2010

Enterprise Information Integration Using a Peer to Peer Approach

Jochen Kokemueller
Fraunhofer IAO, jochen@kokemueller.de

Anette Weisbecker
Fraunhofer IAO, anette.weisbecker@iao.fraunhofer.de

Enterprise Information Integration Using a Peer to Peer Approach

Journal:	*18th European Conference on Information Systems*
Manuscript ID:	ECIS2010-0028.R1
Submission Type:	Research-in-Progress Paper
Keyword:	Heterogeneous data integration, IS integration, Information infrastructure, Information decentralization

ENTERPRISE INFORMATION INTEGRATION USING A PEER TO PEER APPROACH

Kokemüller, Jochen, Fraunhofer IAO, Nobelstr. 12, 70569 Stuttgart, Germany, jochen.kokemueller@iao.fraunhofer.de

Weisbecker, Anette, Fraunhofer IAO, Nobelstr. 12, 70569 Stuttgart, Germany, anette.weisbecker@iao.fraunhofer.de

Abstract

The integration of enterprise information systems has unique requirements and frequently poses problems to business partners. We discuss specific integration issues for micro-sized enterprises on the special case of independent sales agencies and their suppliers. We argue that the enterprise information systems of those independent enterprises are technically best represented by equal peers.

Therefore, we have designed the Peer-To-Peer (P2P) integration architecture VIANA for the integration of enterprise information systems. Its architecture provides materializing P2P integration using optimistic replication. It is applicable to inter- and intraorganizational integration scenarios. It is accomplished by the propagation of write operations between peers. We argue that this type of integration can be realized with no alteration of the participating information systems.

Keywords: Enterprise Information Integration, Peer-to-Peer, Materialized Integration, Optimistic Replication

1 INTRODUCTION

Enterprises Information Systems (EIS) grow increasingly interdependent and require information integration. Here we refer to systems that support the enterprise in its value creation. Typically they are Enterprise Resource Planning (ERP), Customer Relationship Management (CRM) and likewise systems. Bernstein and Haas (2008) refer to the integration of EIS as the "biggest and most expensive challenge that information-technology shops face". Additionally, agile markets demand that EIS stay adaptable to changing business needs (Merritt and Newell 2001). This requires adaptable information integration architectures. Halevy, Rajaraman and Ordille (2006) observe two inhibiting aspects on the integration of interorganizational data: (i) Even though companies want to share data, they don't want to take the responsibility of maintaining an integrated schema and mapping sources to it. This aspect can be addressed by the use of standards. (ii) Furthermore, it is not even clear that a single integrated schema can always be developed. At times the interests of the involved parties are too diverse to get their schemas integrated.

Conway (1968) observed that a design artifact always resembles the design of an organization. In this contribution we address this by a networked structure where all participants are equal with respect to what they are able to do. This Peer-to-Peer collaboration (Walter, Werth and Loos 2006) reflects the organizational structure of autonomous enterprises that directly and equitable share information and are responsible for the integration to their neighbors. Due to the homomorphism between organizational and IT design that the P2P approach creates we expect higher acceptance by small autonomous enterprises.

Design science research contributions present novel Information Systems (IS) artifacts and suitable evaluation approaches that address the artifact's appropriateness to contribute to the problems' solution (Nunamaker Jr, Chen and Purdin 1991). These two facets of rigorous design science-oriented research contribute to the foundations and the methodologies pool of Information Systems research, i.e. they contribute to its knowledge base (Hevner, March, Park and Ram 2004). In our work we follow this research paradigm.

The remainder of the paper is organized as follows. In the next section we discuss related work. We then continue in Section 3 with an extensive description of the problem domain. In Section 4 we derive requirements. We address these requirements by proposing the integration architecture VIANA in Sections 5. We discuss the architecture in Section 6 and conclude in Section 7.

2 RELATED WORK

Several design patterns for enterprise integration can be observed (Schwinn and Schelp 2005). Currently, much attraction is given to the Service Oriented Architecture (SOA) paradigm. It has the advantage that due to virtual integration always the most recent data is accessed. Following a process oriented enterprise a SOA allows the definition of business processes and then supposedly neatly fits technical calls into the business perspective. Practice shows however that this is not the case, technical and business model are kept separate as the view on the process and the requirements for modeling differ substantially.

