A NEW DESIGN FOR OPEN AND SCALABLE COLLABORATION OF INDEPENDENT DATABASES IN DIGITALLY CONNECTED ENTERPRISES

David M Levermore
The Boeing Company, Philadelphia, Pennsylvania

Gilbert Babin
Department of Information Technologies
HEC Montréal, Montreal, Québec, Canada

Cheng Hsu
Department of Decision Sciences and Engineering Systems
Rensselaer Polytechnic Institute, Troy, New York
Email: hsuc@rpi.edu

January 2007
Revised January 2008
Revised September 2008
Final Revision January 2009

Original Title: A New Matching Model for Information Sharing Among Independent Enterprise Databases

Submitted to the *Journal of the AIS*.

ABSTRACT

"Digitally connected enterprises" refers to e-business, global supply chains, and other new business designs of Knowledge Economy; all of which require open and scalable information supply chains across independent enterprises. Connecting proprietarily designed and controlled enterprise databases in these information supply chains is a critical success factor for them. Previous connection designs tend to rely on "hard-coded" regimes, which do not respond well to disruptions (including changes and failures), and do not afford these enterprises sufficient flexibility to join simultaneously in multiple supply chain regimes and share information for the benefits of all. The paper develops a new design: It ***combines matchmaking with global database query***, and thereby supports independent databases to interoperate and form ***on-demand information supply chains***. The design provides flexible (re-)configuration to ***decrease the impact of disruption***, and proactive control to ***increase collaboration and information sharing***. More broadly, the new results contribute to a new Information System design method for massively extended enterprises, and facilitate new business designs using digital connections at the level of databases.

Keywords: information supply chain, information system design**,** information matching and collaboration, distributed database, digital connections scaling

I. NEW INFORMSTION SYSTEM DESIGNS FOR NEW BUSINESS DESIGNS: A DIGITAL CONNECTION PERSPECTIVE

I.1 Design Goals: Open and Scalable Connection of Independent Databases

Examples of digitally connected enterprises encompass social networking, global supply chains, and the latest e-business designs (e.g., convergence of social network with business, globally integrated enterprises [Palmisano 2006], and new services [Cambridge Papers 2008]). The phrase, as discussed in [Hsu and Spohrer 2008], intends to project a sense of massively extended enterprise that potentially scales along both demand chains and supply chains. But more fundamentally, it captures the intellectual essence that these enterprises extend by virtue of (i.e., enabled by) digitizing and connecting their Information System (IS) elements: users and user interfaces; processes and applications; data and knowledge resources; computing and communication; and networking and infrastructure. Each particular connection (configuration) of these IS elements gives rise to particular (feasible) information supply chains in the digitally connected (extended) enterprises. The core of such information supply chains is the enterprise databases that support them. Because enterprise databases are proprietarily designed and controlled - i.e., they are independent of the supply chains, their connection inherently favors open and scalable designs that afford them maximum flexibility with minimum disruption for collaboration.

In this context, the paper develops a new design method: ***open and scalable connection of independent databases*** across (massively extended) digitally connected enterprises for collaboration. Its specific objectives include ***mitigating disruptions and facilitating information sharing*** in information supply chains and other collaboration relations. The research problem is ***how to make the connection open and scalable***. For example, the transaction phase of supply chain integration requires, ideally, the independent databases in the participating enterprises (e.g., those of retailing forecasting, retail inventory, suppliers' ordering, suppliers' production, suppliers' delivering, and other life cycle tasks) to work together as if they were pertaining to one organization using

one data management regime (e.g., ”drilling through” these databases for global scheduling). This oneness reduces the global transaction cost and cycle time of the extended enterprise of the supply chain. Clearly, the integration regime that achieves this oneness needs to be able to reconfigure its connections and respond to new demands, as the oneness is bound to evolve.

In the spirit of [Hevner, et. al. 2004], we employ the following ***supply chain scenario*** to delineate the above design goal: A manufacturer makes different products to supply multiple primes in different industries (including, e.g., Boeing, Cisco, GE, and Wal-Mart). These products share common raw materials, some common parts, and certain common fabrication facilities. All data are controlled under the same enterprise resource planning systems throughout their production cycle; but they are subject to different (simultaneous) supply chain regimes (e.g., data interchange protocols) imposed respectively by these primes. Each prime is also promoting its own goals of (on-demand) collaboration and information sharing throughout its own supply chain, such as e-engineering for design and global coordination of demand-supply schedules. Each chain is in fact recursive since the prime has its own customers (e.g., the prime defense contractors who subcontract to Boeing) who, in turn, have their customers; and the manufacturer has its own suppliers who have their suppliers, too. The situation goes on until it reaches end users and individual production factor providers at the level of persons. The manufacturer needs to reconcile these differing regimes, configure and reconfigure its enterprise databases’ roles in these collaboration relations, and minimize the impact of disruptions due to, e.g., any changes and failures in any parts of these concurrent supply chains. Furthermore, the manufacturer wishes to solicit as many new buyers and select from as many new suppliers as possible, from the global market. In all these cases, it wishes to reap the maximum benefits of shared data resources throughout the extended enterprises to coordinate its production and inventory schedules and reach maximum quality and productivity. Thus, there are numerous potential information supply chains just like there are numerous

potential supply chains. An open and scalable design for connecting the manufacturer's enterprise databases to any supply chains is required.

The ideal is not yet reality. In practice, supply chains tend to use fixed protocols (or, "workarounds") to connect independent databases. This approach is often associated with asymmetrical business relations, where the dominating primes promote asymmetrical sharing of information to their advantages, such as retrieval of on-demand information from supplier databases. While it may also allow the suppliers (e.g., Warner-Lambert) to gain access to select information at the prime (e.g., Wal-Mart's sales forecasting on Listerine), this approach typically presents major obstacles to the suppliers who are subject to multiple concurrent supply chains – e.g., the manufacturer in the above supply chain scenario.

More fundamentally, hard-coded designs by nature do not respond well to disruptions such as connection failures, nor facilitate flexibility dictated by, e.g., shifting demands, evolving requirements, and new technology. In addition, application-based proprietary protocols tend to be intrusive and costly to change. Open technologies such as XML, ebXML, and UDDI help to an extent, but their effectiveness is generally dependent on how standardized these databases are in their design and semantics, since interchanging data is not the same as understanding the data (see, e.g., [Levermore and Hsu 2006] for more analysis). Often, as shown in present B2B practices (see, e.g., Alibaba.com, Ariba.com, and PerfectCommerce.com), only basic file transfer (using, e.g., fixed format) is enabled, rather than database queries; which the new design provides.

In general, open and scalable connection of databases supports an enterprise to simultaneously participate, on demand, in many collaboration relations across many supply chains, as the manufacturer in the scenario wishes to do. The ability to offer/sell as well as request/buy random information from all participants benefits all parties involved (e.g., gaining cost benefits from flexible processes [Gebauer and Schober 2007] and accumulated data resources [Hsu and Spohrer 2009], as well as the previously discussed global coordination). Specific to the research objectives of the paper, the capability of on-demand

configuration and re-configuration of information supply chains responds immediately to the disruption problem, as well as to the need for flexibility.

I.2 New IS Design for New Business Design: Implications of the New Model

The new design method developed in the paper has broad implications for IS design in general. Specifically, the new design features a digital-connection view based on IS elements, which generalizes the traditional enterprise-bounded IS view (see, e.g., Alter 2008) to one that is concerned with massively extended enterprises along demand chains and supply chains. In this sense, we regard the new method as a new IS design for digitally connected enterprises at the level of independent databases, affording all the capabilities discussed above for global information supply chains and other business designs that have similar IS nature. With this design, users are connected with databases everywhere in a "federation" of participating enterprises, such as an industrial exchange, a social networking user community, and a globally integrated (extended) enterprise; where the users and databases can both give and take information on demand.

This capability has business significance: improving economic efficiency of information through accumulation (connection) and sharing (reuse) for new business designs. Supporting evidence includes the transforming roles of IT on, e.g., business [Dhar and Sundararajan 2007], on customer value propositions [Anderson, et.al. 2006], and on achieving perfect market [Granados, et.al., 2006]. This view also draws from the cascade of digitally connected enterprises stated in Section I.1, which feature massive connections of people, organizations, and resources by digital means. Finally, we consider the new design method a response to the call of a new service science in the field [Chesbrough and Spohrer 2006, Bitner and Brown 2006, Spohrer and Maglio 2007; Zhao et. al. 2008; and Cambridge papers 2008]: it supports value cocreation service systems with open and scalable connection of customers and providers at the level of databases. (See [Hsu 2009] for an analysis on a new service science.)

Previous results of IS design for massively extended enterprises (including Web services and other open technology) achieved processes interoperation, but

the new method promises to deepen them with the dimension of database collaboration. Take, first, the business designs that emerged in the initial waves of e-commerce in late 1990's and early 2000's: Exchange [Glushko, et.al. 1999] and ASP, or Application Service Provider [Tao 2001].

The Exchange model expands pair-wise relations of B2B procurement into open and scalable marketplaces for all buyers and sellers to meet and transact; and thereby gain possible economies of scale (by virtue of competition as well as consolidation of transaction supports). The Exchange could either be public, in the style of New York Stock Exchange, or private, led by some prime companies in a particular space such as Convisint.com for automotive. The model entails a technical "federation" design linking the global market servers and the massively distributed information systems of the participating enterprises. Each meeting of buyer and seller forms a (on-demand) B2B, and connecting inter-related B2Bs finds a (on-demand) supply chain. It promoted new IS design paradigms that employ ***matchmaking*** at a global site (to establish the requirement of connection) and ***proxy servers*** at local/enterprise sites (to execute the connection at the level of processes/applications - e.g., swapping XML objects).

However, buyers and sellers do not have the ability to query each other's databases. This ability is, at the least, helpful to establishing the requirements of a supply chain, or a collaboration for information sharing, such as matching buyers with sellers, and information requesters with information providers. More importantly, it could be mission-critical for executing the collaboration such as integrating the (real time) production schedules throughout a supply chain [Cingil and Dogac 2001, Davenport and Brooks 2004, Levermore and Hsu 2006].

The ASP model, on the other hand, turns a software vendor into an online global processor/server of the software for the clients. Therefore, its IS designs promoted shared data and transactions management, featuring client-side computing as well as strong server capabilities. These practices continued to expand and resulted in further design paradigms including the employment and deployment of open source technology, and prompted new models and business

designs such as service-oriented computing [Erl 2005] and computing/software as a service (e.g., SaaS). While the ASP practices have revealed the critical role of process interoperation, those of the SaaS and others have shown the need to make the interoperation open and scalable, for sustention of the practices.

In fact, from the perspective of global information supply chains, all these models need openness and scalability in their connection of processes (see, e.g., [UN/CEFACT 2003]). The Exchange model needs them to facilitate the transaction phase of supply chains formed at an exchange. The ASP model also needs them since the providers would want to scale their services to as many prospective clients as possible. Therefore, the new design method embraces the requirements of openness and scalability while it deepens the previous connections of processes to the level of independent databases. It also employs the global server-distributed proxy paradigm as the basic architecture for all digitally connected enterprises.

