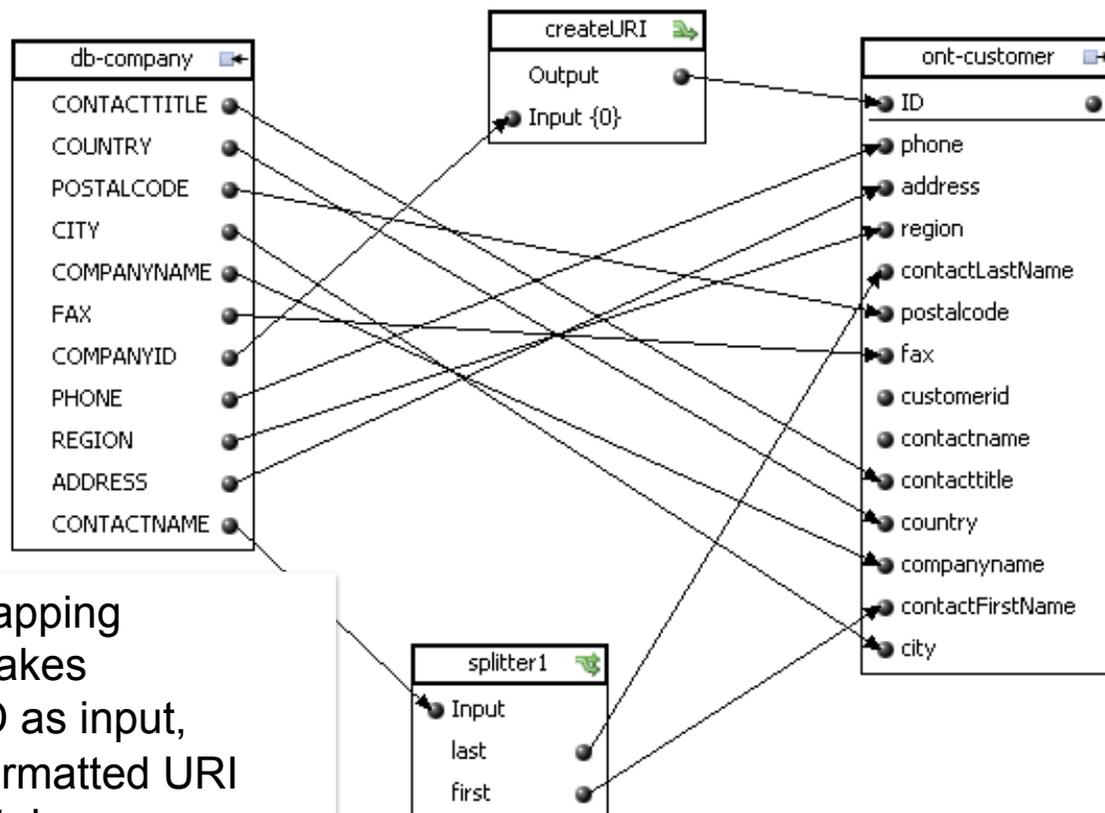
Create the Mappings

The screenshot shows a mapping tool interface. On the left, there is an 'input1' list with fields: CONTACTTITLE, COUNTRY, POSTALCODE, CITY, COMPANYNAME, FAX, COMPANYYID, PHONE, REGION, ADDRESS, CONTACTNAME. In the center, a 'createURI' function is shown with an 'Output' field containing 'Input {0}'. On the right, an 'output1' list contains fields: ID, phone, address, region, contactLastNan, postalcode, fax, customerid, contactname, contacttitle, country, companyname, contactFirstNar, city. A 'Palette' on the right side lists various functions like Select, Marquee, Connect, IO Object, input, output, Date, DateConv, DateForm, Functions, DecimalFc, generate, lookup, property-, String, Aa case, A= constant, length, merge, multi. A 'Regex Function Configuration' dialog box is open, showing a 'Regex Configuration' section with a text area containing a regex pattern: `(?:\s*)([a-zA-Z]+)(?:\s*[,]\s*)([a-zA-Z]+)(?:\s*$)`. Below the text area, the 'Names' field contains 'last; first'. The dialog also has 'Add Pattern', 'Delete', 'Up', 'Down', 'Help', 'Finish', and 'Cancel' buttons.

