Chapter 2

FOUNDATIONS

Global Database Query and Market-based Resource Allocation

1. OVERVIEW

As the evolution of enterprise integration continues (see Chapter 1), the evolution of the technology for information integration continues. Scope of integration has been and will continue to be the driving force of the evolution and the determinant of the technology. The Enterprise Collaboration results developed in this book is a part of the evolution, and hence should be put in the larger context of the field of enterprise information integration. For the purpose of the research, we recognize two particular foundations based on which the TSCM results have been developed. The first is Global Database Query, of which the federated databases results are arguably the most noticeable and influential for the industry. The second is the market-based approaches to information exchange, which include a variety of results ranging from auction-oriented algorithms to agent-based market models. Among them, we focus mostly on the concepts and methods that have impacted our work and that have a direct bearing to the TSCM results from a comparison perspective. In the review of the previous results, we implicitly keep this context in mind: the integration environment that the field faces today is increasing on a global scale, perhaps numbering in the hundreds of millions of data sources. Furthermore, these global resources are owned and managed by disparate groups or individuals, with unique policies, schedules, and agendas. It is in this context that we emphasize independent databases as the target of integration.

The effort to integrate these resources manifest itself in a number of fields from a number of perspectives, which include but are not limited to grid computing and agent-oriented computing systems as well as distributed database systems. Each approach shares similar concerns: (1) how to dynamically scale the integration architecture, (2) how to dynamically include new resources, and (3) how to accommodate heterogeneous resources, both in content and physical capabilities. Although we review only the very limited subset that concerns the TSCM directly, it should still be pointed out that the larger trend in the larger literature certainly helps to solidify our concept and design.

The market-based results, including multi-agents, are reviewed first. The ensuing review on Global Query results also includes some popular Internet-motivated technologies such as Peer-to-Peer systems and Web Services.

2. MARKET-BASED MECHANISMS FOR RESOURCE MANAGEMENT AND ALLOCATION: MATCHING, AUCTION, AND AGENTS

Market-based systems, or systems that simulate a market economy, have emerged as compelling mechanisms for resource allocation. One advantage with this approach is the ability to deal with the integration complexity inherent to heterogeneous, distributed and autonomous resources. In these simulated economies, buyers represent resource consumers (e.g. applications, users) and sellers represent resource providers (e.g. database, CPU). Buyers and sellers trade resources, exchanging goods and/or services for profit. An underlying economic model, such as an auction or fixed price model facilitates the interaction between buyers and sellers. The applications of economic models for resource management are now widespread, including resource allocation and management in computing systems, manufacturing systems, communication networks, Grid computing, multi-agent systems, and distributed database management systems. Clearwater (Clearwater 1996) provides a survey of a diverse set of applications that offer market-based control of distributed resources.

In (Kwiat 2002), an illustration of the similarities between information grids and electric power grids suggests that they both offer dependable service requirements, infrastructure for large-scale pooling of resources, consistency of service, and pervasiveness. However, management issues arise due to the complexity of the resource allocation problem. It is

suggested that this can be solved by creating a market and allowing prices to allocate the resources. Whereas the application of the market to the electric power grid failed (e.g. the notorious California energy crisis (Kuttner 2002; Bushnell 2004)), this was primarily due to the fact that supply was significantly less than demand, and increasing resources required expensive (time and cost) new infrastructure. On the other hand, adding more resources to the information grid is, comparatively, significantly less expensive, and so the advantage of pursuing the market model for the information grid is "appealing" (Kwiat 2002). Doing so provides for arbitrary scale, heterogeneity of resources, decentralized asynchronous operation, and tolerance of localized failures (Kwiat 2002).

A brief survey of various economic models used to manage distributed resources is provided in (Buyya, Abramson et al. 2002). Identified are: (1) the Commodity Market model, (2) Posted Price model, (3) Bargaining model, (4) Contract-Net model, and (5) Auction model among others. In the Commodity Market, consumers are charged for the amount of resources consumed. Posted price is similar to Commodity Market but services are priced to increase resource usage and influence greater consumer interest. In the Bargaining model, consumers bargain with providers on pricing and usage of the services. In Contract-Net, the consumer announces a request in the form of bid contract to which providers compete, while in an Auction, a single provider invites bids to which consumers offer bid responses. The Grid marketplace is unique in its capacity to offer these economic models across various resource management systems, which includes database systems and agent-based systems. This is demonstrated in the Nimrod-G system, a Grid resource broker that supports the commodity market, and contract-net economic models. The Nimrod-G system has the responsibility for resource discovery, resource trading, scheduling, job execution and results aggregation, and works in concert with Grid middleware to provide uniform access to Grid resources and services.

A Market-based architecture to alleviate fraud and counter-speculation that may arise in agent-to-agent negotiation is described in (Collins, Youngdahl et al. 1998; Collins, Bilot et al. 2001). The architecture combines a market, an exchange and a market session, and a series of services that are utilized across the market infrastructure.