Enterprise Application Integration (EAI) as another integration paradigm focuses on message driven integration (Hasselbring 2000). Product from this domain usually allow virtual and/or materializing integration and use so called connectors to access integrated systems. Integration encompasses process, information and data both inside one and spanning multiple enterprises. The integration is organized around a central piece of software functioning as a single hub or as a bus (Puschmann and Alt 2004). Both variants give the benefit, that they provide a single interface for the integration and hide other integrated systems. Unfortunately, a hub scales poorly and forms as single point of failure

endangering the overall integration. On the other hand a bus, also frequently used in SOAs, is a very complex piece of software which is difficult to administer (Erasala, Yen and Rajkumar 2003). Both variants though do not work decentralized, as the reality of small interoperating companies is. Furthermore, it is not clear if even for a simple process the creation of an additional application is easier than an integration in any other approach (Haas 2006).

The integration using a Peer-to-Peer (P2P) topology has been studied in Peer Data Management Systems (PDMS). Many research prototypes of PDMS have been realized. Most of them focus on virtual integration of read operations, as: The Piazza project (Halevy, Ives, Madhavan, Mork, Suciu and Tatarinov 2004) that uses schema mediation for distributed querying or PeerDB (Ng, Ooi and Tan 2003) that employs agent based technologies to retrieve information. Specialized on large-scale web information exchange and retrieval is Peer (Huebsch, Chun, Hellerstein, Loo, Maniatis, Roscoe, Shenker, Stoica and Yumerefendi 2005). All of the above systems are insufficient for EIS integration as they do not provide mechanisms for federating write operations. A system that allows write access for scientific and academic data sharing is Orchestra (Ives, Khandelwal, Kapur and Cakir 2005). Yet, it does not use XML Technologies, thus does not facilitate a unified processing environment using standards like XQuery (W3C 2007) or XSLT (W3C 1999). Additionally, while the update execution on request might be desirable in life sciences, it is not sufficient for enterprise scale operational data stores where updates need to be executed immediately.

Avoiding data redundancy as in virtual integration (SOA, most EAI approaches, PDMS) can result in some drawbacks (Schwinn and Schelp 2005):

- The availability of all components – departmental applications, central database, EAI infrastructure, network, etc. – has to be ensured to allow operation. A failure in one component will bring the whole system down.
- All components must have high capacity. The overall system capacity has to cover the combined maximum load of all systems.
- Maintenance, further development and tests become more complex because of the higher requirements concerning availability, capacity, performance, etc.
- Splitting up the business and selling a business line is more difficult if there is a central database.

Materializing integration on the other hand needs to keep track of the references it distributes. It has to ensure that multiple references to a single real world object are merged correctly. Solutions for this challenge are reference reconciliation algorithms as discussed in the data quality literature (Rahm and Do 2000, Elmagarmid, Ipeirotis and Verykios 2007).

3 BUSINESS CASE

Independent sales agencies (ISA) are companies that represent one or more vendors. Their employees are sales agents who offer the vendor's products to customers. These products vary from standard products that can be ordered from a catalog to highly individualized products manufactured to the specific needs of one customer (Dolmetsch 2000). Independent sales agencies can be categorized in two dimensions: One dimension is their territorial exclusivity or lack of it. Sales agencies that have territorial exclusivity are the only representation of a specific vendor in a particular territory. They may still represent more than one vendor in that territory if the represented products are not competitive, but no other sales agency is permitted to represent the vendor in that territory. ISAs without territorial protection still possess customer protection. Therefore, they receive a commission if they provide at least a minor contribution to a transaction leading to a payment. The second dimension is the power of contract. ISAs that possess this power are able to act in the name of the vendor and to execute a declaration of its intention. These are legally binding to the vendor. Depending on each principal, both of the discussed dimensions can have different values for a particular ISA. An ISA generates revenue by receiving commissions for each transaction of the corresponding vendor that is

legally connected to a payment. For ISAs with territorial exclusivity all revenue created in the granted territory yield to accrued commission

The project M3V (www.m3v-projekt.de) which is funded by the Federal German Ministry of Economy and Technology focuses on the design and development of a mobile multi-supplier sales information platform which electronically supports the sales processes between ISAs and their suppliers. This mobile support system is hosted by a service provider, who integrates the legacy systems of the vendors (Kokemüller, Kett, Höß and Weisbecker 2008). Figure 1 gives an overview of this scenario.

