

# Slideset 2 of

## “E-Commerce Applications of Semantic Web Services”

**Benjamin Grosf**

MIT Sloan School of Management,  
<http://ebusiness.mit.edu/bgrosf>

## EC '04 Conference Tutorial

5<sup>th</sup> ACM Conference on Electronic Commerce, May 17, 2004, New York City

Version Date: April 29, 2004

Copyright 2000-2004 by Benjamin Grosf. All Rights Reserved.

# Sequence Outline of Tutorial

- Introduction & get acquainted
- Overview of Core Technologies of the New Generation Web
  - XML; Web Services; Semantic Web; Semantic Web Services
- Business Process Automation; B2B; EAI; Agents; Standards Role
- Overview of **Research Agenda**; incl. SWS application scenarios
- End-to-end **E-Contracting** as business application of SWS
  - SweetDeal rule-based approach, manufacturing SCM example
- (BREAK midway: about here.)
- More depth on Rules and KR for SWS, incl. RuleML
  - Requirements; uses; maturity; rule-based SWS
  - New **Fundamental KR Theory**, incl. Description LP
- **Financial Info & Reporting**: ECOIN ontologies/contexts integration
- **Trust Management Policies**: rules & delegation in authorization
- **SWS Research Directions**
- **SWS E-C Adoption Roadmap; Market Evolution**

# Overview

- Web software and data infrastructure is undergoing a sea change from HTML to XML... →→
- More detailed descriptions of products/services
- ...Exchanged more automatically
- ...With greater depth of understanding
- → → Radical change in e-markets
  - cross-enterprise integration
  - ...of e-business processes
  - example: contracting

# What is XML

- **eXtensible Markup Language**
- **scheme for defining tags to represent meta-data,**
  - data about the data, intertwined with the data itself
  - HTML has a fixed set of tags, e.g.,
    - <h1> header, <b> boldface, <a> anchor for links
- **a given XML document uses a set of tags defined by a community of potential communicators**
  - industry-specific vocabularies, e.g.,:
    - electronic products, 3-D modelling, ...
  - generic vocabularies for transactions, agent talk, ...
  - “XML is like HTML, where you make up your own tags.”
- **XHTML: new version of HTML, is in XML**
  - The entire Web will soon be (almost all) XML!!!!!!

Copyright 2000-2004 by Benjamin Grosz. All Rights Reserved.

# What is XML (continued)

- **HTML describes visual presentation**
  - XML does NOT
- **XML is a meta-language used to define other domain- or industry specific languages**
- **XML addresses “structuring” of data**
  - Declare to be: a retail price, a sales tax, a book title, ...
- **XML documents by themselves, bear no semantics**

## Sample HTML from SEC's Edgar web site [slide courtesy of Stuart Madnick]

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<html>
<head>
<title>Partes&#174; FreeEDGAR: Free Real-Time SEC EDGAR Filings</title>
<BODY topmargin=18 leftmargin=6 bgcolor="#ffffff" link="#0000ee" VLINK="#551A8B" ALINK="#ff0000">
</TABLE>
<pre><font size=2>
                1998          1997          1996
                -----          -----          -----
NET REVENUES      $26,273      $25,070      $20,847
                -----          -----          -----
Cost of sales      12,144        9,945        9,164
                -----          -----          -----
OPERATING INCOME      8,379        9,887        7,553
                -----          -----          -----
NET INCOME        $ 6,068        $ 6,945        $ 5,157
</font></pre>
<!-- end company info -->
<!-- begin page footer -->
<table cellpadding=0 cellspacing=0 border=0>
<tr><td align=left valign=middle width=455 nowrap height=20>
</td></tr>
<tr><td align=left valign=top nowrap width=455>
<font size=1 face="helvetica,arial">
<p>Copyright &#169;1999 Partes Corporation.<br></font></td></tr></table>
<!-- end page footer -->
</td></tr></table>
</BODY>
</HTML>
```

# XML Example: Companies Domain

[slide courtesy Stuart Madnick and Chris Dellarocas]

<Company>

<CompanyID=642>

<Ticker>WCOM</Ticker>

<Company\_name>MCI WorldCom Inc</Company\_name>

<Market\_cap>\$100,721,720</Market\_cap>

<Competitor\_reference=542/>

<Competitor\_reference=573/>

</Company>

<Company>

<CompanyID=542>

<Ticker>T</Ticker>

<Company\_name>AT&T Corp</Company\_name>

<Market\_cap>\$114,815,359</Market\_cap>

<Competitor\_reference=642/>

</Company>

# DTD fragment for XML Example

[slide courtesy Stuart Madnick and Chris Dellarocas]

- **DTD = Document Type Definition:** Defines the syntax of the tags, i.e., the structure of the data.

...