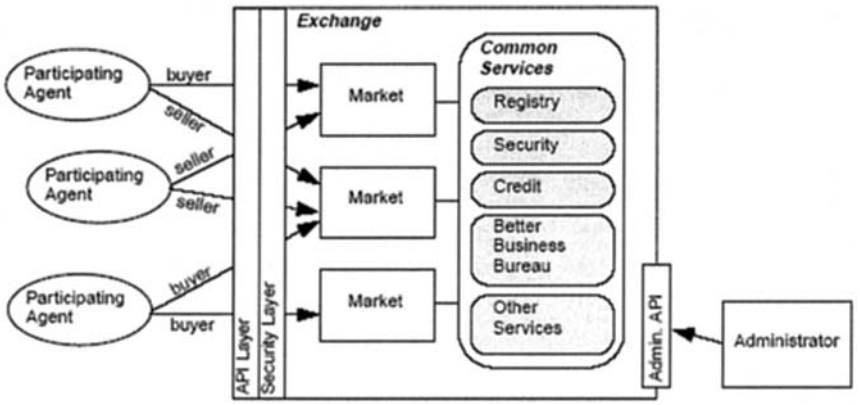

Figure 2-1. A Market Architecture for Multi-agent Contracting (Collins, Youngdahl et al. 1998)

The exchange (See Fig. 2-1) is a collection of domain specific markets in which goods and services are traded. The market facilitates trade in a specific domain, while the market-session maintains the state of agent interaction. This intermediary (the market-session) in the agent interaction provides the aforementioned controls against counter-speculation and fraud. Agents initiate bids which are submitted to the market-session. The market-session registers and timestamps the bid, and queries the registry of agents providing services. Interested agents submit responses back to the market-session which redirects the responses to the initiating client. A bid acceptance is issued to the winning bidder. The market-session enforces the rules of the market, for example, whether trade is by auction; it provides the registry of agents providing services such that no exhaustive search of the market needs to be undertaken; and, a common schema for services description. Since the market-session registers all messages in the agent interaction, it retains the state of the interaction even over periods of time. It removes the opportunity for agents to misrepresent bids, rules and timestamps essentially removing the chance for fraud and counter-speculation.

ObjectGlobe (Braumandl, Keidl et al. 2001) provides an open marketplace where queries are distributed and processed by unrelated Internet applications, although no particular economic model drives this interaction. These Internet applications are manifested as data, function and cycle providers, which can be hosted at a single site, and which offer or sell services to facilitate distributed query processing. ObjectGlobe provides a distributed, open and secure environment for query processing. A query is

processed by identifying relevant providers using the ObjectGlobe lookup service, optimizing this plan according to the capabilities of the providers and user requirements, distributing the plans to the relevant providers, which will then execute the query. Security and privacy in the infrastructure are enforced by Java and popular encryption technologies, in addition to enforcing user and application policies across the distributed resources.

Computational economies have long been apart of the artificial intelligence (AI) domain, particularly multi-agent systems (MAS). The interaction of software agents in various MAS's is guided by the electronic models to facilitate interaction (Maes, Guttman et al. 1999), although other Multi-Agent Systems (MAS) (Sycara, Paolucci et al. 2003) also employ an agent communication language (ACL) that aids communication and interoperation between agents. Software agents are intelligent, autonomous and persistent and perform tasks on behalf of their owners; decision and negotiation strategies may differ from agent to agent, but the context of the interaction (or ontology) must be shared. For example, Kasbah (Maes, Guttman et al. 1999) is a multi-agent transaction system where buyer and seller agents negotiate, on behalf of their users, in a centralized marketplace. Buyer agents bid to seller agents with no restrictions on time or price, although a utility function is employed to manipulate bid amounts over time. Likewise, seller agents also benefit from utility functions in their transactions. See (Maes, Guttman et al. 1999) for a survey of agent systems. Intelligent agents also have the capability to locate themselves to more profitable areas of the market (Want, Fiddian et al. 2001). Doing so affords the agent the opportunity to increase its value in the market, while not doing so may force the removal of the agent from the market.

Manufacturing enterprises also benefit from the use of market-based control or economic models for resource management. A modified Contract-Net protocol is employed in (Heragu, Graves et al. 2002) as the negotiation protocol for real-time task/job allocation. Intelligent agents representing manufacturing systems and manufacturing components bid and negotiate for jobs. The price set for jobs depend on multiple factors including required processing time (e.g., processing time required for a part), the utilization of resources (e.g., a material handling device), whether or not the resource is already committed, as well as system-wide factors.

Business-to-Business (B2B) commerce has been largely aided by large private trading exchanges, e.g, CommerceOne (CommerceOne 2006). These are companies that provide a framework to facilitate interoperability between businesses. The framework may contain a catalog of services offered by participating companies, a unified view of products that can be traded, as well as automated trading mechanisms (Sairamesh, Mohan et al. 2002). The exchanges are largely aided by standards (Sundaram and Shim

2001; Tsalgatidou and Pilioura 2002) that define a common framework to which all participating members must subscribe, in order to facilitate trade.