- ✓ Configure Merge Function to specify URI pattern
- ✓ URI generated from companyID (e.g “http://m3t4.com/customer{0}”)

- ✓ Configure Splitter Function to split contactName (using regex)
 - ✓ contactLastName
 - ✓ contactFirstName

Final Mapping



- ✓ Final Mapping
- ✓ Merge takes companyID as input, output is formatted URI
- ✓ Splitter takes contactName as input, splits name and outputs last, first

Source Record

```
<?xml version="1.0" encoding="UTF-8"?>
<JDBC_MSG>
  <Transaction ID="61">
    <SQL rowcount="1" exception="false">
      <Statement>select * from nwCustomers p where p.....</Statement>
      <ResultSet>
        <Row>
          <Column name="COMPANYID" value="1234" type="varchar" precision="8" scale="0"
          <Column name="COMPANYNAME" value="Metatomix" type="varchar" precision="8" scale="0"
          <Column name="CONTACTNAME" value="Batla, Samir" type="varchar" precision="6" scale="0"
          <Column name="CONTACTTITLE" value="SE" type="smalldatetime" precision="16" scale="0"
          <Column name="ADDRESS" value="3 Allied Drive" type="varchar" precision="9" scale="0"
          <Column name="CITY" value="Dedham" type="varchar" precision="100" scale="0"
          <Column name="REGION" value="Northeast" type="varchar" precision="255" scale="0"
          <Column name="POSTALCODE" value="02026" type="varchar" precision="255" scale="0"
          <Column name="COUNTRY" value="USA" type="varchar" precision="255" scale="0"
          <Column name="PHONE" value="7819076700" type="varchar" precision="100" scale="0"
          <Column name="FAX" value="7819076701" type="varchar" precision="100" scale="0"

        </Row>
      </ResultSet>
    </SQL>
  </Transaction>
</JDBC_MSG>
```

The RDF

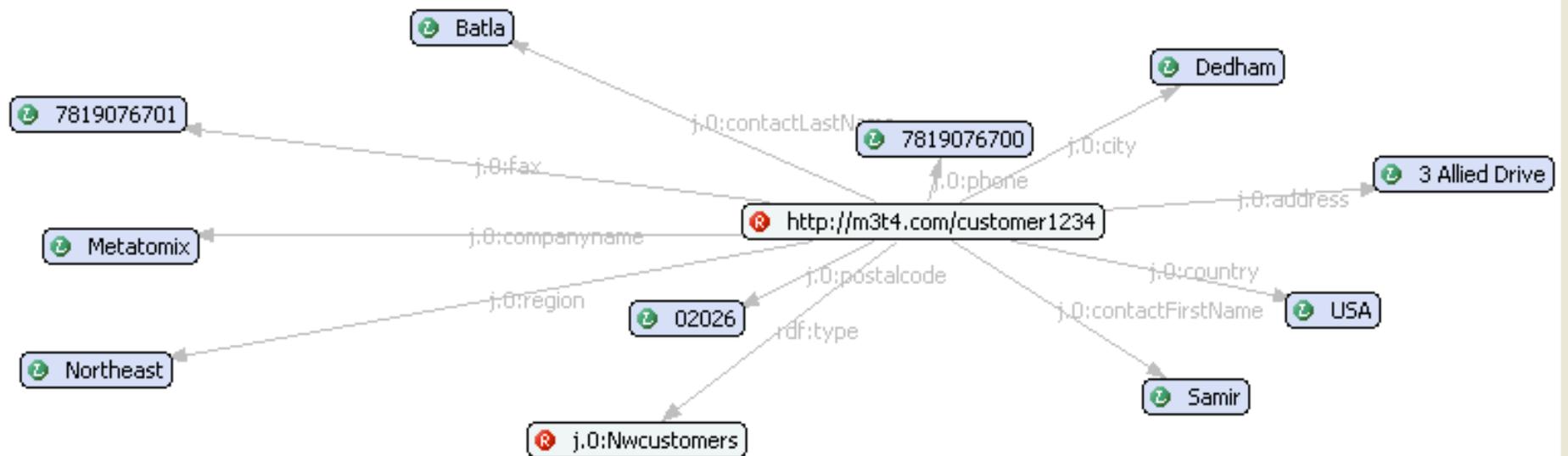
```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:j.0="http://m3t4.com/ont/#">
  <j.0:Nwcustomers rdf:about="http://m3t4.com/customer1234">
    <j.0:contactFirstName rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Samir</j.0:contactFirstName>
    <j.0:fax rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >7819076701</j.0:fax>
    <j.0:contactLastName rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Batla</j.0:contactLastName>
    <j.0:region rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Northeast</j.0:region>
    <j.0:companyname rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Metatomix</j.0:companyname>
    <j.0:postalcode rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >02026</j.0:postalcode>
    <j.0:country rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >USA</j.0:country>
    <j.0:address rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >3 Allied Drive</j.0:address>
    <j.0:phone rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >7819076700</j.0:phone>
  </j.0:Nwcustomers>
</rdf:RDF>
```