```
<!ELEMENT Company (CompanyID, Ticker,  
Company_name, Market_cap, Competitor_reference*)>
```

```
<!ELEMENT CompanyID #PCDATA>
```

```
<!ELEMENT Competitor_reference CompanyID>
```

\* = zero, 1, or several



# XML Example: Books Domain

## -- Document and DTD

[slide courtesy Stuart Madnick and Chris Dellarocas]

```
<?xml version="1.0" ?>
```

```
<Book Author="Anonymous">
```

```
  <Title>Sample Book</Title>
```

```
  <Chapter id="1">
```

This is chapter 1. It is not very long or interesting.

```
  </Chapter>
```

```
  <Chapter id="2">
```

This is chapter 2. Although it is longer than chapter 1,  
 it is not any more interesting.

```
  </Chapter>
```

```
</Book>
```

```
<!DOCTYPE Book [
```

```
  <!ELEMENT Book (Title, Chapter+)>
```

```
  <!ATTLIST Book Author CDATA #REQUIRED>
```

```
  <!ELEMENT Title (#PCDATA)>
```

```
  <!ELEMENT Chapter (#PCDATA)>
```

```
  <!ATTLIST Chapter id ID #REQUIRED>
```

Copyright 2000-2004 by Benjamin Grosz. All Rights Reserved.

# Extensible Stylesheet Language Transformations (XSLT)

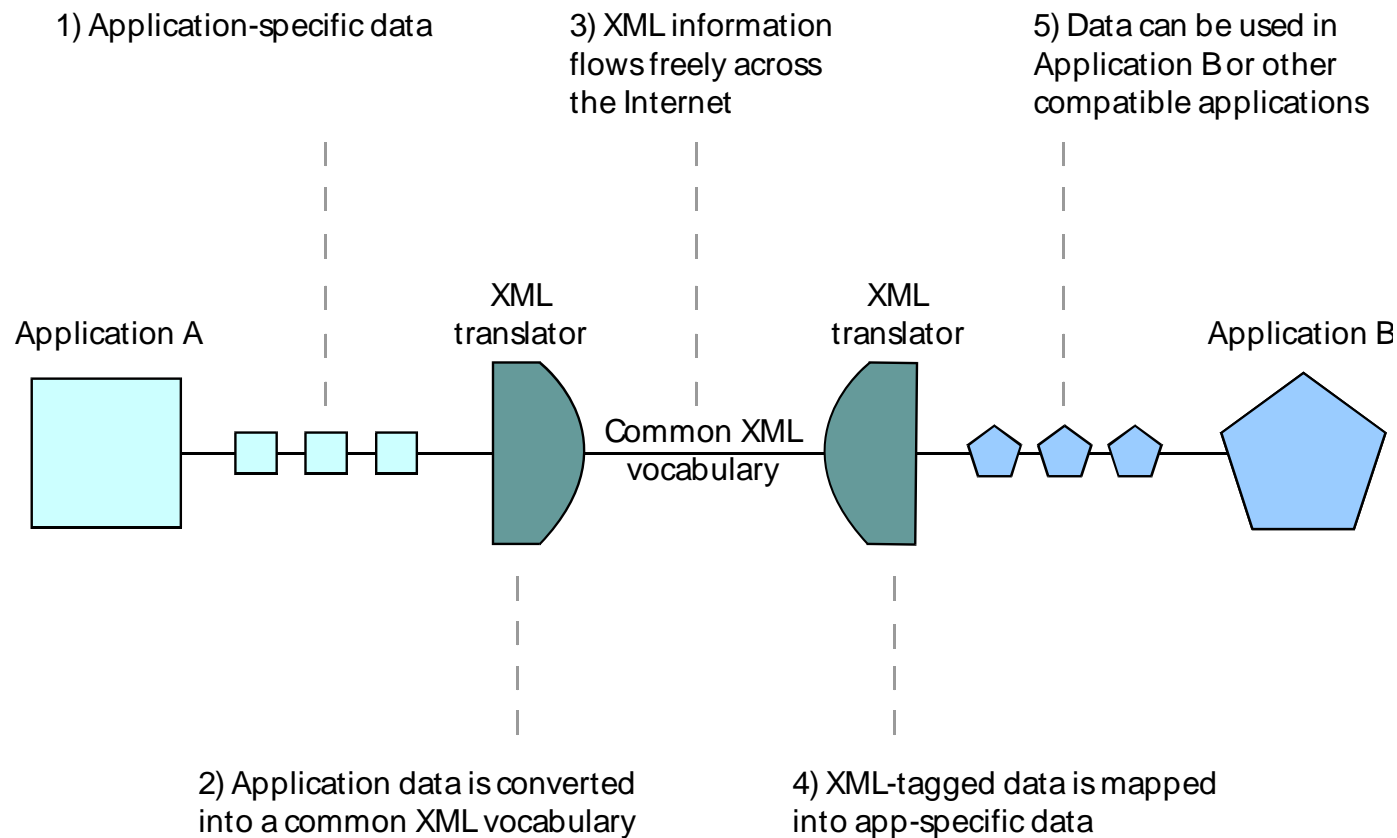
- **Successor to Extensible Stylesheet Language (XSL)**
- **XSL Transformations (XSLT) is an XML-based language that enables you to transform one class of XML document to another.**
- **XSLT offers tremendous flexibility for presenting and exchanging data between disparate devices and business systems.**
  - **with XSLT style sheets, you can dynamically transform an XML purchase order from one schema to another before sending the order to a supplier.**
  - **dynamically transform an XML document so it can be rendered on a variety of Internet-enabled devices, including handheld PCs and TV set-top boxes.**

# XML will help data interchange over the Web

- **Web sites store a proliferation of different document types**