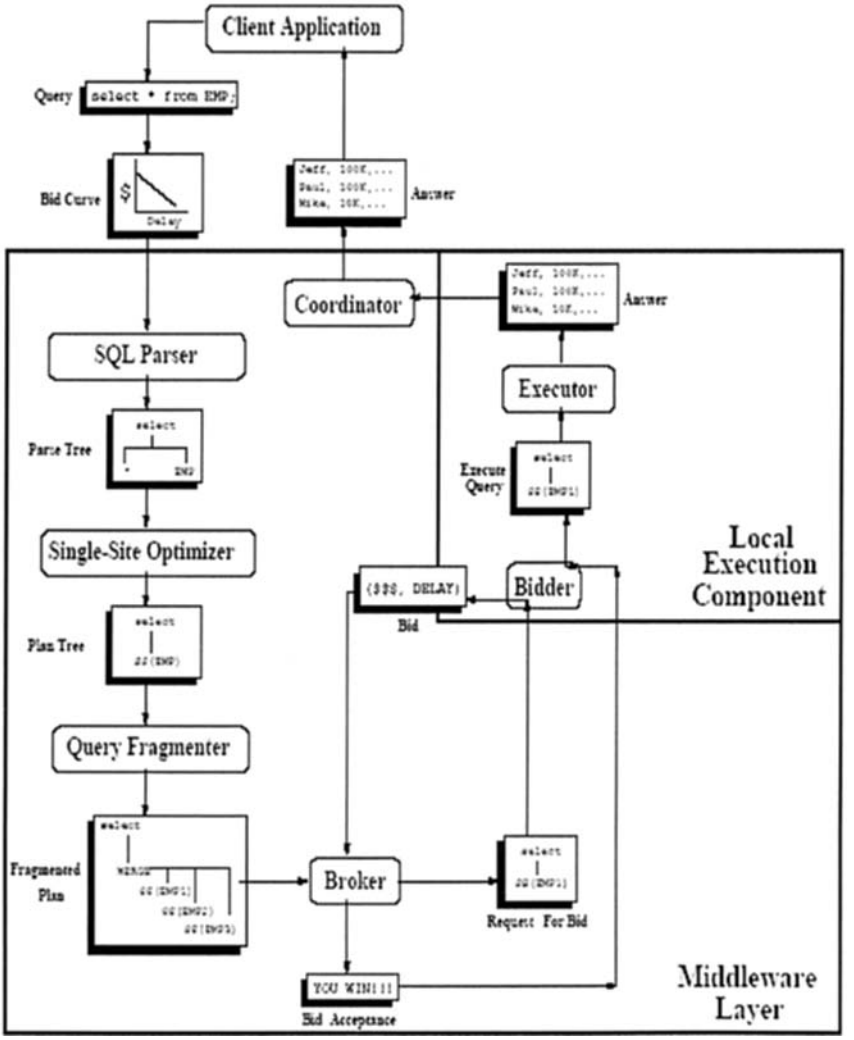

Figure 2-2. The Mariposa Architecture (Stonebraker, Aoki et al. 1996)

Mariposa (Stonebraker, Aoki et al. 1996) is a market-based wide-area distributed database management system (See Fig. 2-2). A primary problem of the distributed database management approach has been the complexity of integrating databases distributed over wide-area networks.

The market-based approach reduces the complexity to a function of price and time. Cooperating databases bid in this framework to process queries initiated by a client application. Each database possesses a client application that enables the construction of queries, a middleware layer that performs query preparation and brokering capabilities, and a local execution component that responds to and executes bids and queries respectively. At the core of the framework is the concept of budgets. When a query is submitted by the client application, a budget represented as a non-increasing function of time, is allocated to the query that represents the value of the query to the client, that is, the amount that will be paid for the query to be answered in a specified amount of time. The query and the budget are both submitted to the Mariposa middleware layer, where it is processed. The resulting query plans (which may be decomposed into multiple queries plans) are passed on to the broker, which sends out bids to other Mariposa sites (the bids consist of the query along with the budget). The bidding process is facilitated by an advertising system consisting of name servers that store advertisements from cooperating databases. These databases post advertisements describing the services offered and brokers read the advertisements to locate databases willing to execute the bid.

Mariposa utilizes two economic models as the underlying bid protocol, (1) expensive bid, and (2) purchase order. In the expensive bid protocol, the broker first submits the bid request to other Mariposa sites. Interested bidders respond to the broker with a bid that defines the cost for processing, the expiration date of the bid and the delay to start processing the bid. The broker assembles all bid responses, chooses the winning bid and notifies the winning bidder of acceptance. It may or may not inform the losing bidders. The purchase order protocol is "cheaper" than the expensive bid due to the lower number of messages required in the bidding process. Here the broker submits queries to other Mariposa sites without an expectation that the bid will be processed, and without knowledge of the costs and delay of the service. Capable and interested bidders process the query and return the results, along with a bill for services.

The term *matchmaking* is used within the multi-agent systems domain to define the entire agent interaction process, from the match on search terms, to negotiation and then agreement.

The matchmaking process in (Sim and Chan 2000; Sim and Wong 2001) involves the comparison of requests from buyers with advertisements from sellers that are stored in a Blackboard database. A broker agent is responsible for identifying matches between requests and advertisements, which are represented by multi-attribute sets. The matching algorithm is enabled by a series of conditional loops that compare the attributes of the requests and advertisements. (Sycara, Lu et al. 1999) on the other hand

utilizes an agent capability description language called LARKS (Language for Advertisement and Request for Knowledge-Sharing) to describe the requests and advertisements of agents. LARKS supports multiple stages of matching (or filtering, as described in the (Sycara, Lu et al. 1999)) that span context matching, similarity matching and constraint matching among others. The matchmaking process qualifies the type of match; it is an exact match, plug-in match or relaxed match, where each type of match is derived from various combinations of the aforementioned filters.

In (Rahwan, Kowalczyk et al. 2002; Kurbel and Loutchko 2003) the authors delineate between concerns that arise in multi-player negotiations, such as, one-to-one, one-to-many, and many-to-many agent interactions. One-to-many and many-to-many interactions are realized through the use of a coordinating agent that manages (coordinates) the individual one-to-one agent negotiations (Rahwan, Kowalczyk et al. 2002). Many-to-many negotiations are achieved by the negotiation of multiple one-to-many interactions (Kurbel and Loutchko 2003).