I.3 Solution Approach to Developing the New Design Method

The above analysis led to a basic solution approach: synthesizing certain open and scalable results of previous IS design (especially the Exchange model) with appropriate proven results in distributed databases; namely, integrating matchmaking into global database query. Appendix II provides a technical analysis of the research problem and substantiates this proposition. For simplicity, we refer to the new design method the ***Information Matching model***. The model, in a nutshell, extends the previous scope of distributed databases (within an enterprise or a finite extended enterprise) to independent databases (across digitally connected enterprises). The algorithms of the model are proven by theoretical analysis, and their implementation in a prototype verifies feasibility.

The basic (also the broadest) concept of Information Matching may be best illustrated by a thought model which we call “an eBay for information resources”. In this vision, a large number of information customers are matched with a large number of information providers on a concurrent and continuous (24/7) basis at an Information Exchange. Unlike eBay, however, the “information

eBay" has a number of important unique properties, stemming from the unique properties of information. They include the fact that information can be presented in many different ways from the same physical data (the notion of "views" of data); that information can be used, re-used, and shared by many without diminishing its value (i.e., not physically "consumed"); and that any participant could possess information resources that others want at any time. Therefore, a participant could post ad hoc requirements to look for suppliers in particular tasks, as a buyer, and simultaneously offer multiple views of its databases for use by prospective suppliers, as a seller. The actual sharing of information will take place as database executions after the match is made. The Information Exchange may be construed for an enterprise, a massively extended enterprise, or a community of participants of any types desired.

As an illustration, the supply chain scenario of Section I.1 will employ an Information Exchange corresponding to the manufacturer's business space, including all willing participants from the population of manufacturers, retailers, and contractors. The Information Matching model will first establish the relationships of a supply chain by posting requests (for buyers or sellers) – or, issuing "global queries" as in database terminology, to find the partners who fit. Then, it will support the transaction phase of the supply chain by its ability to execute database tasks. A virtual sequential supply chain is formed when sequentially inter-related B2B pairs (overlap at either end) are identified and connected at the Exchange. Sequential information supply chains are formed by following some ordered list of database executions for transactions (e.g., forecasting, tier 1 supplier production, tier 2 supplier inventory; and so on). Concurrent processing of all these chains is supported since all pairs are connected at the Exchange in parallel. Configuration and re-configuration are achieved by arranging for these connections through on-demand matching.

The manufacturer in the scenario can connect to different pairs pertaining to many parallel virtual supply chains. When additional managerial controls are added, such as certification of suppliers for particular prime companies, a virtual supply chain can become as binding as desired by the participants. In addition,

the design consolidates all changes at the common infrastructure for all to use, and thereby provides economies of scale.

Technically, the purest form of the Information Matching model will allow for any number of providers from any type of information repository, anywhere in the digitally connected community. A far more modest model will assume a pre-defined community regulating the participants and imposing certain (open technology) protocols to make the model practical. The practices of global supply chains [Cingil and Dogac, 2001, Wisner and Tan, 2000], for example, provide a lower bound to the vision and an upper bound to the requirements for implementing the vision. From this perspective, information customers (users) are comparable to traditional global database queries (subscribing) which will be satisfied either by using single individual information providers or by joining multiple such providers on an as-needed basis. Information provider is, on the other hand, a new type of query (publishing) representing the proactive and dynamic provision of ad hoc data resources which will be satisfied by single or multiple customers. The matching also involves satisfying rule-based negotiation and other matching conditions from each type of query. Finally, both the information customers and providers search for their counterparts on demand; the matching can prolong for a period per demand; and the matched queries are executed automatically to complete the transaction.

These required capabilities are partially found in the literature of matchmaking and distributed databases. However, previous matchmaking results are generally not compatible to global query processing as a synergistic solution for independent databases. In this research, we develop new results which integrate matchmaking into global database query and thereby enable Information Matching. Similar to previous global database query, the new matching model assumes that the global community requires a registration process and some global (open technology) protocols through which the participating databases join the community. However, unlike previous results, the new model uniquely allows for any number of databases subscribing ***and*** publishing with any degree of flexibility (contents, rules, and proprietary control),

within a community ("federation") of digitally connected enterprises. As such, a database can make requests (issue queries) against other databases, just as it can respond to others' requests in the manner of traditional global database query. The new model is integrated with the previously established Metadatabase [Hsu et. al. 1991, Babin and Hsu 1996, Cheung and Hsu 1996, and Bouziane and Hsu 1997], which executes the actual information retrieval after the match is made, through a global architecture [Hsu, et. al. 2007].

The specific ***technical contributions*** to federated databases include improvement on autonomy, heterogeneity, openness, and scalability. To be more precise, the new results provide a unified metadata representation method to define a new query language (for both publishing and subscribing) and a new Query Database, so as to simplify the processing and achieve efficient matching. A new global blackboard design implements the language and the Query Database and administers the ensuing global query processing. The representation method is further integrated with the Metadatabase to also streamline the matching with global query processing at participant databases. As such, the whole life cycle of Information Matching is simplified to achieve computational efficiency for transactions and make the model feasible. Other matching methods in the field do not unify the representation of bids with their processing; and previous global database query results do not support publishing queries and their proactive matching with subscribing queries.

The rest of the paper substantiates the above concepts and claims with technical details, focusing on the Information Matching methods. First, Section II reviews the foundations of the new design mentioned above; viz. the collaboration architecture and the Metadatabase. Then, Sections III and IV present the new methods: the matching logic and algorithms (III) and the matching language and system design (IV), respectively. Section V evaluates the new design using observations from a basic laboratory prototype and conceptual analysis. The last section, Section VI, concludes the paper with an analysis of how the new model improves global database query. More technical details are

provided in the Appendix: Glossary, technical analysis of the research problem, and performance analysis. The supply chain scenario is used throughout.

II. THE OVERALL DESIGN: ARCHITECTURE

The Information Matching model presented here assumes the collaboration architecture developed in [Hsu et. al. 2006 and Hsu, et. al. 2007] as its foundation. The new Information Matching methods add on top of this foundation to provide open and scalable connection of participating independent databases. The architecture is a new design for the general class of technology called federated databases (see Appendix II for a technical review of the problem). It employs the previously established Metadatabase model and general Exchange design to provide open and scalable operations. The overall view of the global architecture is shown in Figure 1. The entire environment is the Information Exchange, with the Global Blackboard embodying the Information Matching model. Export Databases represent enterprises databases to the community, through proxy serves which are implemented at the enterprise sites.

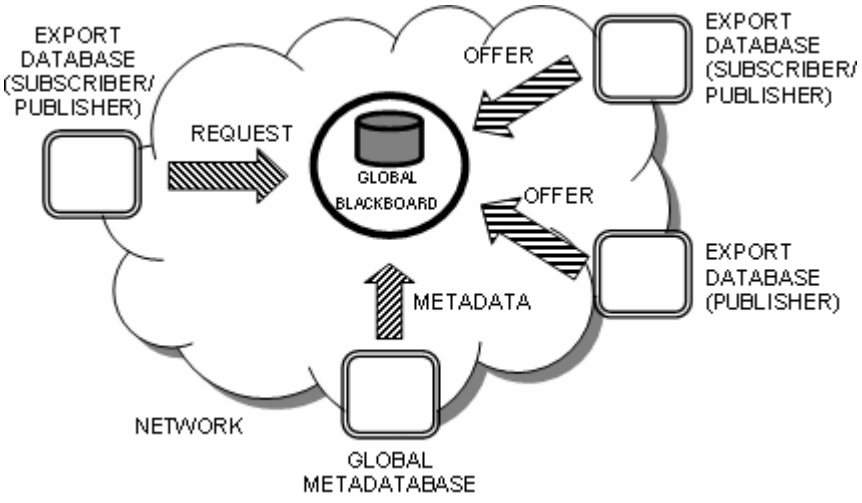

Figure 1: Conceptual Overview of the System Design

We first provide an overview of the collaboration environment using the supply chain scenario of Section I.1. For simplicity, we reduce the scope of the scenario to just a handful companies, which can be readily generalized and hence still provide sufficient representative value. Designate the manufacturer (called P1) a maker of parts A and D, a second (P2) to be a maker of parts B and E, and a third (P3) for parts C and F. We further assume that part A is made of parts E and F; B of D and F; and C of D and E. Three parallel supply chains are possible in this scenario: P1 buying from P2 and P3; P2 buying from P1 and P3; and P3 buying from P1 and P2. More hierarchical layers and possible supply chains are formed if we add a fourth participant (P4) which makes part X from A and E, and a fifth one (P5) making part Y from X and F. Company P4, a prime, may buy either exclusively from P1 (supplying both parts A and E) or separately from P1 and P2. Company P5, another prime, has similar but even more choices.

To "synchronize" semantics, companies P1, P2, and P3 may each register an export database using the Metadatabase ontology (see below), supplemented with industry standards (e.g., part codes), as the common semantics to serve the original three chains. Additional export databases may be added to serve the particular requirements of P4 and P5, if necessary. Different virtual "federations" of export databases will result. On-demand configuration and reconfiguration of information supply chains are formed every time an export databases interacts with others. As illustrated in Figure 1, the participants request (subscribe) or offer (publish) data through their export databases registered at the Global Blackboard. The Blackboard matches requests with offers, assigns and delivers the requests to the export databases for processing, and returns the results to the subscribers. The Metadatabase supports the interoperation of these export databases, and through them the underlying independent databases.

We now turn to define the key elements of the above architecture.

Global Blackboard operates the Exchange at the global site (while a peer-to-peer version will have it duplicated at local sites, as well) of the collaboration community. It implements the matching methods (e.g., the matching

language and algorithms), conducts the matching, and interoperates with the Metadatabase to execute global database queries for the matched requests.

Export Databases are locally-controlled subsets (consisting of an export schema and the export data) of the enterprise databases at the proxy servers of the local sites. An enterprise database can have any number of export databases registered, and each export database is a particular image/personality of the enterprise database(s) presented to a particular business relation for other participants to see and use (i.e., for a particular federation of independent databases), such as a supply chain for a particular prime/original equipment maker. The manufacturer in the supply chain scenario will create three export databases for Boeing, Cisco, and Wal-Mart, respectively.

Proxy Servers are the surrogates of the global blackboard implemented at the local sites. A proxy server includes export database(s), the Metadatabase local shells (for global query processing), and optional components for possible peer-to-peer exchange (e.g., distributed Metadatabase and distributed blackboard). Proxy servers connect the local sites with the global site as well as among themselves, and each participating enterprise requires only one proxy server. For example, the manufacturer in the supply chain scenario will use the same proxy server to connect to the global blackboard for all supply chains, and to support any virtual information supply chains or other collaboration relations.

Metadatabase, in a nutshell, is a relational database of metadata: data models and rules represented in a set of common constructs defined in the Two-Stage Entity-Relationship (TSER) model [Hsu et al., 1991, Hsu et al., 1993]. The TSER model is a neutral representation of the usual data and knowledge modeling constructs provided in the entity-relationship approach, object orientation, and rule-based representation (predicate logic) [Bouziane and Hsu, 1997]. In essence, the TSER constructs constitute an information modeling ontology applicable to relational databases and object-oriented languages such as Express and UML. In other words, the Metadatabase is structured to accommodate any data and knowledge models that are consistent in concept

with the ontology. Application data models (e.g., product design, process planning, and shop floor control) and rules, including the methods (logic and algorithms) involved in them, are then consolidated, stored, and processed as raw metadata entries in the Metadatabase. In this sense, the Metadatabase is a global repository of data semantics and knowledge for the digitally connected enterprises that it represents, for all parties to tap in for any business relations.