  - HTML pages, Word documents, Powerpoint slides, etc.
  - Today, browsers use appropriate “plug-in” for EACH document type to DISPLAY it
- **XML will serve as an “interchange format” for different kinds of data**
  - Aids automated UNDERSTANDING of imported data,
  - ... thence general PROCESSING, not just displaying
  - Web servers and clients need only “speak” XML in order to import arbitrary types of data
  - Web servers will do lots of translation to and fro XML
    - componentization of app's; wrappers for legacy app's
      - similar to use for app integration intra-enterprise (EAI)

# Generic XML Data Interchange

[slide courtesy of Chris Dellarocas and Stuart Madnick]



Source: Forrester Research  
Copyright 2000-2004 by Benjamin Grosf. All Rights Reserved.

# Pricebot: Example Scenario of using XML

- Web “softbot” searches a variety of Web sites to find best prices for a used car
- Today...
  - Each site is storing price data using a variety of formats geared towards human readers
    - formatted text, tables, pdf documents, etc.
  - It is very difficult for the softbot to make sense and compare the information

# Pricebot: XML facilitates Web commerce

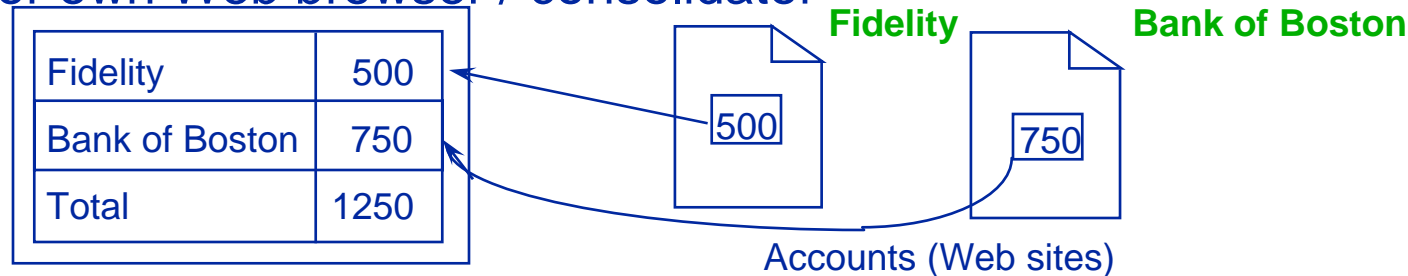
## ■ With XML...

- each site will publish price data using the XML format
- even if different sites use different terms for same notion (e.g. price and cost), XML header will contain information for detecting this
- if the robot can understand XML, it will be able to make sense of and compare the data

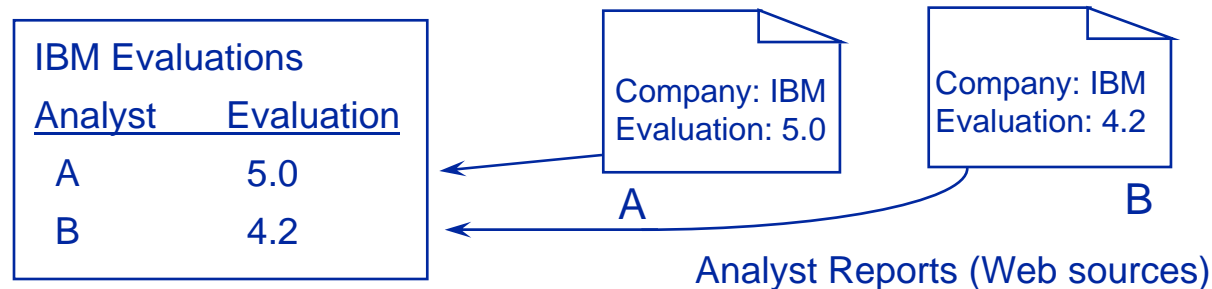
## ■ Incentive issues to publish the data in XML?

# More Sample App's of XML Web Data [slide courtesy of Stuart Madnick]

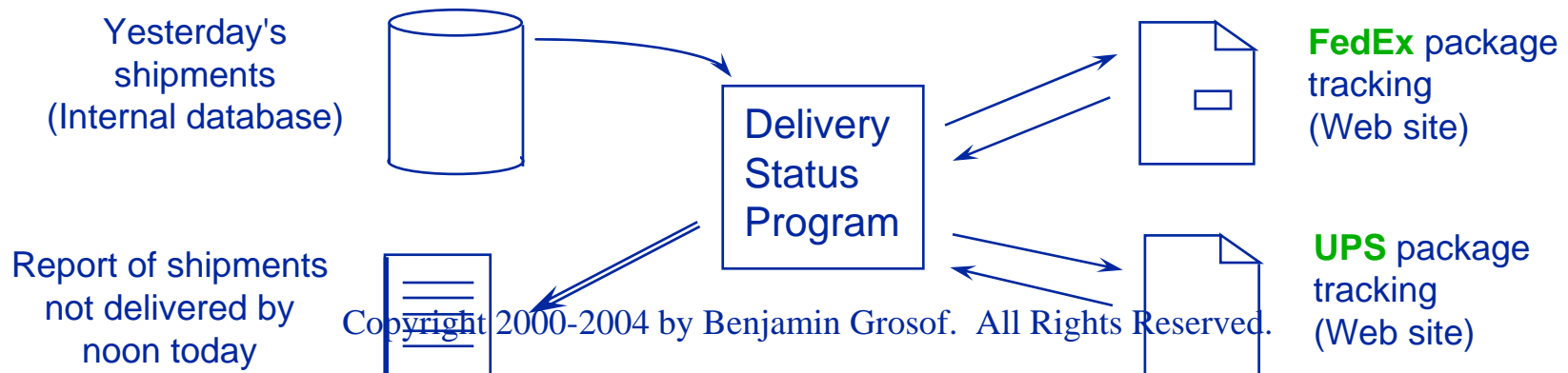