generated.rdf Triples RDF Graph

The RDF

Currently displaying 11 out of 11 statements.

Add Node:

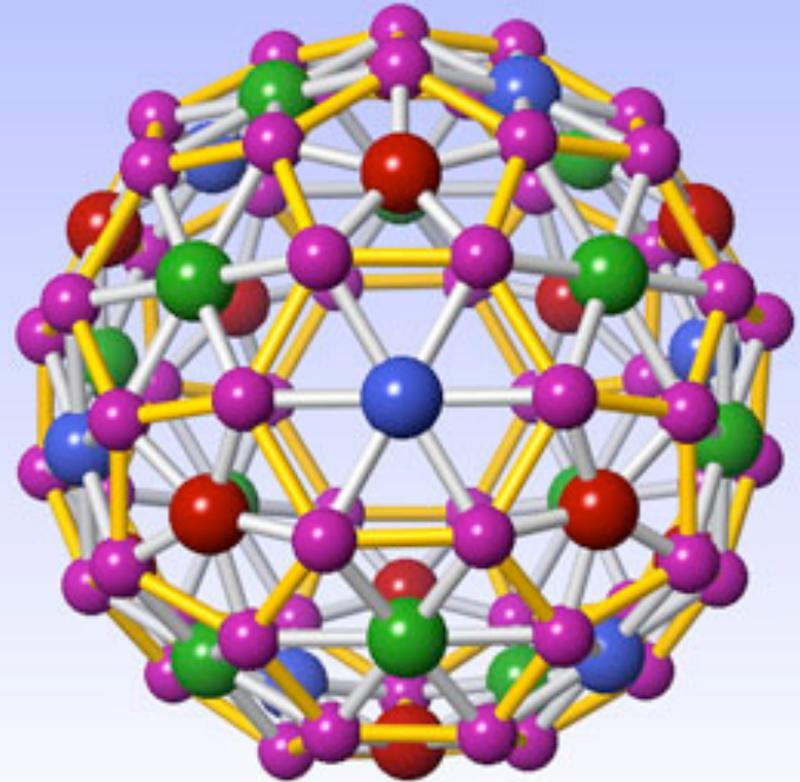


generated.rdf Triples RDF Graph

Metatomix Discovery

Tools for Understanding and Leveraging Unstructured Data

- Bridges the gap between structured and unstructured data
- Provides an integrated suite of tools for the analyst and knowledge worker
- Adds automated categorization of text documents to the Mtx-Runtime Platform
- Translates Information into Knowledge