In (Di Noia, Di Sciascio et al. 2000) the authors deviate from negotiation and focus on the search process and the evaluation (ranking) of matches. They offer two interesting properties of the matchmaking process; first, the absence of information in a demand or supply should imply opportunity for refinement rather than rejection. Second, depending on the perspective taken in the matchmaking process, different evaluations may arise. If a supply appears to be a subset of a demand, then it would rank highly as a match, whereas the converse may not be true.

3. GLOBAL QUERY SYSTEMS

Traditional Global Query methods require varying degrees of control over participating databases, for example, a specific query language must be shared, or a common data model is necessary to integrate large numbers of databases. The following literature review explores data integration in three particular areas, Global Query Systems which is further classified as Federated Database Systems and other Multidatabase approaches, Global Schema Integration, and Multidatabase Languages. The review culminates in a comparative analysis of the related literature with the Two-Stage Collaboration Model (TSCM).

Global Schema Integration methods (Batini, Lenzerini et al. 1986; Beynon-Davies, Bonde et al. 1997; Rahm and Bernstein 2001) consolidate the schemas of multiple distributed databases into a single global schema, which avails the enterprise user with a unified view of enterprise data,

providing system transparency (the user need not be knowledgeable about system configuration), in addition to resolving semantic conflicts that may exist among the multiple systems. Federated Database Systems (FDSs) (Sheth and Larson 1990) provide greater autonomy for local systems, although a global schema may still be employed for data integration. Whereas a single global schema is required for data integration in the aforementioned global schema approach, multiple schemas are allowed in the federated approach. These multiple schemas vary by control and complexity, for example, a global federation administrator can define a global schema, through which all federated databases interact (single controller, high complexity), or local administrators can define their own integrated schemas (multiple controllers, low complexity). Multidatabase Languages (Litwin 1985) are applied to pre-existing heterogeneous database environments that lack a global controller or integrated schema. No integration measures are taken to consolidate the distributed databases; rather, the multidatabase language incorporates the necessary constructs to query the participating databases. In Multidatabase Languages, knowledge of the overall database environment is necessary for operation, such that, users must know where specific data reside in order to perform, joins or scans on database relations, and so on.

These three aforementioned methods differ in the autonomy and heterogeneity of the participant distributed databases, which subsequently affect the scalability of the integration environment. Sheth and Larson (Sheth and Larson 1990) classify databases with respect to autonomy, which includes: (1) Design, (2) Communication, (3) Execution, and (4) Association autonomy. Heterogeneity in distributed databases systems may arise as a result of either differences in hardware, software or communication capabilities; or differences in data semantics. (See (Sheth and Larson 1990) for further details on this subject). For the purposes of this research, scalability pertains to the ability of the data integration solution to add increasingly large numbers of databases without compromising functionality and performance, but rather embracing full advantage of the available resources. Kossmann (Kossmann 2000) provides a survey of the recent developments in query processing architectures.

Garlic (Carey, Haas et al. 1995; Haas, Miller et al. 1999) provides the integration and management of heterogeneous multimedia information repositories, using an object-oriented modeling paradigm. Multimedia data include text, images, CAD drawings, and medical objects. The Garlic architecture consists of data repositories which are independent of the centralized controller, and are integrated into the Garlic framework via wrappers that perform query and data transformations (See Fig. 2-3). Each wrapper translates information about the schemas and queries between

Garlic internal protocols and the repositories native protocols. Query processing is provided by the Query Services and Runtime System components which avails applications and end users with a unified schema of the Garlic database through which queries, updates and method invocation requests are issued. Queries are expressed in an object-oriented extension to the SQL query language. The Garlic query browser provides the end user with a graphical interface that supports interactive browsing, navigation and querying of the Garlic databases.

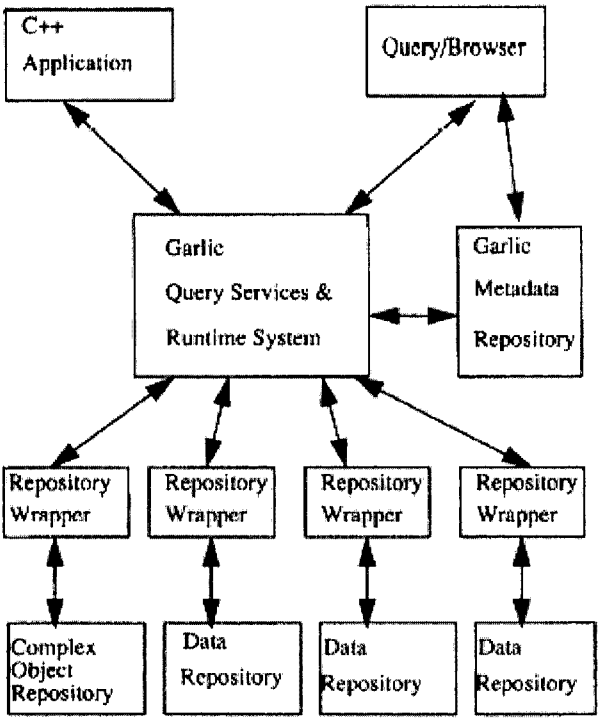

Figure 2-3. Garlic Architecture (Carey, Haas et al. 1995)

The IBM DB2 (Haas, Lin et al. 2002) architecture is rooted in Garlic mentioned above, and integrates federated data sources with user defined functions and wrappers. The simplest approaches to data integration in DB2 is the scalar User-Defined Function (UDF) that returns a scalar result, and the table UDF, which returns a table as output. The third and most powerful is the wrapper, which allows the complete integration of a federated data source. The wrapper is the mediator between the data source and DB2, and

maps the source data model to the DB2 data model while also transforming operations on DB2 to operations at the source. DB2 facilitates system transparency, heterogeneity, extensibility and autonomy. The underlying "idiosyncrasies and implementations" are hidden from the user, which arise due to the variety in the data source, i.e. hardware and software, and so on. New data sources can be dynamically added, and the functionality of the data source is not compromised by its addition to the federation.