The Metadatabase model assumes a regulated community (a federation) and employs a registration process to construct the community Metadatabase. The registration process (reverse-) represents local application models into TSER metadata and populates them into the Metadatabase, where the equivalence of data items (i.e., local attributes) across applications is established and stored as a metadata relation. An application model is typically represented by a set of metadata entries. The schema of the Metadatabase, called the GIRD model or global information resources dictionary [Hsu et al., 1991], structures the repository of metadata and implements the TSER ontology. The GIRD is generic and steady to all metadata as long as the ontology stays relevant to all models. Therefore, the Metadatabase is open and scalable to new enterprise databases to the extent that the ontology fits their export databases.

The Metadatabase is also operationally open and scalable for adding, deleting, and updating enterprise (export) databases without disruption, since adding, deleting, and updating their representation are but ordinary relational operations against the Metadatabase. The proxy servers (and Metadatabase shells – see below) mitigate the disruption in physical connections. The registration process, including reverse-engineering, is amenable to automation for certain local data models that use standard relational design and object models [Shvaiko and Euzenat, 2005]. The Metadatabase can be implemented in a peer-to-peer manner, where the maximum implementation will involve distributed copies of the Metadatabase, with the minimum version calling for distribution of only the data equivalence meta-relations. In any case, maintaining the distributed metadata is a attainable requirement, since metadata do not

amount to huge volumes and their change is relatively infrequent when compared to raw data of industrial production databases.

The Metadatabase Model includes an SQL-like global query language (the Metadatabase Query Language or MQL [Cheung and Hsu, 1996]) for processing. It connects to the local systems through a network of shells called ROPE, or Rule-Oriented Processing Environment - see [Babin and Hsu, 1996]. The MQL language does not require users to possess detailed knowledge of the local databases; instead, the Metadatabase determines the local details and translates the MQL expressions to local-bound sub-queries in SQL and possibly other local data languages. The ROPE shells are Metadatabase proxies at each local site to interoperate the sub-queries with the local database, and deliver the results back to the Metadatabase for final assembly. These elements accommodate new systems and further enhance operational openness and scalability.

The new matching methods extends the MQL to include publishing queries, and augments both subscribing and publishing query syntax with operating rules. The extended MQL is reconciled with the Metadatabase schema to structure a new Query Database, serving as the repository of these two types of queries; which, in turn, seek matches against the repository. The matching algorithms are, therefore, the particular query processing logic performed on the Query Database. The matched queries, then, are executed as traditional global database queries by the Metadatabase; which also participates in both stages to reconcile semantics and identify specific database processing tasks.

We now develop the new matching methods: matchmaking, the query language exMQL), and the Query Database, in the next sections.

III. BASIC LOGIC OF THE NEW MATCHING MODEL

III.1 Overview of Matching

A **match** can come from a single publishing query of a single export database, or a join of multiple queries from multiple export databases, with a subscribing query, in a virtual federation. In the above supply chain scenario,

each match forms a possible connection in the virtual supply chain; and the joint of such connections in sequence, if they exist, forms a supply chain hierarchy. When actually committed and executed, then the possibility realizes into an instance of a particular supply chain. In other words, the matching is oriented to direct connections among participants; leaving the sequencing of these connections to the actual utilization of the transactions by the participants. However, negotiation rules and other matching conditions derived from the sequencing requirements can be established as a part of the publishing queries to impose the managerial constraints of sequencing, if necessary.

The first step of matching is to identify all the sets of publishing queries that contain all the required data items, or attributes (e.g., data that pertain to the buying of parts A and E from P4, and the selling of parts A and E from P1 and P2). These sets are qualified as **data item feasible**, as each may be used to extract all the data items required from the export databases. Second, we verify that all publishing queries in a data item feasible set (if there is more than one) can indeed be joined. This is done by verifying the existence of common data items among publishing queries within the set (e.g., establish possible connections). Such sets are said to be **join feasible**. This verification process may result in the addition of new publishing queries to the set to make the (extended) set join feasible. Third, we verify that the constraints on a join feasible set match the constraints on the subscribing query (e.g., the sequencing rules). When this is the case, the set is said to be **constraint feasible**. Finally, the best constraint feasible set is selected for allocation. Define the following notations:

Let M_k: the metadata from a single system *k*, and $\bigcup_{k=1}^{n} M_k$ the collective set of metadata from all systems available in the Metadatabase.

Let *I*: the set of all data items, where $I \subseteq \bigcup_{k=1}^{n} M_k$.

Let Q^S: a set containing search terms used in a subscribing query, where a search term is a data item $i \in I$ and $Q^S \subseteq \bigcup_{k=1}^{n} M_k$.

Let Q^P: a set containing search terms used in a publishing query, where a search term is a data item $i \in I$ and $Q^P \subseteq M_k$.

Let *R*: a set containing rules associated with a query. A rule $r \in R$ contains a set of conditions *C*, and optionally a set of actions *A*.

Let *C*: a set of conditions used to qualify the search terms in the query, or the query in general. There are three classes of conditions: *selection conditions* (C^S), *join conditions* (C^J) and *negotiation conditions* (C^N).

Let *Card (B)*: the cardinality of the set B.

Given these definitions a **subscribing query** takes the following form:

$$S = (Q^S, R \mid Q^S \subseteq \bigcup_{k=1}^{n} M_k \wedge R = (C, A)), \text{ where } C \subseteq C^S \cup C^J \cup C^N$$

As such, a subscribing query is composed of a set of data items and a set of rules, where the data items may be selected in the global information model (i.e., the Metadatabase), and the rules formulated by the users to represent selection and join on data items, ***and*** negotiation conditions and actions.

A publishing query derives its search terms from its export database (schema). Accordingly, a **publishing query** takes the following form:

$$q = (\boldsymbol{Q^P}, R \mid Q^P \subseteq \boldsymbol{M_k} \wedge R = (C, A)), \text{ where } C \subseteq C^S \cup C^J \cup C^N$$

The set of all publishing queries is denoted as *Q* (i.e., $q \in Q$).

III.2 QUERY MATCHING: IDENTIFYING COMPLEMENTARY QUERIES

The matching process (1) identifies matching data items, (2) combines queries to identify item and join feasible solutions, and (3) matches constraints, to qualify sets of publishing queries.

Step 1 – Identify Matching Data Items

This step determines the match category for each publishing query in the Blackboard. The extent to which a publishing queries $q \in Q$ may be used to fulfill a subscribing query S is categorized as *Exact* Match, *Superset* Match, *Subset* Match, and *Intersect* Match based on the level of overlap between the required data items and the data items it provides. The match between two data items occurs if both data items $i^S \in Q^S$ *and* $i^q \in Q^P$ are one and the same or if they are semantically equivalent - the Metadatabase would contain metadata specifying such an equivalence. Hence, if i^S and i^q are different but equivalent, the $q \cap S$ operator presumes they are the same and only returns i^S or i^q.

An **exact match** occurs when all data items in S are in q and vice versa:

$Card\ (S) = Card\ (q \cap S)$ and $Card\ (q \cap S) = Card\ (q)$, where $Card\ (q \cap S) > 0$

A **superset match** is when all data items of the publishing query q are present in the subscribing query S. This means that:

$Card\ (S) > Card\ (q \cap S)$ and $Card\ (q \cap S) = Card\ (q)$, where $Card\ (q \cap S) > 0$

A **subset match** is when all data items from the subscribing query S are found in a publishing query q. Hence, we have:

$Card\ (S) = Card\ (q \cap S)$ and $Card\ (q \cap S) < Card\ (q)$, where $Card\ (q \cap S) > 0$

Finally, an **intersect match** is when some item of the subscribing query S are found in the publishing query q, and vice versa. Formally:

$Card\ (S) > Card\ (q \cap S)$ and $Card\ (q \cap S) < Card\ (q)$, where $Card\ (q \cap S) > 0$

Step 2 – Combine Queries to Identify a Feasible Solution

This step determines the sets of publishing queries that are data item feasible and join feasible. Formally, a set $Q = \{q_1, q_2, \ldots, q_m\}$ of publishing queries (also called a **combination query**) is **data item feasible** with respect to subscribing query S if and only if

$$\left(\bigcup_i q_i \right) \cap S = S$$

In other words, a combination query Q is data item feasible if it provides all the data items from the subscribing query S. A combination query $Q = \{q_1, q_2, \ldots, q_m\}$ is **join feasible** with respect to subscribing query S if it is item feasible and

$$\forall q_i \in Q, \ \exists\, q_j \in Q, j \neq i \mid q_i \cap q_j \neq \varnothing$$

That is, not only does Q provide all the data items, but there exists data items in each publishing query that may be used to join the results together. Otherwise, joining different queries (Cartesian product) may not be of any value. It follows that a publishing query that is an exact or a subset match is data item feasible and join feasible, as $Q = \{q\}$. When a query is a superset match or an intersect, it must be combined with other queries in order to be join feasible.

Consider, $S = \{item_1, item_2, item_3, item_4\}$, a subscribing query, and $q_A = \{item_1, item_2, item_n, item_{n+1}\}$, $q_B = \{item_2, item_3\}$, and $q_C = \{item_4, item_m, item_{m+1}\}$, publishing queries. As can be seen in Figure 2, q_A, q_B, and q_C match S on particular data items.

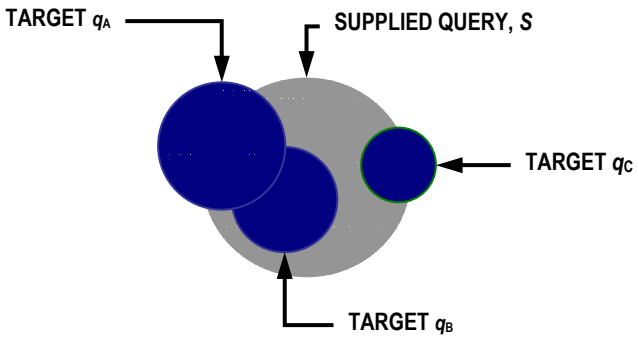

Figure 2: Conditions Required for a Combination Match.

The union of q_A, q_B, and q_C, which is denoted as Q is a combination query that contains all data items found in S; however Q is only data item feasible since q_C does not share data items with q_A and q_B. For Q to become join feasible, logical relationships among q_A, q_B, and q_C are necessary.

In this case, and by employing the Metadatabase, we can determine if a combination query can be made join feasible. Specifically, we identify logical connections between queries: In the example, the intersection of q_A and q_B is non-empty. We also find that $q_A \cup q_B$ and q_C do not share common data items. Thus, the Metadatabase may be consulted to determine how q_C may be connected to q_A or q_B for Q to be considered a join feasible solution. This requires adding new publishing queries to Q.

In order to construct join feasible solutions from a combination query, the results from Step 1 are combined and each combination is evaluated to determine if it is item feasible with respect to S. The resulting combination queries are classified as, (1) **combination exact** match, (2) **combination superset** match, (3) **combination subset** match, and (4) **combination intersect** match, analogous to those defined for Step 1.