We analyzed the business case of ISAs conducting two empirical surveys (Kokemüller et al. 2008). The major cornerstones are: ISAs have an average of 4.1 employees. Additionally 96% have no more than 5 sales agents; here the average is 1.7. Obviously, they qualify as micro-sized enterprises. In spite of that, 93% operate for more then one supplier. In fact, 83% have relationships to 2 to 10 suppliers with an average of 6.7 suppliers.

Most ISAs make use of some sort of IT-system. Most use email and an office package but only 56% use IS support in contact management, 31% in financial accounting and 15% make use of an ERP system. Only one agency uses a web shop. In focus groups we could not identify significant attempts for the integration of those systems. Additionally, 47% of the participants responded that they have very poor to poor IT knowledge, even though most of them administrate their IT themselves.

The most important process executed by ISAs is the sales process. It usually starts with accessing information of previous encounters with the prospective customer. This includes previous orders as well as visit reports. During the sales visit the sales agent presents the services or products of their principals. After the sales visit the ISA generates documents that may be a request for quotation or a visit report. The second is a document that is legally demanded in Germany as a proof of activity.

Additionally, a sales agent represents the principal in front of its customers. S/He communicates information on the status of previous orders to the customer. This may include shipped but not paid orders as well as paid but not shipped orders. Both are vital information for the sales agent demanding contrary actions. Not surprisingly, ISAs spent substantial time in the process of retrieving and aggregating information.

The main weaknesses of their current sales processes are (Kett, Kokemüller, Höß, Engelbach and Weisbecker 2008, Kett, Höß and Kokemüller 2008):

- High manual effort for the sales agents in maintaining information,
- lack of up-to-date information and
- problems of not having the required information at the right place.

To improve the initial situation it becomes crucial to (better) integrate and support the sales processes of ISAs and their suppliers. Therefore, a mobile multi-supplier sales information platform (Figure 1) is designed and developed which provides:

- back office and mobile sales cockpits which are user interfaces for the sales agents to access the stored sales information of the sales information system with mobile and back office IT-devices and IT-systems, and
- the system integration of suppliers and ISAs that assures the provision of up-to-date sales information.

The latter is the focus of this paper. Which we reformulate as:

Integration using a central service provider: Both ISAs and their suppliers connect their EIS' to a central hosted service. This service provider negotiates the integration and exchange of data and provides the above mentioned platform for the sales agencies. In this inter-organizational integration a data hub is present. It integrates data from all parties into its integrated database.

Analyzing this scenario we observe a cost-benefit asymmetry: The major benefit of the platform is received by the ISAs while the major investments – especially for the integration of multiple EIS – are