- Automate Extraction of data from specific Web sites into user tool, like Excel, or own Web browser / consolidator



- Automatically Select and Consolidate information across Web sites



- Integrate Internet / Intranet / Client Server networks for internal operations



# XML Semantics: The Road Ahead

- XML in itself provides only syntax not semantics
- Still need agreements about the meaning of the tags
- Today: by “convention”: DTD/Schema + text documentation
  - A community of users agree to use a particular Schema to describe a particular kind of domain data.
- Want automated semantics
  - Ontologies: vocabularies
  - Knowledge Representation: sanctioned inferences
- Current R&D: industry-wide/partner agreements, mechanisms for such agreements
- The *Semantic Web*; Agent-Agent Communication
  - Next fundamental steps in standards and technology



# The Semantic Web

The 1st generation, the Internet, enabled disparate machines to exchange data.

■ The 2nd generation, the World Wide Web, enabled new applications on top of the growing Internet, making enormous amounts of information available, in human-readable form, and allowing a revolution in new applications, environments, and B2C e-commerce.

■ The next generation of the net is an “agent-enabled” resource (the “**Semantic Web**”) which makes a huge amount of information available in machine-readable form creating a revolution in new applications, environments, and B2B e-commerce.

...by enabling “agent” communication at a Web-wide scale.

## **Some Commercial Applications of what Agents may soon Communicate in KBEM**

- **Bids** in auctions and reverse auctions
- **Orders** in supply chain (or B2C)
- **Contracts/Deals/Proposals/RequestsForProposals**
  - prices; product/service descriptions; refunds, contingencies