We construct a connected graph to identify combination queries. The graph is made of nodes representing queries resulting from Step 1, where every node is connected to every other node. Every node generates a unique message comprising the query name (identifier) and the attributes of the query (body). At the start of the step, the number of cycles that this process should run is determined, which equals to the number of superset and intersect queries found, minus 1. Table I illustrates the combinations of three queries, q_A, q_B, and q_C.

Table I: Evaluation of Combined Queries.

Cycle	Initial Combinations	Final Combinations
1	q_A, q_B, q_C	q_{AB}, q_{AC}, q_{BC}
2	q_{AB}, q_{AC}, $_{qBC}$	q_{ABC}

In the first cycle of the process each node broadcasts its message, while we record all combined queries created from the broadcast, and ignore the duplicates found. A broadcast message received at a node has its message body combined with the contents of the node, and a combination query is created. Finally, the combination queries determined in each cycle of the process

constitute the nodes for the next cycle, with the same broadcast messages. The order in the combination is not significant. If a combination query shares data items with the input query *S*, then the number of shared items constitutes p_{best}, the largest number of items shared with *S*. If this number is greater than an earlier round of processing, then we test the join feasibility of the query and return new solutions p_{best} and P_{best}, the set of all query sets used to find p_{best}, if the function returns true. If a join feasible solution cannot be found then the modified *Shortest Path Algorithm* [Cheung and Hsu, 1996] determines if the entities and relationships to which the data items in the queries belong, are logically connected. In the Shortest Path Algorithm, the graph is constructed from the same set of nodes but two nodes are connected only if they share a common (or equivalent) data item. We then proceed to find a spanning tree that minimally contains the queries to combine. The Shortest Path Algorithm searches for additional metadata, which logically connects these disjoint queries, perhaps allowing for the subsequent modification of one or more of the queries, *q*. This process is repeated if p_{best} is unchanged in the current round of processing.

The step returns all the combination query(s) (P_{best}) that contain the greatest number of data items (p_{best}) common to the input query *S*.

Step 3 – Constraint Matching

A successful query match also requires a compatible match between constraints, if any constraint exists. We have a **constraint feasible** solution if the constraints in the subscribing query *S* satisfy the corresponding constraints in all the matching queries $q \in Q$. For example, if a subscribing query *S* contains a negotiation constraint, *price* < $20.00 and a matching publishing query $q \in Q$, contains a constraint, *price* = $10.00, then these constraints are compatible.

The challenge that arises with the constraint matching process is to evaluate not only the semantics of the constraints, but the quantitative aspects as well. However, there are no actual data values available for evaluation during the matching process as we want to match the constraints before actually running the queries. Therefore, the effect that the operators have on these data items

cannot be readily identified. Consequently, a new method to estimate the constraints must be devised. To this end, we measure if the negotiation attributes/data items of the constraints have the same domain; and if so, whether there is any possibility that the data values will satisfy each other, by utilizing *truth tables* in the evaluation; as follows.

Each constraint consists of a data item / negotiation attribute *i*, a comparison operator from the set {=, <, >}, and a data item / literal value *v*. Therefore, for each constraint in *S* and *Q*, we can establish all the possible constraint variations. For simplicity, we use *Q* to refer to all $q \in Q$. For example, given a constraint, *price* < 20, the variations are *price* = 20, and *price* > 20. Therefore, given *S* and *Q*, a *matrix of assertions* V is created, consisting of all the provided constraints and their variations. It asserts the value of the different data items/variables in a truth table and assesses the truth value of the constraints.

Next, we assess the compatibility of all combinations of the assertions in *V*. The number of combinations is 3^n, where *n* is the total number of constraints in *S* and *Q*. A combination corresponds to a set of sub-domains. The assertions within a combination are compatible if they may occur at the same time. A constraint is *true* if the combination is compatible and if the assertions in the combination match the original constraints (See Table II).

Table II: Compatibility and Truth Table for Constraint Matching

Constraint Combination			**Compatible**	***S***	*Q*
$x = 1$	$y = 4$	$x = 5$	No		
$x = 1$	$y = 4$	$x < 5$	Yes	T	T
$x = 1$	$y = 4$	$x > 5$	No		
…	…	…	…	…	…
***x* > 1**	$y > 1$	$x > 5$	Yes	F	F

Note that a single constraint corresponds to a sub-domain of a data item. A pair of constraints is compatible if the sub-domains they represent intersect. It

follows that constraints in different domains are necessarily compatible. Consider the combination $x = 1$, $y = 4$, and $x < 5$. This combination is compatible since (1) $x = 1$ does satisfy $x < 5$ (the sub-domains intersect), and (2) $y = 4$ and $x = 1$ are compatible (the sub-domains are independent). We must then assess that constraints in *S* and *Q* hold true for this assertion combination, since each of the assertions in the combination match the originally provided constraints.

The compilation of the results found in Table III reveals the numbers of *true*/*true* (*TT*), *true*/*false* (*TF*) and *false*/*true* (*FT*) results for the given set of constraints in *S* and *Q*. A *true*/*true* result is when constraints on both *S* and *Q* hold, and corresponds to the intersection of the set *S* and *Q*, whereas a *true*/*false* (constraint on *S* is true, constraint on *Q* is false) corresponds to the region bounded by *S* constraint, and conversely a *false*/*true* (constraint on *S* is false, constraint on *Q* is true) is bounded by *Q* constraint. A *false*/*false* result is discarded since it indicates that neither constraints match. The *TT*, *TF* and *FT* results are further classified according to the *exact*, *superset*/*subset*, and *intersect* classification described in Step 1. This is summarized in Table III; where, if the *TT* quantity is greater than zero, with the *TF* and *FT* equal to zero, then an exact match between the constraints has been identified; and so on.

Table III: Classification of Constraint Match Results

	TT	**TF**	FT
Exact	> 0	= 0	= 0
Superset	> 0	> 0	= 0
Subset	> 0	= 0	> 0
Intersect	> 0	> 0	> 0

III. 3 QUERY ALLOCATION: ASSIGN QUERIES TO WINNING EXPORT DATABASES

Once a successful match has been found, then the query *S* is allocated to the corresponding export database, or databases, of the matching query *Q*. It is trivial if *S* matches a single *Q*. However, if multiple queries $Q \in P_{best}$ are a match

for S (i.e., the $Q \in P_{best}$ are similarly item, join and constraint feasible, and they can be substituted for each other to provide a single, equivalent successful match for S), then the model allows for two basic strategies to process the "tie", depending on the requirements of the actual applications of the Information Matching. One is to present all matched sources for the participant to decide, including the choice of buying from them all (binding all matched export databases for execution of the query). Another is to provide some automated selection, either binding all sources matched by default, or identifying the optimal $Q \in P_{best}$ given some decision rules. We discuss some possible strategies next.

To reduce multiple queries to the case that S corresponds to a subscribing query and $Q \in P_{best}$ a publication query (i.e., there is a single $q \in Q$ for $Q \in P_{best}$), five decision criteria which can be specified by the user as (multiple) actions during query formulation, in any combination from "grouped together" to "only a single criterion".

- *Most Favorable Conditions* – use price, delivery date, and other major indicators designated as tie breakers.
- *First-Come-First-Serve* or *Last-Come-First-Serve* – uses the system-defined timestamp of each query to select a winner.
- *Network Performance* – base selection on the geographical location of the export databases, such as proximity rating of the computing and/or the logistics network involved.
- *Past History* – the export database that has most frequently provided answers and/or reliability in previous matching sessions will be chosen.
- *Preferred Organizations* – the user may specify preference for export databases (including the owners/participants) during query formulation.

Once a selection is decided upon, matching may be handled in a straightforward manner - that is, allowing the publisher to service all subscribers matched. This approach follows from previous results with MQL [Cheung and Hsu, 1996]. Information, unlike physical goods, can be shared infinitively.

IV. THE MATCHING LANGUAGE AND QUERY DATABASE

The matching logic and algorithms of the above section require a new language, exMQL, to represent and process all the subscription and publication queries, and a new Query Database to store them as relations. We provide these results in this section. They can be proven through successful execution in a prototype. The designs as presented here are self-evident for verification.

IV.1 Query Database Schema

The schema of the Query Database is shown in Figure 3.

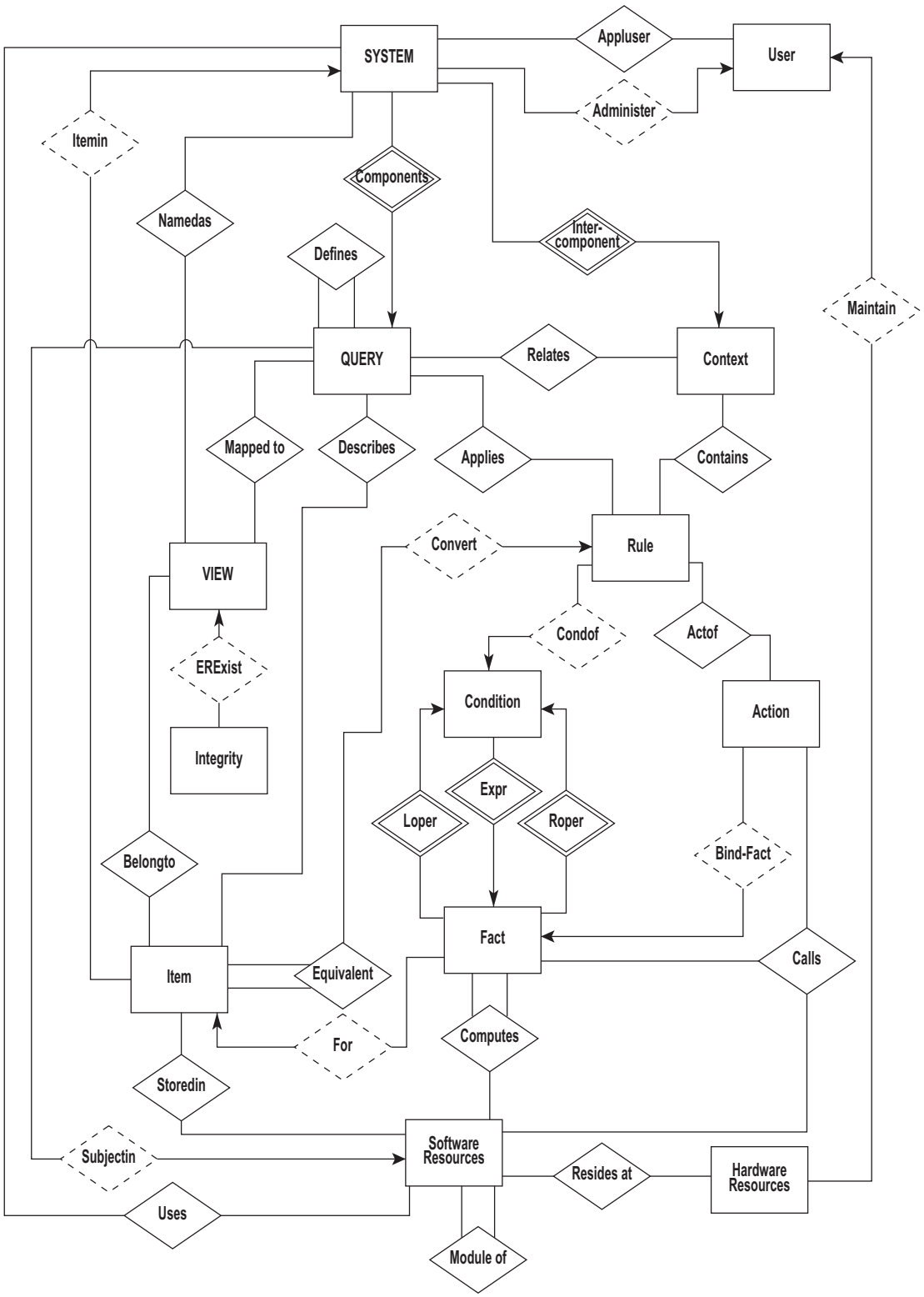

Figure 3: Conceptual Structure of the Query Database

Albeit based on the GIRD model, the above conceptual schema relaxes some of the GIRD requirements for the purposes of Blackboard processing. The main changes to the GIRD model take place at the SYSTEM, QUERY, and VIEW meta-entities, which replace the APPLICATION, SUBJECT and ENTREL meta-entities in the original version. The changes are summarized below.