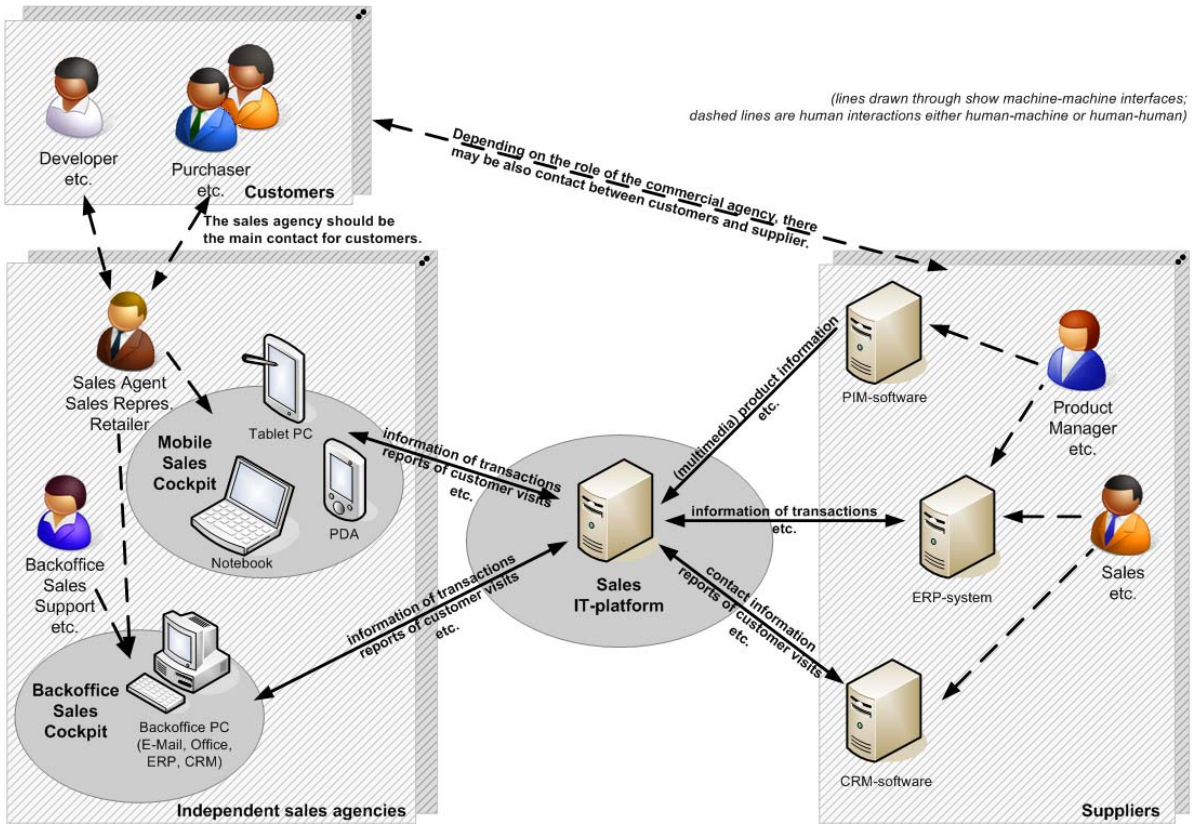

Figure 1: Overview of key players, systems, and interfaces (Kett et al. 2008)

to be invested by the suppliers. We assume that the integrating artifact will be accepted best if all participating partners achieve a substantial benefit from it. Consequently, we augment the business case with two additional scenarios to create an overall Win-Win situation both addressing integration and data quality needs. We choose these particular two scenarios as we often observe them in practice as being drivers for integration projects.

Integration without a central service provider: This scenario is usually found in Supply Chain Management where product catalogs are published along the value-chain to principals who in turn create transactional and store inventory data. Especially for highly integrated manufacturing processes as just-in-time or just-in-sequence (Frazier, Spekman and O'Neal 1988) processes, high data quality is of vital importance. In this scenario usually no single participant of the primary value chain is able or interested in providing a central service for high data quality.

Internal integration of master data: This is often referred to as Master-Data-Management (MDM). Its purpose is to create a consolidated view on master data and to achieve and maintain a high standard of data quality. According to Kokemüller and Weisbecker (2009) master data builds the foundation of inventory and transactional data. Its quality has therefore direct influence on the quality of inventory and transactional data. Additionally, we assume that by using references between these three classes we will be able to create a higher data quality in all of them (Chen, Kalashnikov and Mehrotra 2005).

4 REQUIREMENTS

We now derive requirements from the business scenarios. Following Pohl (2008) we classify them into functional (*FR*) and quality (*QR*) requirements. We further use constraints to incorporate domain specific aspects. In the following subsection we formulate the requirements, before we validate them in the second subsection.

4.1 Formulation of requirements

The major functional goal of an integration architecture is that it facilitates access to information. We explicitly require therefore one general functional requirement:

FR I Information of an information source has to be made available to an information sink.

This general requirement has to be met by every integration architecture. Based on the empirical analysis given in (Kokemüller et al. 2008) we identified several constraints to this general requirement: (i) ISAs are *independent autonomous enterprises* and (ii) they are mostly micro-sized enterprises with *limited financial resources*. (iii) Sales agents posses low to *very low IT-knowledge*. (iv) The information and the integrated EIS are *mission critical*. Finally, (v) Chandra, Dahlin, Gao and Nayate (2001) report that the *network infrastructure is not sufficiently reliable*. To every constraint we will now derive special requirements. Several requirements are influenced by more than one constraint. We discuss those under the most influential constraint.