InfoSleuth (Bayardo, Bohrer et al. 1997) resembles a market-based system with its use of cooperating agents within an open and dynamic architecture, but the absence of a computational economy disqualifies it as such. The heterogeneity of data sources on the World Wide Web and the inability to access information based on semantic "concepts" in this environment are the primary concerns of the InfoSleuth project. Accordingly, agent technologies and domain ontologies are employed to facilitate information brokering in a dynamic and open environment. The InfoSleuth architecture consists of cooperating agents that represent information resources, from users to databases, communicating via Knowledge Query and Manipulation Language (KQML) (Finin, Fritzson et al. 1994), which encapsulates queries and requests represented in SQL and Knowledge Interchange Format (KIF), respectively. User agents represent users, which interact with the network of agents via a Java applet. The user agent facilitates the formulation of queries using domain ontologies, and presents the user with the results. Other agents in the network, including ontology, broker, resource, task execution, and others, all interact to support the interoperation of distributed data and services. In particular, the ontology agent serves as the overall knowledge base of ontologies, providing all agents in the architecture with an agreed upon terminology of agent contexts as well as the ontology for the information handled by agents.

The Observer system (Mena, Illarramendi et al. 2000) is concerned with the loss of semantic information when a query is translated from one domain to another. Accordingly, Observer's *vocabulary sharing* translates queries into a target ontologies given pre-defined mappings defined in an inter-ontology relationship manager. Observer accounts for inexact matches in the translation of queries from one domain to another; regarded as partial translation, by measuring the loss of information given alternative translations and chooses the one with the least loss of information.

The Carnot project (Collet, Huhns et al. 1991; Singh, Cannata et al. 1997) utilizes the Cyc knowledge base as the basis of a global schema (Lenat 1995) to facilitate resource integration. Resource integration is achieved by translating individual resource schemas to the global schema via *articulation axioms* that describe the equivalence between components of different domains. Consequently, queries issued at an individual resource

are first translated into the global context language (GCL), both semantically and syntactically, and then to the local database manipulation languages. These queries can also be issued against the global view which is then distributed to the individual resources, although this requires knowledge of the GCL. The Carnot approach avoids the traditional global schema management problem by merging individual schemas with the global schema, as opposed to with each other. This not only retains the integrity of individual and global schemas, but provides for simpler construction and management of the global schema.

Pegasus (Ahmed, DeSmedt et al. 1991; Ahmed, Albert et al. 1993) is a heterogeneous, multidatabase management system, based on the object-oriented data modeling paradigm, that provides native access to heterogeneous and autonomous databases, and database management systems. The data abstraction and encapsulation facilities of the object-oriented paradigm, creates an extensible framework for dealing with the heterogeneities common in traditional database systems.

Query processing is made more efficient by deploying necessary application functionality (for example, query operators) to remote sites, as opposed to consolidating and processing data at a global site. MOCHA (Rodríguez-Martínez and Roussopoulos 2000) is database middleware, developed in JAVA, that provides such functionality. In traditional systems, tremendous effort would be undertaken to deploy the operators throughout the distributed computer network, or to interconnect multiple data sites, due to heterogeneities that may exist in the hardware and software, as well as the overhead realized in data shipping and query shipping. MOCHA deploys JAVA code dynamically to remote sites, dubbed *code shipping*, resulting in improved and efficient query optimization and subsequently reduced query execution times. The Query Processing Coordinator (QPC) provides the query processing functionality and deploys all necessary application functionality to clients and remote sites. The Data Access Provider interfaces with the data sources, providing an execution engine that processes the specific application functionality, and so differs from wrappers found in traditional systems.

The MDV system (Keidl, Kreutz et al. 2002) is a distributed metadatabase management systems that speeds up access to distributed data sources by replicating and caching metadata about participating resources and services in the middle-tier of its three-tier architecture. The architecture is comprised of Metadata Providers (MDP), Local Metadata Repositories (LMR) and MDV clients. MDP's synchronize metadata amongst themselves to provide uniform access to metadata by LMR's; while LMR's cache and replicate metadata relevant to local users and applications (MDV clients), using a publish and subscribe algorithm.

4. EMERGING INTEGRATION TECHNOLOGIES: WEB SERVICES, P2P AND THE SEMANTIC WEB

Peer-to-Peer (P2P) networks and Web Services are emerging as *de facto* standards for data integration and information sharing. Data providers and consumers participate in *ad-hoc* data sharing arrangements on their own terms, in real-time, and on-demand. These technologies are inherently scalable and heterogeneous; however the data sources are partially autonomous as the data providers typically must subscribe to some global information sharing standard or proprietary data format of a facilitating application. It is also important to note that these technologies are applications that sit a layer above data sources, not core technologies such as databases and query languages, such that the data sources or database facilitate data sharing but are typically passive functions of the application.