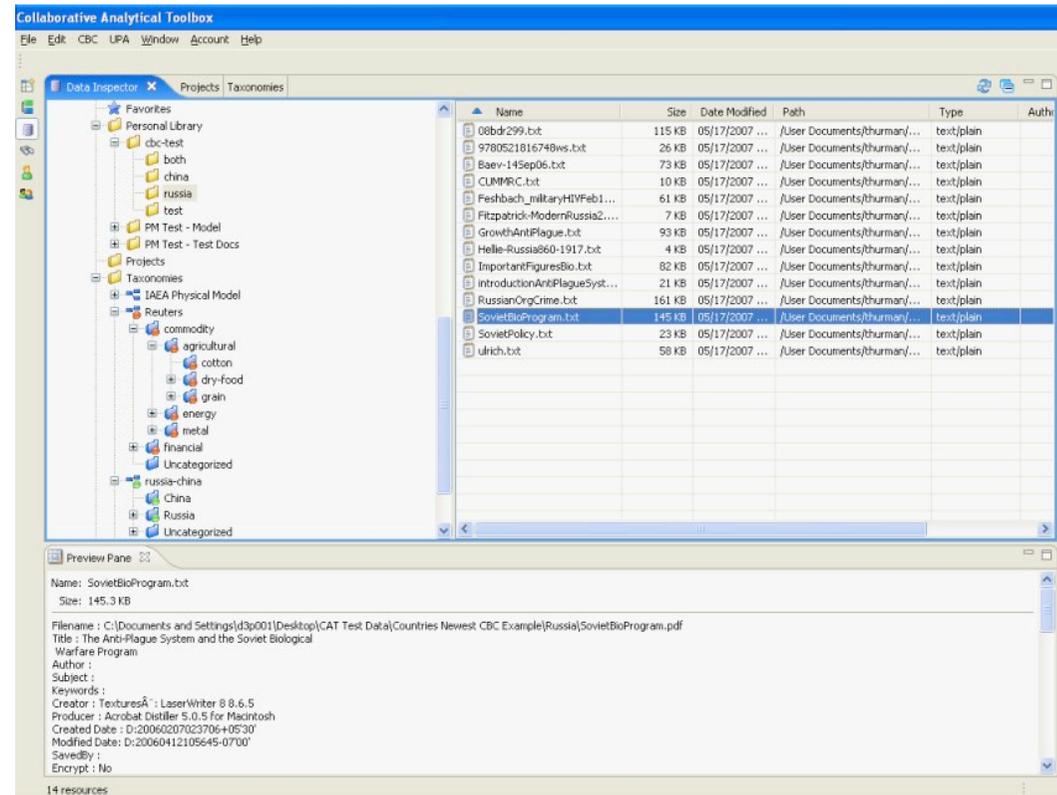
Metatomix Discovery - *Document Analytics*

- ***Mtx Document Studio*** - Workgroup-enabled document organization & analytic framework
- ***Mtx Visual Document Analysis*** - Unstructured text visualization & analysis
- ***Mtx Document Clustering and Categorization*** - User-driven and automated document clustering & visualization
- ***Mtx Parsing Agent*** - Document parsing and transformation
- ***Mtx Semantic Document Analysis*** - Semantic extraction, visualization & enhancement with support for enterprise RDF stores
 - ***Temporal Display*** - Web-enabled temporal data presentation (Simile)
 - ***Geospatial Display*** - Geospatial data presentation (Google Earth)

Metatomix Document Studio

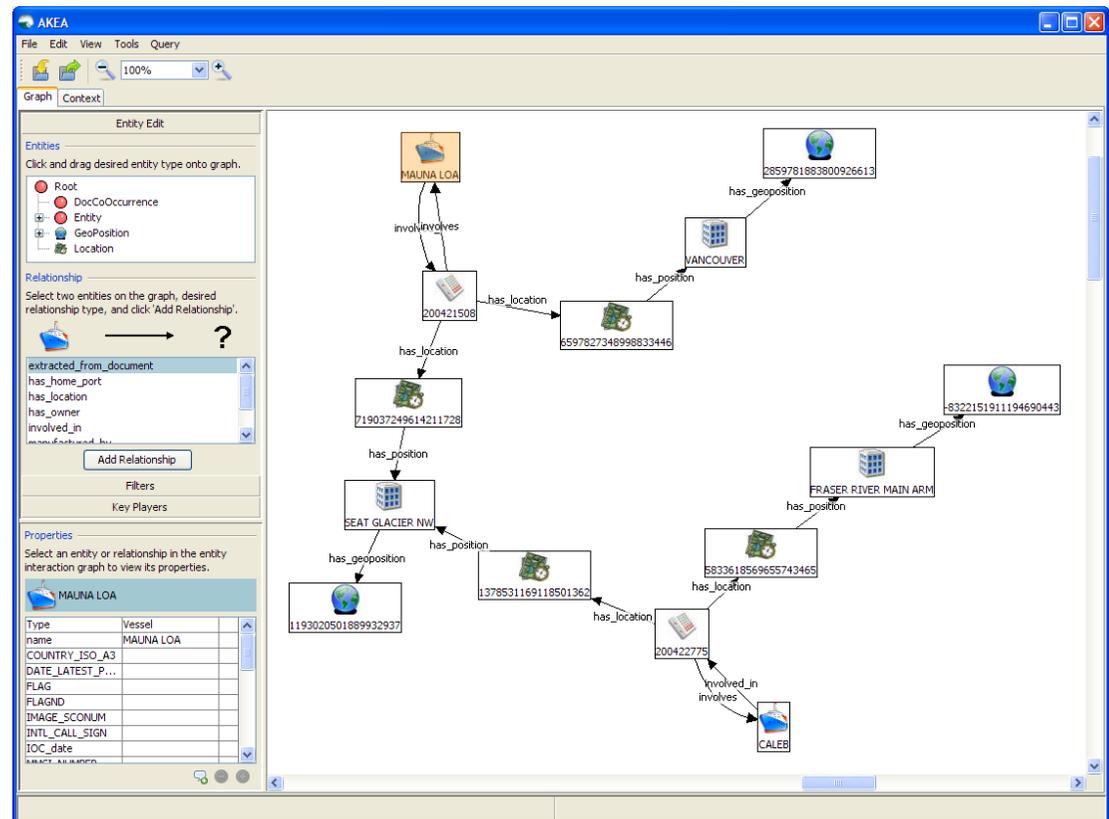
Provides a desktop analytic framework for:

- Harvesting information from documents, data feeds, Internet resources
- Building and organizing an information repository
- Collaboration among users and teams, sharing data, templates, annotations, etc.
- Access, search and analysis of documents in the repository
- Interoperability with the Mtx Runtime Platform
- Plug-in analytical tools



Mapping in Mtx Discovery

- Enables analysis of linkages between entities extracted from documents (individuals, organizations, locations, etc.)
- Integrates information from structured and unstructured data sources
- Identify relationships between concepts
- Network analysis tools support identification of key players and strengths of relationships
- RDF triple store integration with Mtx Runtime Platform



Structured Data Import Into Discovery

- Mapping structured data from a database, csv file or XML document
- Original data shown in table on top
- Entity and property info from ontology shown on lower left
- Created entities and relationships shown on lower right
- Mapping shown in floating window

The screenshot shows the Triplet Mapper application interface. At the top, the source is identified as a CSV file. Below this, a table displays the source data with columns: TRANSIT_CODE, VESSEL_REGISTRY_TYPE, VESSEL_REGISTRY_NUMBER, and VESSEL_SUB. The data rows are as follows:

TRANSIT_CODE	VESSEL_REGISTRY_TYPE	VESSEL_REGISTRY_NUMBER	VESSEL_SUB
2004946080	L	9169158	P
2004944235	L	9141235	P
2004945219	L	9153496	P
2004945224	L	9153496	P
2004944338	L	9159402	P
2004944943	L	9140229	P
2004943692	L	7915644	P

The interface is divided into several sections:

- Entity Information:** Shows a tree view of ontology entities. The 'Vessel' entity is selected.
- Properties:** A table showing the mapping of source columns to Vessel properties. A floating window is open over this table.
- Created Entities And Relationships:** Shows the resulting 'Vessel' entities and relationships like 'involved_in' and 'has_position'.

The floating window displays the following mapping table:

Property	Column	Sample V...
REGISTRY_...	VESSEL_RE...	9169158
length	VESSEL_LEN...	267.99539
name	VESSEL_NAME	LT UNITY
weight	VESSEL_DE...	62700
width	VESSEL_BEAM	39.9796

At the bottom of the interface, there are navigation buttons: '< Back', 'Next', and 'Close'. A 'Run Mapping' button is also present in the 'Created Relationships' section.

Additional Mapping Dialogs

Triplet Mapper

_TYPE	VESSEL_REGISTRY_NUMBER	VESSEL_SUB_TYPE	VESSEL
	9169158	P	3PCD9
	9141235	P	3FCI7
	9153496	P	3FDQ8
	9153496	P	3FDQ8
	9159402	P	3FFS7
	9140229	P	3FGF6
	7915644	P	3FHO5
	8709119	P	3FHW8
	9050292	P	3FST4
	9116591	P	3FTL6
	9108594	P	3FTY4
	9123635	P	3FYK6
	7801609	P	3FYW2