4.1.1 Constraint: Independent autonomous enterprises

In his seminal work Conway (1968) defined the rule, that the design of an infrastructure follows the organisational design. We therefore require that the autonomy is preserved:

FR II The autonomy of the involved business parties regarding their internal representation of information must be preserved.

Every enterprise needs to access and modify its data. Therefore, we require:

FR III The integrated information has to be readable and writable by every EIS autonomously

4.1.2 Constraint: Limited financial resources

The constraint is especially important as we know from Merritt and Newell (2001) that about one supplier is added and one is dropped per year. Implying, that every year the EIS of a new supplier need to get integrated and that the investments of another supplier in the integration get potentially worthless. Usually, those EIS are legacy systems where the main effort in the integration is invested into their alteration. We require:

FR IV The integration of a system has to be possible with minimal or best none alteration of that system.

This constraint needs additionally to be addressed by high reusability as discussed by Schwinn and Winter (2005). We formulate a quality requirement:

QR I As much components of the architecture as possible have to be designed reusable.

Likewise, as a major cost factor in information integration is the creation of data mappings, we formulate:

FR V The Architecture needs to provide functionalities to enable easy reusability of transformations.

Chari and Seshadri (2004) describe, that information costs may be reduced by standardization. While the integrated IS needs to stay autonomous, we require for the internal architecture:

FR VI The syntactic interoperability needs to be standardized.

4.1.3 Constraint: Very low IT-knowledge

This constraint demands, that the integration may be administrated by an external service provider. Therefore, we require:

FR VII The administrative user interface has to be executable remotely.

4.1.4 Constraint: Mission critical EIS

This constraint poses implications regarding the integrity, accountability and availability of the integration service. We require that the effects of the integration service are not altered as this would have severe influences on the integrity of the data:

FR VIII The architecture must provide means to detect violations on the integrity of the integration service.

In the case of write operations triggered from outside of the EIS it has to be assured, that only trusted entities may submit write operations. Therefore, we require:

FR IX Access to the service platform should only be granted to clients that have been securely identified and authenticated.

If the architecture is used to integrate legally binding transactions or information that forms the basis of such, then we require:

FR X The architecture should provide means to ensure that the integrated information cannot be reputed.

Low data quality may lead to misleading, mistrusted and outdated information. We consider high data quality of vital importance to the business outcome and require therefore:

FR XI The architecture has to provide means to establish and maintain high data quality.

Furthermore, leads this constraint to several quality requirements. We require that the integrated EIS must stay available independent of the integration architecture:

QR II The integration architecture should not interfere with the availability of the EIS.

As the architecture is used to integrate sensible business data, confidentiality of the transferred data should be preserved at all times (Ghosh and Swaminatha 2001). Therefore, we require:

QR III The confidentiality of the transferred data should be preserved at all times.

Availability of the integrated information is of major importance. In order for the service to stay available the architecture must be able to grow together with its user base. Consequently, we require:

QR IV The integration architecture has to be scalable. The complexity should increase only approx. linearly.

In the integration of productive EIS the owning party depends on the availability of the integrated EIS. As the access to the data may result in unpredictable load we require:

QR V The load produced by the integration must be predictable by the business party running the particular IS artifact.

At the same time ISAs depend on the provided service. We formulate from the opposing perspective:

QR VI The response time on the integrated data has to be always fast.

4.1.5 Constraint: Network infrastructure is not sufficiently reliable

The infrastructure component most important to data integration in distributed scenarios is the connection. We can observe that the service level of broadband internet connections, especially in the low cost segment, is not sufficient for applications that always need access to data (Chandra, Dahlin, Gao and Nayate 2001). We formulate:

QR VII The service has to be available also under the circumstances of low profile connections.

4.2 Verification of requirements

To verify the completeness of the formulated requirements, we check whether all security goals are met. This is a reasonable measure, as the primary requirement for a mission critical integration is, to provide secure access to information. For our analysis we used the confidentiality, integrity, and availability (CIA) triangle that forms the fundamental basis of IT security (Swanson 2001, Kesh and Ratnasingam 2007). We also added the security goal of accountability (Pfitzmann 2001) to our analysis. While this approach cannot provide the same confidence as an empirical verification, it may still serve as a reasonable indicator.