The SYSTEM meta-entity identifies the enterprise databases that are currently participating in global query, and accordingly the export databases that represent them. Each export database is defined by a unique identifier, which is determined at design-time when the local data model is integrated into the Metadatabase. The QUERY meta-entity identifies the queries submitted by the export database. Each query submitted to the Blackboard is associated with a unique identifier that is assigned at run-time, along with a timestamp. The related COMPONENTS meta-MR associates queries with a particular export database and upholds existence and dependency integrity. The VIEW meta-entity is an alias for the QUERY meta-entity, analogous to the traditional definition of a database *view*. Indeed, the conceptual model provides this opportunity, since an export database can submit more than one query to the Blackboard. It is important to note that there cannot be multiple instances of unique identifiers in the Query Database. The ITEM meta-entity remains unchanged from its original definition [Hsu et al., 1991], and represents the data items specified in each query. The BELONGTO meta-PR associates data items to a specific VIEW, while DESCRIBES specifies the data items that belong to each QUERY.

The rulebase maintains its original definitions as described in [Bouziane and Hsu, 1997], although the context in which it is used has changed. In the original context, the RULE meta-entity consolidated the decision, business and operating rules in the global data model. These rules took the form, IF *condition* THEN *action*, and only operated on the data items in the Metadatabase. In its new context, the RULE meta-entity consolidates the various constraint types and actions as defined in a query. The fact that a constraint takes the form of an operation between an attribute or data item, and literal value in the case of negotiation and selection constraints respectively, and between data items in the

case of join constraints, is depicted in Figure 3 by the CONDITION meta-entity. It abstracts the negotiation, selection and join constraints, while the FACT meta-entity provides additional details about the components of this abstraction.

IV.2 THE SYNTAX OF EXMQL

The exMQL provides a uniform query format for the various query operations required. The extensions from the original MQL are concerned mainly with the new publication provisions for the collaboration, and the rule specification. The full syntax specification is illustrated in Figure 4.

The GET and PUT commands specify a subscribing query (information request) and publication query (information offer), respectively. The FOR command specifies constraints on the data items specified in the query, as well as constraints on the query in general. Three classes of constraints are considered: selection conditions (C^S), join conditions (C^J), and negotiation conditions (C^N). They are used in the evaluation of a match and in the processing of the query. Multiple conditions are conjoined by the logical operators AND and OR. The DO command is used to specify the procedural actions of a query. An action can be associated with a particular condition, and accordingly will be executed if the condition is determined to be true. In addition, an action can be associated with a query in general, and will be executed on the successful match of a query. The specification of actions in a query is optional.

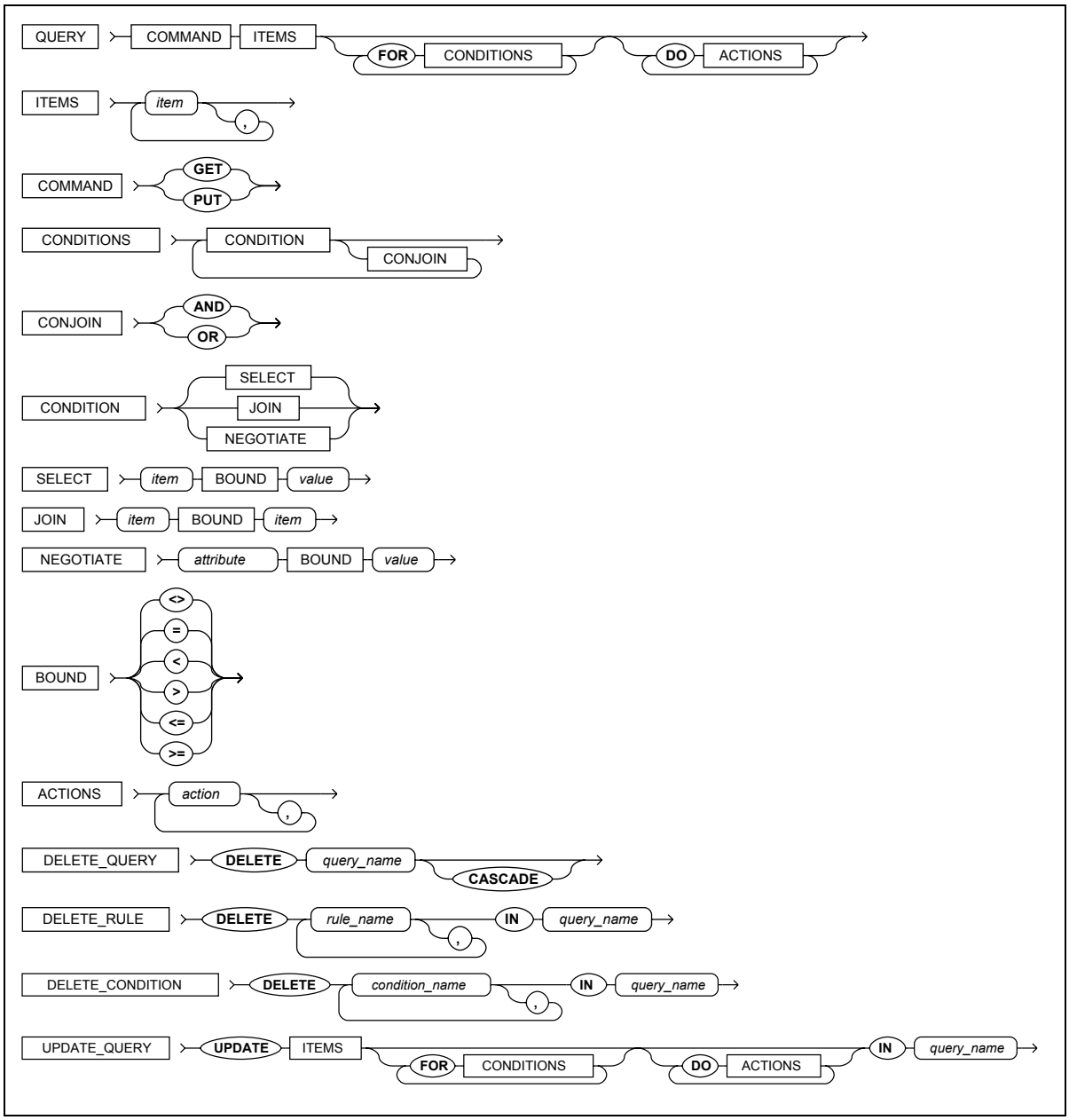

Figure 4: Extended Metadatabase Query Language Syntax

V. DESIGN EVALUATION: THE FEASIBILITY OF INFORMATION MATCH FOR COLLABORATION

A prototype of the Global Blackboard has been created in a laboratory environment – see [Levermore and Hsu, 2006] for details. The objective of the prototype was to reduce the concept of the Information Match model to practice: proving the technical correctness of matching methods (algorithms, language,

and, Query Database design) by virtue of their actual software implementation. We tested the prototype, using both subscription and publishing queries in a set of experiments to establish the soundness of the new model. The experiments focused on the computational correctness of the algorithms. A theoretical analysis on the computational complexity asserted its efficient performance and amiable scalability - see Appendix III. On this basis, the supply chain scenario of Section I.1 indicates how the prototype may flexibly connect independent databases and facilitate on-demand information supply chains.

The overall prototype consists of two basic designs: The new Information Matching model sits logically on top of a previously established Metadatabase system [Hsu, et.al. 1995]. The latter comprises three application databases (product design, process planning, and shop floor control) running on three separate client computers (various relational systems), in addition to the Metadatabase itself. The Metadatabse (the GIRD of these application systems) runs on a relational database server (Oracle on Unix), completed with a Metadatabase Management System (including the query language MQL and the distributed shells system ROPE at these local systems) running as added shells to the database engine. This Metadatabase system had been tested extensively, including testing at some industrial sites as well as in the laboratory, and reported in a number of previous archival publications (e.g., [Hsu, et. al. 1995]).

The Global Blackboard in Figure 1, which implements the Information Matching model in the prototype, was resulted from the following components: new match algorithms added to the Metadatabase Management System; exMQL obtained from extending the previous MQL; and Query Database created according to Figure 3. These components utilized the Fedora Linux Core 2 operating system running on a dual processor Dell workstation with Pentium 3 CPU, 900 MHz, and PostgreSQL Version 7.4.7, Apache and PHP. The Blackboard used the programming facility provided by PostgreSQL which served as the underlying database management system to both the Metadatabase and the Query Database (including the rulebase), to implement the match algorithms. The prototype used only the core functions of PostgreSQL (including PL/pgSQL),

which are generally compliant to the standard relational model and the established ANSI requirements. No implementation-specific optimization was employed. The prototype employed embedded PHP functionality to connect the Metadatabase and the Blackboard to Web environments. As such, the prototype offered a Web-based user interface via which both subscribing and publishing queries were constructed from personal PC as simulated users.

The experiments were designed according to the syntax of exMQL and the types of computation required of the algorithms. In other words, the design was not concerned with obtaining statistical analyses of, say, performance under various conditions – since a theoretical analysis was conducted to assert the absolute general property of the algorithms for the purpose. Instead, the design was based on the variety of queries that the model has to process – to assert that they are performing correctly as designed.

Therefore, the experiments entailed a set of representative queries: the inclusion of new exMQL provisions of PUT, GET, and conditions, and the processing of partial matching, full matching, time-delayed matching, and a variety of other computational situations. Collectively, they tested the correct coverage of subscription and publication, correct matching, and correct query processing. These queries tested these provisions individually and in combination, with their results checked throughout the intermediary steps. For this purpose, the Metadatabase was populated with previous metadata of product design models, process planning models, and shop floor control models. Although only one instance per query type was really required for the purpose of computational verification, a number of subscribing and publishing query instances were randomly generated for checking, repetitively, for the full range of the exMQL syntax. All these experiments also represented generic queries that could appear in cases of information sharing in the supply chain scenario. Since the testing queries were submitted via Web connections, they emulated exchange of exMQL expressions of information sharing between client computers and the Blackboard.