Entity Information

Vessel

Properties

- FLAGND
- IMAGE_SCONUM
- INTL_CALL_SIGN
- IOC_date
- MMSI_NUMBER
- NAME_SHIP
- PHOTO_GRADE_COE
- REGISTRY_NBR_LLO
- REGISTRY_NBR_OFF
- SCONUM
- SHIP_ACTIVITY_COE
- SHPCAT
- SHPTYP

*Use mouse to select columns

Columns: 3

< Back Next Close

Triplet Mapper

Selected Columns

VESSEL_REGISTRY_NUMBER
9169158
9141235
9153496
9153496
9159402
9140229
7915644
8709119
9050292
9116591
9108594
9123635

Data Mapping

Vessel

Properties:

Property	Column
<input type="checkbox"/> MMSI_NUMBER	
<input type="checkbox"/> monetary_value	
<input type="checkbox"/> name	
<input type="checkbox"/> NAME_SHIP	
<input type="checkbox"/> PHOTO_GRADE_CODE	
<input type="checkbox"/> REGISTRY_NBR_LLOYDS	
<input checked="" type="checkbox"/> REGISTRY_NBR_OFFICIAL	
<input type="checkbox"/> SCONUM	
<input type="checkbox"/> serial_number	
<input type="checkbox"/> SHIP_ACTIVITY_CODE	
<input type="checkbox"/> SHPCAT	
<input type="checkbox"/> SHPTYP	
<input type="checkbox"/> status	

Check properties to use in coreference

< Back Done Close

Running the mapping

The screenshot displays the AKEA Query Manager interface. The top section, titled 'Query', contains controls for data import and mapping execution. The 'Data' field is set to 'cg_small.csv'. Below this, there are buttons for 'Run Mapping', 'Apply Existing Mapping', 'Load Existing Mapping', and 'Create New Mapping'. The 'Properties' panel on the left shows details for 'OCEAN HOPE II', including its type (Vessel) and various attributes like name, registry number, and length. The main workspace displays a graph visualization with nodes and relationships. The central node is 'VANCOUVER'. It is connected to three 'Location' nodes via 'has_position' relationships. One 'Location' node is further connected to a numeric node '2004946118' via a 'has_location' relationship. This numeric node is connected to an 'OCEAN HOPE II' node via 'involves' and 'involved_in' relationships. At the bottom of the workspace, there are three buttons: 'Add All To Workspace', 'Add Selected Items To Workspace', and 'Add All to Triple Store'.

- Mapping is run and then displayed in the query manager
- Items can then be added to the workspace
- Items can also be added directly to the triple store

Items Are Shown In The Workspace

The screenshot shows the Desktop Triple Store interface. On the left, there is a sidebar with the following sections:

- Desktop Triple Store**: # of Entities: 438, # of Relationships: 362
- Entity Edit**: A tree view showing 'Entity' as the root, with sub-entities 'Account', 'Artifact', 'Vessel', and 'Document'. 'Artifact' is selected.
- Relationship**: A list of relationship types including 'involves', 'related_to', 'verb_body', 'verb_change', 'verb_cognition', and 'verb_communication'. 'involves' is selected.
- Filters** and **Key Players**: Empty sections.
- Properties**: A table for the selected entity 'OCEAN HOPE II':