- **Buyer/Seller interests, preferences, capabilities, profiles**
  - recommender systems; yellow pages; catalogs
- **Ratings, reputations; customer feedback or problems**
- **Demand forecasts** in manufacturing supply chain
- **Constraints** in travel planning
- **Creditworthiness, trustworthiness, 3rd-party recommendations**
- ***Industry-verticals: computer parts, real estate, ...***

# OPTIONALS

# Early History of the WWW

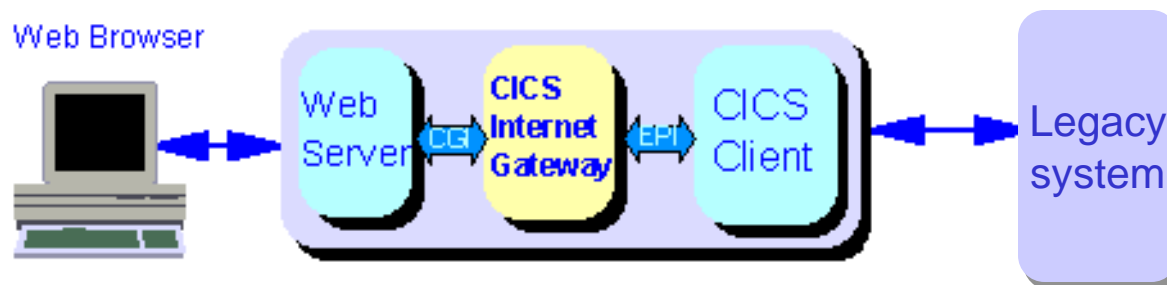
- 1972 DARPA starts research leading to the Internet
- 1989 Tim Berners-Lee proposes a networked Hypertext system for CERN
- 1990 The name “World-Wide Web” is coined
- 1992 The first (non-graphical) browser is released by CERN as freeware
  - The world has 50 servers
- 1993 Marc Andreessen and Eric Bina write Mosaic at NCSA (Nat. Center for Supercomputer Appl., Illinois)
  - We have 250 servers

## History of the WWW (continued)

- 1994 Jim Clark (of Silicon Graphics) founds MCC, later Netscape
    - We have 2500 servers
  - 1995 Sun introduces Java
    - There are 73500 servers
  - 1995 Microsoft realizes Web potential and makes 180 degree shift in strategy
  - 1997 shopping becomes major on the Web
  - 1997 XML basics (standards, alpha tools) developed
  - 2000: There are about 2,000,000 www servers
- ... and the rest is history (still being made including by MIT EECS and Sloan!)

# The role of (1<sup>st</sup>-generation) Web technologies in enterprise integration

- Facilitate access to a heterogeneous set of legacy computer systems through a common interface (browser)
- Good for connecting humans to computers
- Don't help so much in making heterogeneous computer systems talk to each other



# HTML specifies appearance not meaning!

- **Web sites store a proliferation of different document types**