4.1 Web Services

Business Process Management (Dayal, Hsu et al. 2001) provides for the automation and integration of business processes, and is presently manifested as a system of Web Services, that foster a services-oriented paradigm. As described in (Fremantle, Weerawarana et al. 2002; Tsalgatidou and Pilioura 2002), the underlying Web Services technology include SOAP, UDDI, WSDL and WSIL. SOAP, Simple Object Access Protocol, provides messaging capabilities; while UDDI, Universal Description, Discovery and Integration protocol, provides directory or lookup services, which categorize businesses according to industry and so on. WSIL, Web Services Inspection Language, provides the method to determine what services are located at a particular site; and WSDL, Web Services Description Language, offers the ability to describe a Web Service. The attractive feature of the BPM approach is software and applications can be componentized and deployed as Web Services, without disruption of their original functionality. Furthermore, any application or data source can be deployed as a Web Service as long as they can be described using the open and standards based WSDL and its associated technologies.

4.2 Peer-to-Peer Networks

In P2P networks, individual nodes connecting to the Internet can access real-time index of files shared by other active nodes (Parameswaran, Susarla et al. 2001). P2P networks provide various advantages, most importantly, improved search capabilities relative to web-based search engines. Here, data shared is current, since the node refreshes its content

whenever connected to the network. Load balancing, redundancy and fault tolerance, though typically found in more advanced P2P implementations, are additional benefits of the P2P architecture, such that content is distributed throughout the network, and most likely will not be lost if parts of the network were to fail. However, the downsides of P2P include *noise* in resulting query results since there is no standard to describe shared resources, as well as the semantic heterogeneities that will arise due to individual naming conventions and content representation.

JXTA on the other hand, is a suite of protocols that facilitate P2P communication (Waterhouse, Doolin et al. 2002). The protocols are XML-derived, which provides platform independence and network transparency. JXTA peers can exist as providers and consumers as well as hubs that redirect query requests to other peers.

Freenet (Clarke, Miller et al. 2002) is a self-organizing and decentralized P2P global information storage system, which promotes the autonomy of system participants. It provides stability and fault tolerance by automatically replicating and relocating files according to user demand.

4.3 The Semantic Web

As the World Wide Web continues to evolve, it is necessary for information providers to describe their content with terms that are universally shared. Hendler (Hendler 2001) posits that ontologies fill this need by providing a set of terms, including a vocabulary and simple rules of inference and logic, for some particular topic such as shopping for pets. In these situations, information providers define and markup content in terms derived from a central ontology, such as the DARPA Agent Markup Language (DAML) (McIlraith, Son et al. 2001). The nature of ontologies however, allows them to be extended such that information providers can create a derived ontology which can in turn be used by other providers. The challenge therefore to achieve widespread use of ontologies, is to develop tools to simplify these procedures for the average user, and make it trivial to create semantically defined content.

4.4 XML

XML (W3C 2004) provides the foundation for a number of the nascent technologies used for data integration and sharing. The ubiquity of XML stems from its acceptance as an open standard, and the simplicity with which XML content can be created and exchanged between heterogeneous systems. XQuery (Chamberlin 2002) provides the opportunity to query an XML document, akin to SQL and relational databases. Associated

technologies, XML Path Language (XPath) (W3C 2004), Extensible Stylesheet Language Transformations (XSLT) (W3C 2004) facilitate the selection of elements in an XML document as well as the transformation of XML documents from one format to another, respectively.

5. METADATABASE AND ROPE

The Metadatabase project at Rensselaer Polytechnic Institute explores information integration in the enterprise. The results, after a decade of research, include the Metadatabase – an information resource management system for distributed, autonomous and heterogeneous environments; and ROPE – a programming environment that extends the interoperability and adaptiveness of the Metadatabase, through the use of extensible software shells.