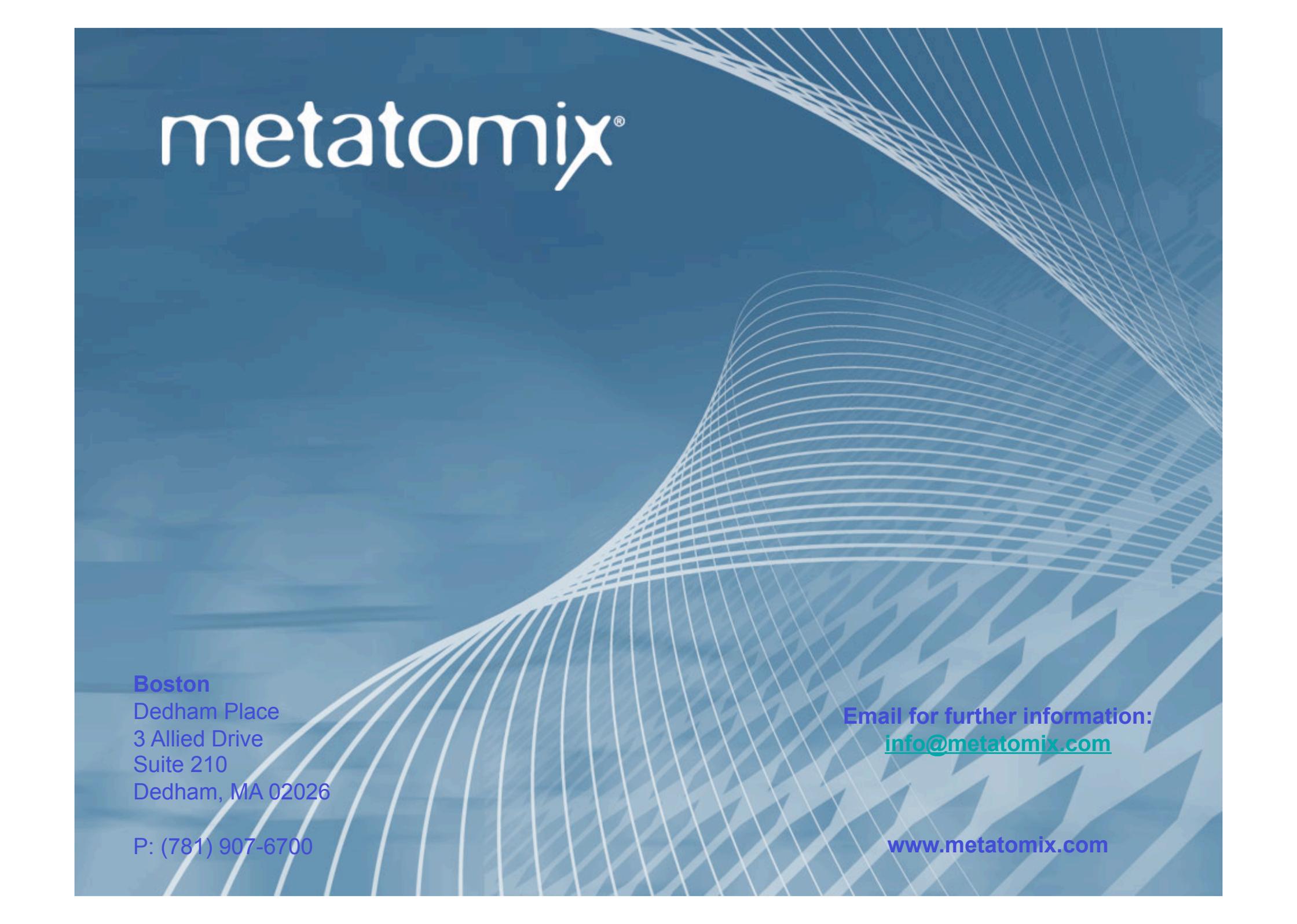
Type	Vessel
name	OCEAN HOPE II
COUNTRY_ISO_A3	
DATE_LATEST_P...	

The main workspace displays a graph of relationships:

- 2004946118** (Document icon) is *Involved_in* and *Involves* **OCEAN HOPE II** (Vessel icon).
- 2004946118** *has_location* **Location** (Location icon).
- 2004945224** (Document icon) *has_location* **Location** (Location icon).
- Location** (Location icon) *has_position* **VANCOUVER** (City icon).
- GeoPoint** (GeoPoint icon) *has_geoposition* **VANCOUVER** (City icon).
- Location** (Location icon) *has_position* **VANCOUVER** (City icon).
- VANCOUVER** (City icon) is connected to other entities via unlabeled arrows.

Discovery Entity and Relationship Mapping

- Integrates information from structured and unstructured data sources
- Supports most common document formats (PDF, MS Word, plain text, etc), plus web pages, blogs, social sites, text data in database fields, etc.
- Maps the extracted concepts to a domain ontology

The background of the slide is a deep blue color. It features a complex graphic of white, curved lines that sweep across the frame from the bottom left towards the top right, creating a sense of motion and depth. The lines are thin and densely packed in some areas, while more sparse in others. In the upper right quadrant, there is a faint, light blue geometric pattern consisting of interconnected lines forming a grid-like structure.

metatomix[®]

Boston
Dedham Place
3 Allied Drive
Suite 210
Dedham, MA 02026

P: (781) 907-6700

Email for further information:
info@metatomix.com

www.metatomix.com