  - HTML pages, pdf files, powerpoint slides, etc.
  - to correctly process these documents, browsers need appropriate “plug-ins” for EACH document type
- **Traditional document formats contain information about the appearance of data, not its meaning**
- **This is OK is the only readers of the data are humans**
  - but it makes automatic processing of Web data by computers very difficult

## What needs to be done

- When storing documents on the web, specify not only their appearance, but also their semantics (their meaning!)
- Most promising technology for achieving this:
- XML (eXtended Markup Language)



# The B2B Technology Market

- **Market today:**
  - very dynamic
  - fairly fragmented and confusing
  - ... thus hard to track
  - some big players, lots of smaller/niche players
    - IBM, Oracle, Ariba, Microsoft, CommerceOne, SAP, ...
  - much jockeying; lots of airware
  - lots of consultants
- **open concept; yet advantages of big-ness**
- **still DB-centric largely; XML rising fast**
- **integrating with: CRM, EDI, core intra-enterprise e-biz processes**

# DTDs and XML documents

- One uses XML to define an XML language
- The newly-defined XML language is described in a Document Type Definition (DTD)
- A DTD (a .dtd file) is a text file
- A properly-defined XML document (a .xml file) is another text file that adheres to some particular DTD
- A XML document contains no information for displaying its contents

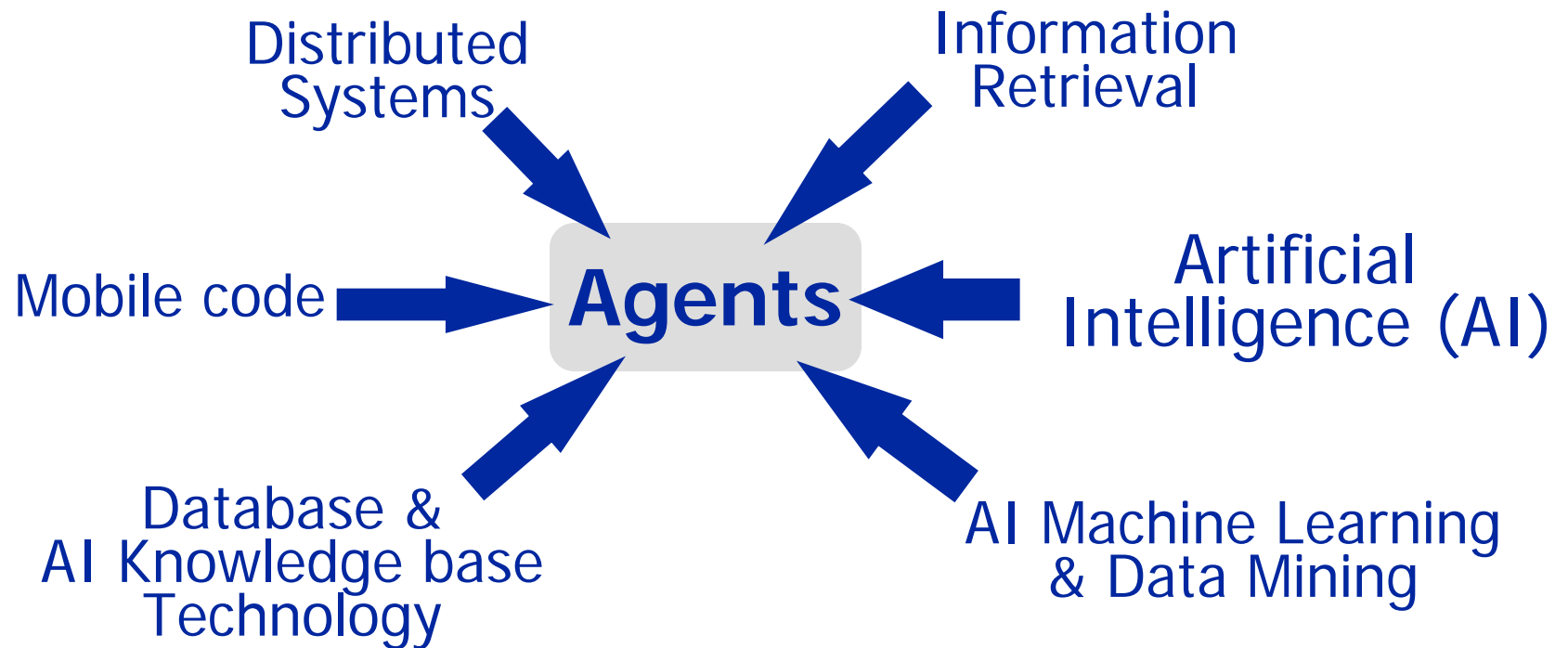
# Some B2B Technologies

- **Overall:**
  - “Infrastructure” is buzz du jour.
  - Typically requires most of what B2C does, plus much more
- **Database integration**
- **Messaging Protocol stack: queues, ack’s, routing, connectors, workflow and biz-process-automation, “business messages” (e.g., RFQ)**