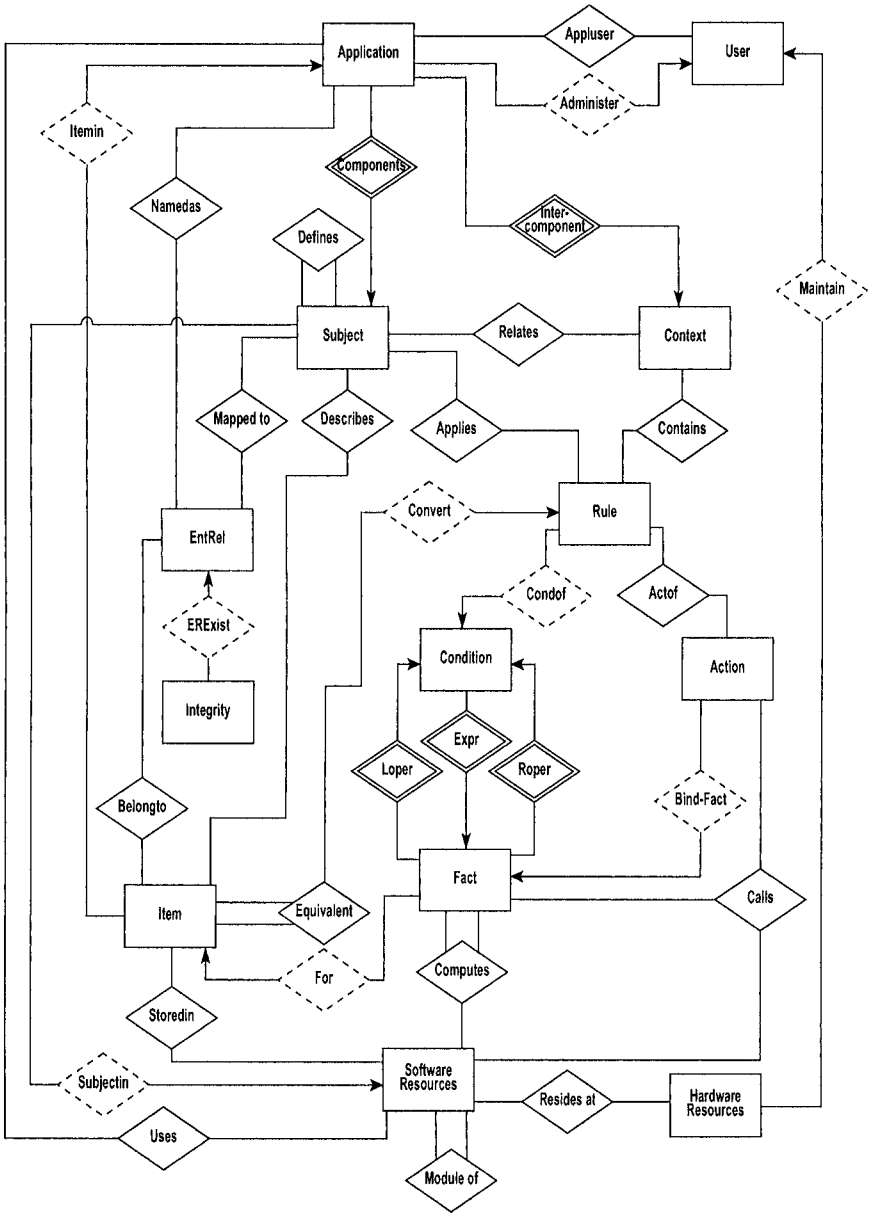

Figure 2-4. Global Information Resources Dictionary (GIRD)

The role of traditional distributed systems has been to integrate distributed data sources, without regard for the context in which the data is used. Conversely, the Metadatabase approaches this integration problem

from a holistic perspective, that is, how the applications / systems / databases interact and contribute to intra-enterprise synergies. This is regarded as enterprise intelligence, or enterprise knowledge, expressed in the form of business rules which include triggers, integrity constraints, and decision knowledge that describe information workflows between applications / systems / databases; as well as, control knowledge that delineate global equivalence knowledge and data transfer rules between the information resources. This enterprise knowledge is metadata, and is regarded as the basis of database integration in the Metadatabase architecture.

5.1 Two Stage Entity Relationship Method (TSER)

The Metadatabase architecture is comprised of three elements, a conceptual model of the enterprise, which includes all knowledge and information resources; a physical representation method that any capable relational database management system (RDBMS) can provide; and the Metadatabase management system, a management framework that provides query and metadata management and modeling facilities. The first element, the conceptual model, is manifested in the Global Information Resources Dictionary (GIRD) (See Fig. 2-4), a unified representational model of enterprise metadata. The GIRD model is created using the Two-Stage Entity-Relationship (TSER) approach (Hsu, Bouziane et al. 1991; Hsu, Tao et al. 1993; Hsu 1996), a modeling methodology that provides a representation method for the contextual knowledge of the enterprise as well as its data objects. TSER encompasses a multi-stage modeling methodology that begins with the system analysis and representation of the application / information system / user level (or functional layer). Two constructs are used to model this functional layer, SUBJECTS which describe the data objects and CONTEXT which is used to describe the intended context of the data objects. The functional layer is recursively decomposed, to produce additional functional views representing components of the enterprise information system. Dependency theory-based algorithms are then applied to the completed functional model to map the functional layer to a normalized structural model – described using four general constructs, ENTITY (OE) and three classes of integrity RELATIONSHIPs: functional (FR), plural (PR) and mandatory (MR) – which guarantees the model to be at least in third normal form. A subsequent phase of the TSER methodology generates an import schema that is amenable to input into an RDBMS (See (Hsu, Bouziane et al. 1991; Hsu, Tao et al. 1993; Hsu 1996) for a complete description of the TSER approach).

5.2 Metadatabase Management System (MDBMS)

The Metadatabase Management System (Bouziane 1991) provides basic metadata management capabilities, that is, insert, delete, update and retrieve similar to traditional relational database manipulation language constructs, however, these actions are performed on metadata. An additional management tool provided by the MDBMS is the Model-Assisted Global Query System (MGQS) (Cheung and Hsu 1996) that provides syntax-free online assistance for query formulation and processing, supports local autonomy, local system transparency and local systems interoperation. Users interact with MGQS through a graphical user interface (GUI) that accommodates model traversal and subsequent query formulation. The user selects metadata items relevant to his/her interest or perspective, which may correspond with application, functional, structural views, or actual metadata items, and MGQS produces a completely formulated and optimized global query (additional metadata items, if necessary, are included by the system – hence online assistance). Global query formulation capabilities are provided by the non-procedural Metadatabase Query Language (MQL) (Cheung and Hsu 1996), a global query language that supports queries across distributed and heterogeneous local systems with different schemata as well as different data semantics.