The Global Blackboard correctly processed all matches as designed and required for all experiments. The Query Database managed all these queries and supported the matching. All matching results were checked to be accurate. All intermediary results were confirmed to be correct at each step as calibrated by manual verification. The computational results showed that the prototype was fully integrated with the underlying database management system as designed (e.g., operating as a PostgreSQL application), and thereby proved the correctness of the software implementation. On this basis, the prototype establishes the basic computational correctness of the software design of the match model: exMQL, the Query Database, and match algorithms.

Now we consider what the prototype shows in the way of collaboration in a supply chain. For the design objective scenario formulated in Section I.1, we simulate that the Metadatabase system represents conceptually an extended enterprise of supply chains in this way: the process planning database being an export database residing at the manufacturer (P1), the product design database an export database at a prime (e.g., P4), and the shop floor control database an export database at a supplier to the manufacturer (e.g., P2). The collaboration of these three independent databases represents a virtual digitally connected (extended) enterprise to conduct Computer-Integrated Manufacturing (CIM) among these three companies. Previous CIM designs would require hard connections of the enterprise databases at P1, P2, and P4; which would incur inhibitive complications. The Information Match model, however, allows them to volunteer only the export databases in the collaboration. They post their information needs as requests (publish or subscribe) for others to match, and each match results in an information supply chain (requirement and execution) for the virtual CIM. Collaboration becomes more feasible, technically, at least.

This design makes information sharing easier since the manufacturer can pull as well as push its information needs to P2 and P4 without having to know exactly what they have to offer and in what formats, as traditional global query systems require. The same applies to P2 and P4, as well. Each only needs to post its requests (publish or subscribe) from its own perspective using the proxy

as it sees fit, and can change the request dynamically. For example, the manufacturer can switch the information supply chains between those for Part A and those for Part D as easily as posting the requests on either. Disruptions would be mitigated, too, since the companies can evolve their metadata and export databases to accommodate changes in, e.g., their production databases; and they can also form alternative information supply chains (matches) should failure of existing ones occurs. Needless to say, P3 and P5, and for that matter other interested companies, could also join in the same manner and expand the possibilities of their on-demand collaboration relation.

The prototype confirms that the Information Match model, along with the Metadatabase, can be implemented as an open system. The theoretical scalability of the design is reviewed in the Appendix III. The analysis suggests that the match algorithm has linear complexity, $O(n)$, proportional to the number of queries involved. This is a favorable performance for any information matching method, especially when complex data semantics also have to be considered. Since matching represents the major added cost to the extended global query processing method, its linear performance verifies that the match model is as scalable as the previous global database query models.

VI. CONTRIBUTIONS AND DISCUSSION

The new Information Matching model achieves the design goal of providing open and scalable connections of independent databases in digitally connected enterprises, especially those that are massively extended such as global supply chain. In particular, it contributes to mitigating disruptions and facilitating collaboration and information sharing in information supply chains. This class of capability adds to the ability of designing new information systems to support new business designs that scale data and knowledge resources.

The unique new technical concept here is integration of matchmaking into global database query - i.e., formulating publishing queries from community schemas and integrating them with traditional database query languages. The concept is applicable to employing any global database query model, although

the Metadatabase Model is chosen in the paper to develop the particular results. The specific results developed include a language that effectively formulates subscribing queries and publishing queries for Information Matching; an algorithm that efficiently matches these two types of queries for database collaboration; and an integrated design that automatically executes both matching and database query processing.

The claim on the language is substantiated in Section IV, with the effectiveness of the language evidenced by the extent of its syntax and its implementation in the matching algorithms, and further by the illustration of its execution in a prototype. The claim on the algorithm is substantiated in Section III (the logic) and Section V (the computing efficiency). The claim on the design is justified on the basis of its core elements (Section II). The fundamental argument is that, the use of the same metadata representation method unifies all the query language, the schema of the Query Database, and the Metadatabase; therefore, the entire transaction is simplified. The Blackboard becomes a database management system and the matching algorithms the query processing programs for the Query Database, with the results (matches) automatically becoming global database queries for execution. Other matching methods in the field do not unify the representation of bids with their processing; and previous global database query results do not support publishing queries and their proactive matching with subscribing queries.

Information Matching promises to enhance the applicability of global database query for, in particular, global supply chains. Consider the scenario, again: First, the manufacturer requires local **autonomy**. Traditional global query systems, including the previous Metadatabase Model, do not allow its databases to control when and how their data resources are utilized, beyond the addition and removal of their data models to and from the global query system. Information Matching separates the registration structure (e.g., the Metadatabase) from the negotiation structure (e.g., the Blackboard); therefore, the manufacturer's databases participate in the global query process ***only*** when

the data to be shared are made public, by submitting publishing queries to the Blackboard. Otherwise, they remain connected, but are not involved actively.

Heterogeneity, scalability and openness are interwoven; which are limited by the global model's requirements on the community semantics and semantic mapping infrastructures: e.g., how to build and maintain a global administrator, a common schema, and/or a semantic ontology. Information Matching does not alter fundamentally this situation. However, the concept of publishing queries based on the Export Database (see Section III) affords, e.g., the manufacturer more tools to manage its databases' global representations for target users. In the example, it can now take part in different federations on demand. This design eases the burden of data conversion at the global site and thereby makes it easier to accommodate heterogeneous local systems. Therefore, the design facilitates the openness and scalability of the supply chain community. The new results require only open source technology.

Several important issues remain open concerning especially the distribution design and evaluation. The Information Matching model needs empirical validation. On the theoretic front, the global blackboard and Metadatabase may be extended to support peer-to-peer inter-operations, e.g., using distributed proxies that embed a minimum Metadatabase. Better results are also possible for updating semantics in massively distributed environments.

More broadly, the implications of open and scalable connection of databases on new business designs (Section I) deserve exploration. Similar implications on new IS designs to enable new business designs also deserve study. These broad views of digital connections at the database level may be profoundly relevant to a digitally connected society.

ACKNOWLEDGEMENT

The authors wish to express their sincere gratitude to the guest editors and the anonymous reviewers for their thorough and invaluable comments. They made the paper much more readable and enhanced its accuracy.

REFERENCES

Alter, S. (2008), “Service System Fundamentals: Work System, Value Chain, and Life Cycle, “*IBM Systems Journal* (47) 1, pp. 71 – 85.

Anderson, J.C., J.A. Narus, and W. van Rossum (2006) "Customer Value Propositions," *Harvard Business Review,* March, pp. 91-99.

Babin, G. and C. Hsu (1996) "Decomposition of Knowledge for Concurrent Processing," *IEEE Transactions on Knowledge and Data Engineering* (8) 5, pp. 758-777.

Batini, C., M. Lenzerini, and S. B. Navathe (1986) "A Comparative Analysis of Methodologies for Database Schema Integration," *ACM Computing Surveys* (18) pp. 323-364.

Bowen, P., R. O'Farrell and F. Rohde (2006) "Analysis of Competing Data Structures: Does Ontological Clarity Produce Better End User Query Performance", *J. Association of Information Systems*, (7) 8 Article 22

Bayardo Jr., R. J., W. Bohrer, R. Brice, A. Cichocki et al. (1997) "InfoSleuth: Agent-Based Semantic Integration of Information in Open and Dynamic Environments," *ACM SIGMOD International Conference on Management of Data,* Tucson, Arizona, 1997.

Beynon-Davies, P., L. Bonde, D. McPhee, and C. B. Jones (1997) "A Collaborative Schema Integration System," *Computer Supported Cooperative Work: The Journal of Collaborative Computing* (6) 1, pp. 1-18.

Bitner, M. and S. Brown (2006) "The Evolution and Discovery of Services Science in Business Schools," *Communications of the ACM*, (49) 7, pp. 73-78.

Bouziane, M. and C. Hsu (1997) "A Rulebase Management System Using Conceptual Modeling Tools," *J. Artificial Intelligence* (6) 1, pp. 37-61.

Braumandl, R., M. Keidl, A. Kemper, D. Kossmann et al. (2001) "ObjectGlobe: Ubiquitous query processing on the Internet," *The VLDB Journal The International Journal on Very Large Data Bases* (10) 1, pp. 48-71.

Cambridge Papers (2008) Succeeding through Service Innovation: a service perspective for education, business and government, University of Cambridge and IBM Corp.

Chesbrough, H. and J. Spohrer (2006) "A Research Menifesto for Services Science", *Communications of the ACM*, (49) 7, pp. 35-40.

Cheung, W. and C. Hsu (1996) "The model-assisted global query system for multiple databases in distributed enterprises," *ACM Transactions on Information Systems* (14) 4, pp. 421-470.

Cingil, I. and A. Dogac (2001) "An Architecture for Supply Chain Integration and Automation on the Internet," *Distributed and Parallel Databases* (10) 1, pp. 59-102.

Collins, J., C. Bilot, M. Gini, and B. Mobasher (2001) "Decision Processes in Agent-Based Automated Contracting," *IEEE Internet Computing* (5) 2, pp. 61-72.

Davenport, T. H and J. D. Brooks, (2004) "Enterprise Systems and the supply chain," *Journal of Enterprise Information Management* (17) 1, pp. 8-19.

Dhar, V. and A. Sandararajan (2007) "Information Technologies in Business: A Blueprint for Education and Research," *Information Systems Research*, (18) 2, pp. 125-141.

Elmasri, R. and S. Navathe (2000) *Fundamentals of database systems, 3rd ed.*, 3rd ed edition. Reading, Mass: Addison-Wesley.

Erl, T. (2005) *Service-Oriented Architecture: Concepts, Technology and Design*, Prentice-Hall, Upper Saddle River, NJ.

Fonseca, F. and J. Martin (2007) "Learning The Differences Between Ontologies and Conceptual Schemas Through Ontology-Driven Information Systems", *J. Association of Information Systems*, (8) 2 Article 4

Garcia-Molina, H., J. D. Ullman, and J. Widom (2002) *Database systems: the complete book*. Upper Saddle River, NJ: Prentice Hall.

Gebauer, J. and F. Schober (2006) " Information System Flexibility and the Cost Efficiency of Business Processes", *J. Association of Information Systems*, (7) 3 Article 8

Glushko, R., J. Tenenbaum, and B. Meltzer (1999) "An XML Framework for Agent-Based e-Commerce," *Communications of the ACM*, (42) 3, pp. 106-114.

Granados, N., A. Gupta, and R. Kauffman (2006) "The Impact of IT on Market Information and Transparency: A Unified Theoretical Framework," *Journal of the AIS*, (7) 3, pp. 148-178.

Haas, L. M., M. A. Hernandez, H. Ho, L. Popa et al. (2005) Clio grows up: from research prototype to industrial tool, in *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, Baltimore, Maryland: ACM Press, pp. 805-810.

Halevy, A. Y. (2001) "Answering queries using views: A survey," *The VLDB Journal* (10) 4, pp. 270-294.

Hevner, A.R., S.T. March, J. Park, and S. Ram (2004) "Design Science in Information Systems Research," *MIS Quarterly*, (28) 1, pp. 75 - 105.

Hsu, C., M. Bouziane, L. Rattner, and L. Yee (1991) "Information Resources Management in Heterogeneous, Distributed Environments: A

Metadatabase Approach," *IEEE Transactions on Software Engineering* (17) 6, pp. 604-625.