  - Old & being revamped: EDI (Electronic Data Interchange)
  - Newthink: **XML, Agents**
- **Object Oriented software design & architecture:**
  - low-level “brokers” for inter-operability (even cross-prog.-lang.)
  - frameworks of shared code, e.g., by consortia
- **Connectors; data mapping, mediation, extraction**

# What is (agent) communication?

- **Communication means** “communication in a common language”
- **Understanding a “common language” means:**
  - understanding of its vocabulary, i.e., understanding of the meaning of its tokens
  - knowing how to effectively use the vocabulary to perform tasks, achieve goals, effect one’s environment, etc.
  - including knowing what are sanctioned inferences
    - alias: knowledge representation & reasoning
  - translating/mediating between different contexts
- **For software agents, today’s Agent Communication Languages (ACLs) are primarily concerned with the knowledge representation and vocabulary**

# Agents: A system-building paradigm



**Agents**

**2001**

=

**Objects**

**1982**

=

**Structured Programming**

**1974**

Copyright 2000-2004 by Benjamin Grosz. All Rights Reserved.

## **The ideal software agent is ...**

- Autonomous, **taking the initiative as appropriate**
- Goal-directed, **maintaining an agenda of goals which it pursues until accomplished or believed impossible.**
- Situated **in an environment (computational and/or physical) which it is aware of and reacts to.**
- Cooperative **with other agents (software or human) to accomplish its tasks.**
- Communicative **with other agents (human or software)**
- Adaptive, **modifying beliefs & behavior based on experience**

**But many (less than ideal) software agents are simple on some of these dimensions.**