As is, the Metadatabase is a stand-alone, semi-active knowledge-base. The global system administrator creates a global data model using the individual schemata of the distributed applications using the TSER methodologies mentioned earlier. Global queries can then be executed against the Metadatabase via MQGS or MQL (Cheung and Hsu 1996) from the global or local perspective. Local users within the enterprise interact directly with the Metadatabase, or through an interface deployed at the local application site. It is important to note that users of the Metadatabase need not be knowledgeable about the underlying distributed architecture; the Metadatabase avails the enterprise user of full system transparency. Global queries are decomposed and transformed into the local query format by the global query processor and translator of the MGQS. Local queries are sent to the relevant local applications via the communication network and executed at the local application. Local results are returned to the MGQS, merged into a global result by the result integrator of the MGQS, and then presented to the enterprise user (Hsu, Babin et al. 1992).

5.3 ROPE

ROPE (Babin 1993) transforms the Metadatabase into an active and adaptive integration architecture. Here, the operating and decision rules are

instead located at the local sites, and reside there, as opposed to being completely referenced in the Metadatabase architecture. This adds the functionality for local sites to actively respond to rule-based changes in the local architecture, without consideration of the Metadatabase (hence, an increase in local autonomy). Thus, if changes to the local application takes place, through temporal or event-based triggers, and these changes affect the global architecture (the Metadatabase and other local sites) then ROPE provides the functionality to distribute these changes. This is facilitated by software shells that surround, and enhance the functionality of the local applications (Hsu and Babin 1993). The software shells are identical for all local applications; however, the knowledge possessed by the shells (represented as operating rules) are tailored to each application. ROPE is a programming environment that defines (1) how the software are created, (2) how the shells behave, and (3) how the shells are managed (Hsu and Babin 1993). It also integrates with the Metadatabase to push rule-based changes from the Metadatabase, downstream to the local applications, and for the local applications to push new knowledge upstream to the Metadatabase, or across to other local applications.

6. A COMPARATIVE ANALYSIS OF THE TWO-STAGE COLLABORATION MODEL WITH THE RELATED LITERATURE

A critical and required feature required of collaboration on the Internet is the resolution of semantic and structural heterogeneities that exist as a result of heterogeneous data models. The proposed Two-Stage Collaboration Model (TSCM) offers a solution for semantic heterogeneity in the global equivalence feature of the TSER modeling methodology (See Chapter 5). Semantically equivalent data items or objects that exist throughout the enterprise, in the disparate data models are made equivalent to each other during the TSER modeling process. An item with the same semantics, which is determined by the designer, is "mapped" to the global model, such that queries against the Metadatabase or a participating local database will retrieve the variants of the data items and present these in global query results. On the contrary, the Carnot project (Collet, Huhns et al. 1991) maps local data models to the CYC knowledge-base, but the scalability of this architecture is limited, as it is a manual effort to define the knowledge-base. Also, various other integration architectures, for example, Mariposa (Stonebraker, Aoki et al. 1996) and ObjectGlobe (Braumandl, Keidl et al. 2001) assume homogeneous semantics, where participant databases speak the same language and data items and objects share the

same meaning throughout the integration framework. Ontologies (Hendler 2001) offer improved classification of data semantics; however, the rigorous modeling and representation method of the Metadatabase supersedes any offering currently provided in this area.

Structural heterogeneities are addressed using the TSER modeling process (See Chapter 5). Relational, object-oriented, and object-relational databases represent a sample of the data models that can be modeled using TSER methodologies. Each data model is first represented using TSER functional constructs, then structural constructs, and then finally transformed into a comprehensive physical model corresponding to the relational paradigm. The process is repeated for each database to be integrated into the global data model. Conversely, for new databases to be added to Garlic (Carey, Haas et al. 1995) and DB2 (Haas, Lin et al. 2002) a wrapper must be created, and extensive work is required if the data model is entirely new. In Pegasus (Ahmed, DeSmedt et al. 1991; Ahmed, Albert et al. 1993), an import schema is generated for each external database, which span object, relational, and hierarchical data models – although the complexity of importation increases moving from object to hierarchical – and these are imported into the Pegasus schema to form a unified schema. Agent-based systems such as InfoSleuth (Bayardo, Bohrer et al. 1997) address semantic and structural heterogeneities in ontologies and agent modeling respectively. Typically, the agent architecture is homogeneous, with respect to architecture, such that integration across heterogeneous agent architectures requires manual intervention to resolve differences between agent communication languages, and so on.

Market-based systems in general, address the traditional global query problem with respect to integration scalability, given various degrees of database autonomy and heterogeneity, by regarding the member databases as buyers and sellers of information that trade resources for financial benefit. This approach offers numerous advantages: because of the market paradigm, member databases are not tied to a specific architecture and are free to join and disjoin the integration. Moreover, the integration and global query can span increasingly greater numbers of databases than that found in traditional approaches to database integration and global query. In fact, the scalability of traditional distributed database management systems are compromised by the optimization phase (Özsu and Valduriez 1991) of the query processor. Traditional approaches (Ribeiro, Ribeiro et al. 1997) generate query execution plans through algorithmic searches or heuristics, and measure these based on various costs: inter-site communications/network cost, response times, CPU and I/O costs. Consequently, as the search space grows (that is, the number of databases participating in the integration increases to a very large number), then the

evaluation becomes exponentially complex or intractable. ObjectGlobe (Braumandl, Keidl et al. 2001) employs a lookup service to identify unrelated data sources, query operators and servers on which to execute a query, but, the resources must register beforehand to participate in query operations. ObjectGlobe also requires that query operators be created using JAVA, which compromises heterogeneity within the architecture.