Hsu, C., Y. Tao, M. Bouziane, and G. Babin (1993) "Paradigm Translations in Integrating Manufacturing Information Using a Meta-Model," *Journal of Information Systems Engineering*, (1)1, pp. 325-352.

Hsu, C., J. Cho, L. Yee, and L. Rattner (1995) "Core Information Model: A Practical Solution to Costly Integration Problems," *Computer and Industrial Engineering*, Vol. 28, No. 3, pp. 523-544.

Hsu, C., C. Carothers, and D. M. Levermore (2006) "A Market Mechanism for Participatory Global Query: A First Step of Enterprise Resource Allocation," *Information Technology and Management* (7) 2, pp. 71-89.

Hsu, C., D. M. Levermore, C. Carothers, and G. Babin (2007) "Enterprise Collaboration: On-Demand Information Exchange Using Enterprise Databases, Sensor Networks, and RFID Chips," *IEEE Transactions on Systems, Man, and Cybernetics Part A*, (27) 4, pp.519-532.

Hsu, C. (2007) "Cyber-Infrastructure-Based Enterprises and Their Engineering," in C. Hsu ed. *Service Enterprise Integration: an Enterprise Engineering Perspective*, Springer Academic Publishers, Lowell, MA, pp. 209-243.

Hsu, C. and J. Spohrer (2009) "Improving Service Quality and Productivity: Exploring the Digital Connections Scaling Model," *International Journal of Service Technology and Management*, in press.

Hsu, C. (2009) *Service Science: Design for Scaling and Transformation*, World Scientific and Imperial College Press, Singapore, 2009

Kalfoglou, Y. and M. Schorlemmer (2003) "Ontology Mapping: the State of the Art," *The Knowledge Engineering Review* (18) 1, pp. 1–31.

Kim, H., M. S. Fox, and A. Sengupta (2007) "How To Build Enterprise Data Models To Achieve Compliance To Standards Or Regulatory Requirements (and share data)", *J. Association for Information Systems*, (8) 2, Article 5.

Kossmann, D. (2000) "The state of the art in distributed query processing," *ACM Computing Surveys* (32) 4, pp. 422-469.

Kurbel, K. and L. Loutchko (2003) "Towards Multi-Agent Electronic Marketplaces: what is there and what is missing?," *The Knowledge Engineering Review* (18) 1, pp. 33-46.

Levermore, D. M. and C. Hsu (2006) *Enterprise Collaboration: On-Demand Information Exchange for Extended Enterprises*, Springer Science Publishers, Lowell, MA.

Madhavan, J. and A. Halevy. (2003) Composing Mappings Among Data Sources. *29th VLDB Conference, Berlin, Germany, 2003*.

Maes, P., R. H. Guttman, and A. G. Moukas (1999) "Agents That Buy and Sell," *Communications of the ACM* (42) 3, pp. 81.

Mena, E., V. Kashyap, A. Sheth, and A. Illarramendi (2000) "OBSERVER: An Approach for Query Processing in Global Information Systems Based on Interoperation Across Pre-Existing Ontologies," *Distributed and Parallel Databases* (8) 2, pp. 223-271.

Miller, R., L. Haas, and M. Hernandez. (2000) Schema Mapping as Query Discovery. *26th VLDB Conference, Cairo, Egypt, 2000*.

Palmisano, S.F. (2006) "The Globally Integrated Enterprise," *Foreign Affairs,* (85) 3, pp. 127-136.

Rahm, E. and P. A. Bernstein (2001) "A Survey of Approaches to Automatic Schema Matching," *The VLDB Journal* (10) 4, pp. 334-350.

Rodríguez-Martínez, M. and N. Roussopoulos. (2000) MOCHA: a Self-Extensible Database Middleware System for Distributed Data Sources. *2000 ACM SIGMOD International Conference on Management of Data, Dallas, Texas, 2000*.

Sheth, A. and J. A. Larson (1990) "Federated Database Systems for Managing Distributed Heterogeneous and Autonomous Systems," *ACM Computing Surveys* (22) 3, pp. 183-236.

Shvaiko, P. and J. Euzenat (2005) "A Survey of Schema-based Matching Approaches," *Journal on Data Semantics*.

Silberschatz, A., H. F. Korth, and S. Sudarshan (2002) *Database system concepts, 4th ed*. Boston: McGraw-Hill.

Sim, K. M. and E. Wong (2001) "Toward Market-Driven Agents for Electronic Auction," *IEEE Transactions on Systems, Man and Cybernetics, Part A* (31) 6, pp. 474-484.

Spohrer, J. and P.P. Maglio (2007) "The Emergence of Service Science: Towards Systematic Service Innovation to Accelerate Co-creation of Value," *Production and Operations Management,* in press.

Stonebraker, M., P. Aoki, A. Pfeffer, A. Sah et al. (1996) "Mariposa: A Wide Area Distributed Database System," *International Journal on Very Large Databases* (5) 1, pp. 48-63.

Sycara, K., M. Paolucci, M. van Velsen, and J. Giampapa (2003) "The RETSINA MAS Infrastructure," *Autonomous Agents and Multi-Agent Systems* (7) 1-2, pp. 29-48.

Tao, L. (2001) "Shifting Paradigms with the Application Service Provider Model," *IEEE Computer,* (34) 10, pp. 32-39.

UN/CEFACT, United Nations Center for Trade Facilitation and Electronic Business, Core Components Technical Specification, version 2.01, November 15, 2003.

Wisner, J. D. and K. C. Tan (2000) "Supply Chain Management and Its Impact on Purchasing," *The Journal of Supply Chain Management* (36) 4, pp. 33-42.

Zhao, J.L., C. Hsu, H. J. Jain, J. Spohrer, M. Taniru, and H. J. Wang (2008) "ICIS 2007 Panel Report: Bridging Service Computing and Service Management: How MIS Contributes to Service Orientation?" *Communications of the Association for Information Systems*, Vol. 22, pp. 413-428.

APPENDIX: Glossary, Global Query Processing, and Complexity

I. Glossary

Massively Extended Enterprise: an extended enterprise that includes many organizations, such as the collaboration along demand chains and supply chains.

Digitally Connected Enterprise: an enterprise or extended enterprise that connects people, processes, and resources by digital means.

Independent Database: a proprietary database owned and designed by individual organizations in digitally connected enterprises.

Metadata: data about data; such as file headers and trailers, able definitions, entity-relationship models, object definitions, rules, and database catalogs.

Metadatabase: a relational database of metadata.

GIRD (global information resources dictionary): the schema of the Metadatabase constructed according to TSER; which includes four basic sub-models (application-user, subject-entity-relationship-data, context-rule-condition-action-fact, and software-hardware) with each consisting of a fix set of meta-relations.

TSER: two-stage entity-relationship, a data and knowledge modeling method using six basic concepts: data item, rule, entity, relationship, subject, and context, and a mapping method to correspond to relational and object models.

Data Item: attributes or variables pertaining to rules, entities, and relationships (comparable to attribute in the general entity-relationship-attribute model).

Rule: the usual if-then representation of causal relationship; predicate logic.

Entity: nouns of conceptualization, such as person, place, thing, time, etc. (defined with normalized collection of data items; otherwise comparable to the usual concept of entity in the general entity-relationship-attribute model).

Relationship: three types of association of entities: PR or plural relationship (defined with normalized collection of data items), FR or functional relationship (defined with referential integrity), and MR or mandatory relationship (defined with existence dependency rules – comparable to the usual weak entity).

Subject: application-oriented encapsulation of rules, entities, and relationships (comparable to the usual concept of objects in the general object orientation).

Context: application-oriented encapsulation of rules that define interactions among subjects.

II. Technical Analysis of the Research Problem

The fundamental differences between the new model presented here and its comparable results in the literature are that previous global query languages do not support automated matchmaking between multiple subscribing queries and multiple ad hoc provisions of data resources (i.e., publishing queries) over a participant-specified period of time; while the previous matchmaking results are not integrated with global query processing. They are insufficient to resolve the problem of open and scalable collaboration of independent databases.

Consider the case of using traditional global database query results alone for Information Matching. The only publication mechanism in this case is the registration of the database schemas (which are inherently static); and the subscription is accomplished through some global query languages using these

schemas. We submit that the problem is inadequate matching. The field offers two basic approaches to coordinating the schemas of distributed, heterogeneous, and autonomous databases: global integration/federation and peer-to-peer schema mapping [Batini et al., 1986, Bayardo Jr. et al., 1997, Haas et al., 2005, Kalfoglou and Schorlemmer, 2003, Madhavan and Halevy, 2003, Miller et al., 2000, Rahm and Bernstein, 2001, Shvaiko and Euzenat, 2005]. The global/common schema approach affords accuracy, but faces the challenge of maintaining the global consistency of data semantics among local participants and of administering the (often layered) structure of the global schema itself [Beynon-Davies et al., 1997, Sheth and Larson, 1990, Stonebraker et al., 1996]. The peer-to-peer mapping, on the other hand, requires the development and maintenance of comprehensive ontology and industrial standards for the entire community [Bowen, et.al. 2006, Braumandl et al., 2001, Fonseca and Martin 2007, Halevy, 2001, Kim et.al. 2007, Kossmann, 2000, Mena et al., 2000, Rodríguez-Martínez and Roussopoulos, 2000]. Both approaches require the participants to register with the community, which in turn maintains a global schema and/or the community ontology. The schema matching approach promises to avoid the hard task of global schema administration at the expense of requiring community ontology, which is not always available or even feasible. In any case, the local data as represented in the registered schemas are constantly available for querying by information users. The information providers are not supported with the control and flexibility to proactively ***offer random (or, ad hoc) data resources*** seeking users. They cannot automatically join force with other providers on an ad hoc basis to satisfy users, including ***combining multiple publishing queries*** for one or multiple subscribing queries, and combining multiple participants to form ad hoc federations on demand.

To provide ad hoc information provisions, information providers would have to rely on database views (materialized or not, distributed or not, and pro forma or not), to frequently add, delete, and modify them at run time. Two technical problems arise in this case. First, database views are matched to queries in a static, “yes or no” manner, at an instant. They do not support

automatic matching against ***pending global queries*** under prescribed matching rules, through a query language, a software agent, or similarly flexible procedures. Second, to add, delete, or modify these views on the fly could incur non-trivial maintenance overhead in many global schema/ontology methods (except designs that employ a database to manage the metadata as ordinary database operations [Cheung and Hsu, 1996, Hsu et al., 1991]). Furthermore, registering ad hoc views could require the cooperation of the local authorities of the participating databases. The complexity could increase as a combinatorial function of the number of simultaneous changes (ad hoc data provisions).

The ***technical problem*** here is how to support the information providers to publish their on-demand database provisions (dynamic views), including the accompanying conditions, without triggering global schema maintenance. The dynamic views must support the ensuing global query processing, as well. This added layer of publishing queries enhances local control and flexibility, and facilitates on-demand, open and scalable collaboration of independent databases, as the stated design goals require. For example, participants of a supply chain can “open” their databases through these easy dynamic publications, while preserving their proprietary control, to allow collaborators retrieving on-demand data for scheduling and other transactions.