Analyzing the security goals as shown in Table 1, every security goal is addressed by at least one requirement. Therefore, we are confident that the formulated requirements are reasonably complete.

Security Goal	**Addressed by requirements**
Confidentiality	QR III
Integrity	FR VIII, FR IX
Availability	QR II, QR IV, QR V, QR VI, QR VII
Accountability	FR X

Table 1: Verification of requirements

5 ARCHITECTURE

We will now describe the architecture of VIANA and show how we derive it from the above presented requirements. Two paradigms for information integration (FR I) can be observed: virtual and materialized integration. Most research prototypes of Peer Data Management Systems (PDMS) integrate data virtually, that is while querying it. A plan is generated that queries data locally and distributes locally unanswerable parts of the query to nodes it knows or assumes to be able to provide an answer. These nodes answer the parts to their abilities and forward the remaining parts to other nodes. In generating the query plan and integrating the results exact knowledge of the data model and schemas of the queried nodes is necessary. Virtual integration is not limited to PDMS, a prominent architecture for virtual data integration is given by Wiederhold (1992) and was refined by Roth and Schwarz (1997). An example for their usage is the integration of web sources by meta search engines. In comparison, materialized integration replicates information from the sources into an integrated database. Read queries are then executed locally. Prominent examples of materialized integration are Data Warehouses.

The process of integrating new sources into an existing scenario differs substantially from virtual to materialized integration. Correspondences in the form of mapped schemas are necessary in both variants. Yet, maintenance of materialized views needs distribution of changes, while virtual integration distributes read operations. We see one major disadvantage in the process of integrating new sources that virtual integration exposes: The information consuming EIS has to integrate new sources into its query plan. If the system's query plan is not alterable this is a severe barrier. In comparison, materialized integration needs to execute write operations on the EIS' information source.

These write operations can be handled by an IS artifact. By building this IS artifact during the implementation of the integration the integration architecture is under full control of the implementer. Therefore, it is adaptable to a variety of architectural variants and may not need any alteration in the information consuming EIS (FR IV). Additionally, the data quality of virtually integrated sources can only by improved on the basis of the query result which might be incomplete (FR XI).

Optimistic replication as described by Saito and Shapiro (2005) is especially suitable to comply with the presented requirements. It refers to a concept of materializing integration where conflicts are handled optimistically. If according to Demers, Greene, Hauser, Irish, Larson, Shenker, Sturgis, Swinehart and Terry (1987) conflicts are expected to happen rather infrequently, it is better to wait for it and react then to prevent conflicts. This approach is therefore referred to as being optimistic. Preventing conflicts as in pessimistic replication relies according to Bernstein and Goodman (1983) on blocking mechanisms which reduce the scalability of the integration architecture (QR IV). Furthermore, blocking not only blocks the integration but needs to block the integrated IS. It influences therefore its autonomy and availability (QR II).

5.1 System outline

The Architecture of VIANA is build for optimistic replication of master and operational data. It propagates write operations and incorporates mechanisms to enhance data quality. The key concepts are shown in Figure 2. The communication and integration is established by peers. A peer wraps an information source and transports changes in the data along configured paths. We emphasize that those paths are not used for querying data as in a virtual integration but for the propagation of write operations. Read operations are executed exclusively locally (FR III). This makes a deterministic response time feasible (QR VI). In addition, only local users and locally relevant data interfere with the local system (QR V).

We only demand from a peer, that it provides the necessary functionality to cooperate with other peers. Like in the Wrapper/Mediator Architecture (Roth and Schwarz 1997), the functionality a peer provides towards the integrated information source depends on the capabilities of that source. We require that the peer, independent of the implementation details, always initiates an operation in VIANA when a local atomic transaction is executed. We understand that this is a challenge if the corresponding EIS cannot trigger events on atomic write operations. In this case periodic checks for changed data could be implemented. We emphasize that a sufficient high frequency is needed to lower the chance of conflicting write operations (Wang, Reiher, Bagrodia and Kuenning 2002) which would demand user interaction. We now discuss several types. The numbers in parenthesis are references to Figure 2.