The case of using matchmaking alone to perform Information Matching also faces undesirable requirements and inadequate results. Take the industrial exchanges that support supply chain management (e.g., covisint.com, ariba.com, and perfectcommerce.com) as examples. They focus only on the swapping of the document objects (usually represented in XML) accompanying the bids and do not handle global database querying (which requires synchronization of the semantics in the XML code). More generally, the field has focused on the self-allocation of resources to users, leaving the ensuing tasks of processing the allocated resources largely to the trading systems. The matchmaking results are generally not compatible to being used directly as queries for global database query processing. Significant manual processing and overhead are required to connect these two phases. Another performance concern is computing efficiency

of the matchmaking methods themselves. Major results in the field tend to employ software agents to perform matchmaking [Collins et al., 2001, Kurbel and Loutchko, 2003, Maes et al., 1999, Sim and Wong, 2001, Sycara et al., 2003]. However, their specific models and designs of agents may not provide definitive reference points to allow for analysis of their complexity; for instance, they may convolute proprietary technology with their own data structures. The ***technical problem*** here is how to integrate matchmaking into global query processing with acceptable computing performance, for efficient Information Matching.

These two technical problems are in fact complementary. Global database query methods provide flexible processing of distributed, federated databases (for, e.g., flexible retrieval of supply chain data from participants' proprietary databases beyond using hard-coded protocols); but it lacks flexible provision views and matching (for, e.g., establishing the requirements of the retrieval). Matchmaking, on the other hand, provides flexible provisions and matching, but lacks flexible database processing.

Therefore, the overall research problem is how to integrate matchmaking into global database query methods and thereby solve the problems mentioned above. More specifically, the new matching model needs to ***unify the representation, expression, and processing of subscribing and publishing queries, separated from but based on the global schema, with computing efficiency***. It must also support ***group matching*** (e.g., combinations of publishing queries) and ***prolonged matching*** (i.e., matching over a participant-specified period of time), and include the ***matching rules and constraints*** in the queries. As such, the new model will be amenable to software agent technology as well as the database technology, and will facilitate the many-to-many relationships (federations) among users (collaborators), on an on-demand basis.

The basic solution approach employed in the research is to extend the global database query results that are already amenable to supporting large numbers of independent enterprise databases over the Internet. For the purpose of this research, we employed the Metadatabase Model as the foundation

because it provides an open and scalable way to administer the global schema in a federation manner. Moreover, the model simplifies the maintenance of global data semantics by making the task one of registration of peer-to-peer data equivalence. This approach is amenable to the application of community ontology when one exists. In this sense, the model represents a robust choice.

The extensions formulate ad hoc information provisions as publishing queries, consistent with traditional database queries, to allow matchmaking to be performed as relational query processing and thereby simplify the matching and the integration with global database query processing. The specific contributions include (1) a language for participants to formulate subscribing and publishing queries, (2) an algorithm to match multiple subscribing queries and multiple publishing queries with desirable performance, and (3) a design to execute the matched database queries using previously available results.

III. Performance Analysis: Assessing Matching at the Global Blackboard

The field does not have a common measure to meaningfully compare the new information matching model with the numerous results in matchmaking (e.g., [Collins et al., 2001, Kurbel and Loutchko, 2003, Maes et al., 1999, Sim and Wong, 2001, Sycara et al., 2003]). Therefore, we contend with an analysis of the theoretical computing performance of the new algorithms, as an absolute way to justify the relative merit of the new model. In this section, we assess the core operations of the new matching algorithms.

We use relational algebra to determine acceptable query trees and identify the query plans. The quantitative part of the analysis is based on an implementation on the PostgreSQL mentioned in Section IV. We use a generic analysis of the matching algorithms to assess the cost estimate with the worst case number of pages/blocks transferred from disk during a database query (matching). This is a standard measure of database performance [Elmasri and Navathe, 2000, Garcia-Molina et al., 2002, Silberschatz et al., 2002]. We exclude other query costs that are irrelevant to the settings, difficult to acquire, and/or platform-specific. The model of the analysis is shown below:

```
SELECT D.QNAME, COUNT(D.ITEMCODE)
    FROM describes AS D, query AS Q
    WHERE ITEMCODE
        IN (itemcode_list)
        AND D.QNAME = Q.QNAME
        AND Q.TYPE ≠ query_type
        GROUP BY D.QNAME;
```

$$\pi_{qname,\,itemcount}\left(\begin{array}{l}\pi_{qname}\left(\sigma_{type \neq \text{QUERY\_TYPE}}(Q)\right)\\ \infty_{Q.qname = D.qname}\\ {}_{qname}\Im_{COUNT\ itemcode}\left(\pi_{qname,\ itemcode}\left(\sigma_{itemcode\ \in\ \text{ITEMCODE\_LIST}}(D)\right)\right)\end{array}\right)$$

The above Analysis Case depicts an SQL query and its relational algebraic expression that corresponds to the *matching algorithm* (see Section III). The expression depicts an acceptable query plan for the algorithm, in that it moves the select operation to the bottom of the query tree, uses equi-joins to join tables, and projects necessary attributes when possible. The size of the QUERY table (*Q*), and the DESCRIBES table (*D*) are restricted by applying the selection conditions, thus reducing the size of the relations participating in joins. Note also that a non-standard symbol $\Im$ is employed to describe the GROUP BY clause – the prefix indicates the attribute the query should be grouped on, whereas the suffix indicates the aggregate functions applied to the adjacent attribute.

Table IV summarizes the parameters used in the analysis. The sizes include only essential attributes such as {QNAME, QTYPE, TIMESTAMP} in the QUERY table which are required for query matching, and {QNAME, ITEMCODE} in DESCRIBES table. The page/block size of 8192 bytes is a PostgreSQL parameter.

Table IV: Parameters for the Blackboard Database.

Feature	*Value*
Cardinality, *Q*	|*Q*|
Cardinality, *D*	|*D*|
Page Size/Block Size	8192 bytes
Tuple Size, *Q*	118 bytes
Tuple Size, *D*	200 bytes

The blocking factor (*bfr*) defines the number of records that are contained in a block, and so it is possible to determine the number of blocks required for each table, which is a function of the number of tuples in a table. Accordingly,

$$bfr_Q = \lfloor 8192/118 \rfloor \Rightarrow b_Q = \lceil |Q|/bfr_Q \rceil = \lceil |Q|/69 \rceil$$

$$bfr_D = \lfloor 8192/200 \rfloor \Rightarrow b_D = \lceil |D|/bfr_D \rceil = \lceil |D|/40 \rceil$$

Given this information, a cost estimate for the matching algorithm is determined, taking into consideration that the nested-loop join algorithm is employed in the query join. We denote the initial state of the query plan as state 1, and the finishing state of query plan as state 2, then the cost of the query is found as [Garcia-Molina et al., 2002],

$$C = \frac{r_2}{bfr_2} + \left(\frac{r_2}{bfr_2} * \frac{r_1}{bfr_1} \right)$$

where r_1 refers to the results at the beginning of the query plan – i.e., the number of records for the result of the top-most sub-query, and r_2 refers to the number of records for the bottom-most sub-query in the query plan (the finishing of the query plan). Similarly, $b_1 = \lceil r_1/bfr_1 \rceil$ and $b_2 = \lceil r_2/bfr_2 \rceil$, are the number of blocks required for each result set, respectively.

The biggest contributor to the cost of the matching algorithm is the nested-loop join algorithm, and so adjustments to improve the matching performance are

made here first. The alternative sort-merge algorithm will introduce a cost: $C = b_2 + b_1$, essentially a cost having linear complexity $O(n)$; but this requires that the corresponding input tuples, r_1 and r_2 are sorted on the join attribute QNAME, which currently is not guaranteed. The sorting in the sort-merge join operation increases the associated cost shown below [Elmasri and Navathe, 2000] due to the fact that the sort-merge algorithm must make multiple passes on r_1 and r_2, first to sort then to merge. The estimate also includes the cost to write the results back to disk.

$$C = (2 * b_2 * (1 + \log_2 b_2)) + (2 * b_1 * (1 + \log_2 b_1)) + b_2 + b_1$$

By choosing this adjustment, the performance complexity of the matching algorithm then becomes at most $O(n \log n)$.

As indicated above, the sort-merge has linear complexity if both r_1 and r_2 are already sorted. The QUERY table already contains an index on QNAME, but the WHERE clause in the select operation specifies the QTYPE attribute, which does not have an index. Therefore, a sorted result is not guaranteed. Creating a secondary index on this attribute will improve the select operation, such that $C = x + s$, where s is the selection cardinality matching ¬QTYPE, and x is the number of levels in the secondary index. A *B+-tree* search tree used for the secondary index allows for this linear complexity, $O(n)$.

The DESCRIBES table contains an index on <QNAME, ITEMCODE>, but the IN clause in the select operation leads to a disjunctive condition, which requires the union of the results from the individual conditions. A secondary index could also be applied to ITEMCODE, resulting in the similar cost derived above, but modified to include the multiple passes required by the union of the results, and also the cost required to sort on QNAME. The complexity of this operation is limited to $O(n)$.

ABOUT THE AUTHORS

David M. Levermore is currently with the Boeing Company. He earned his BS in Mechanical Engineering from Howard University in Washington DC in

1994. He received his MS in Mechanical Engineering in 1995 from Rensselaer Polytechnic Institute, and worked at the Boeing Company as a Senior Design Engineer from 1996 to 1998. He completed his doctoral studies in Decision Sciences and Engineering Systems at Rensselaer in 2005. He is a member of the IEEE Computer Society and ACM and has interests in information matching in collaborative environments, within the context of distributed database systems; and distributed information systems and applications that exploit advanced Web technologies. Email: leverd@alum.rpi.edu

Gilbert Babin received his B.Sc. and M.Sc. from Université de Montréal (Canada) in 1986 and 1989, respectively. He then completed his doctoral studies in 1993 at Rensselaer Polytechnic Institute (Troy, New York, USA), where he studied integration approaches for heterogeneous, distributed systems. His doctoral thesis earned him the Del and Ruth Karger Dissertation Award in 1995. He worked at the Computer Science department at Université Laval from 1993 to 2000. Since then, he then joined the Information Technologies Department at HEC-Montreal (Canada) as Associate Professor. Gilbert Babin is a member of ACM and the Computer Society of the IEEE. He has more than 45 papers published in refereed journals and conferences. Some of his research results may be found in the transactions of the IEEE. His research interests revolve around distributed systems and approaches to integrate them. Email: gilbert.babin@hec.ca

Cheng Hsu is a Professor of Decision Sciences and Engineering Systems at Rensselaer Polytechnic Institute, Troy, NY 12180-3590. He earned his BS from Tunghai University, Taiwan and MS-ISE and Ph.D. from the Ohio State University, Columbia, Ohio. His teaching covers databases, information systems, e-business enterprise engineering, and manufacturing. Dr. Hsu is currently working on service enterprise engineering. He is the originator of the Metadatabase Model (a model-based ontology and scalable common schema)

and the Two-Stage Entity-Relationship model (for data and knowledge systems analysis and design). Dr. Hsu has published a few books on these subjects and guest edited a special issue of IEEE SMCA on e-Commerce. His scholarly papers have appeared in a number of major journals and refereed conferences. His research has been supported by both government agencies and industry. Website: http://viu.eng.rpi.edu Email: hsuc@rpi.edu