Wrapper to a database: (1a) This kind of interaction extends common databases with PDMS facilities. We show the integration of the database using publish/subscribe interaction. That is, because we want every write transaction in the database to be published immediately to neighboring peers. A standard conformant way to achieve this in relational databases would be by using SQL/Trigger. More efficiently, it may be implemented using vendor dependent transaction logs. Interaction directly with databases is not limited to stand-alone databases but includes the databases of EIS (1b). While this may impose some difficulties in applying business logic to data it may at times be the only way to integrate legacy systems (FR IV).

Wrapper to an EIS: (2) Many modern EIS provide a way in which external applications can monitor their data and provide write access by some kind of interface. This type of interaction is similar to integrating databases directly. Yet, it eliminates the need to care about business logic. Thus, it is our preferred way of integrating EIS.

Hybrid Wrapper for observation and access: If an information system does not provide the ability to observe its data for changes but merely provides the functionality to read and write data by its

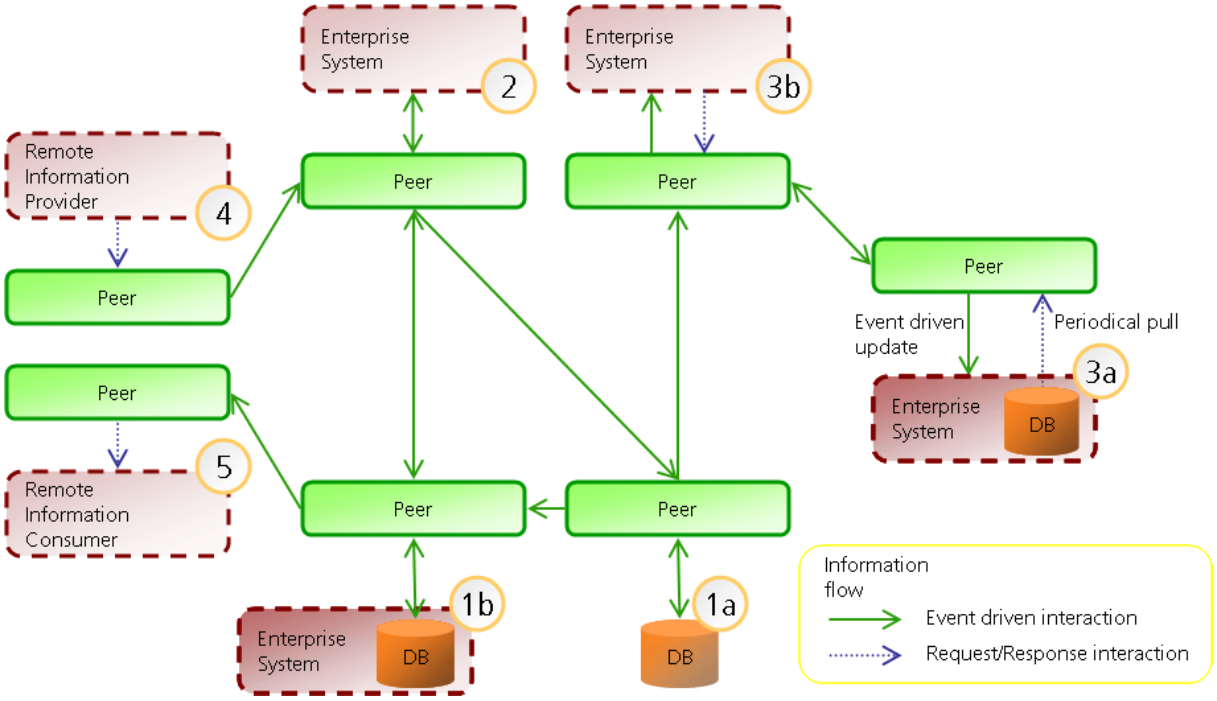

Figure 2: System overview depicting interaction patterns and information flow

interfaces, then the observation task can be done directly on the database (3a) or periodically checking the system for updates (3b). The data access remains using techniques of the information system.

Interface for a remote information provider: (4) In this scenario a peer publishes an interface and is invoked by a remote EIS. The information flow is unidirectional from the remote system to the peer. As a consequence, the peer only provides outgoing connections to other peers.

Interface to a remote information consumer: (5) Here, the information flow is unidirectional from the peer to the remote system. As a consequence, the peer only provides incoming connections from other peers.

The security of the architecture can easily be enhanced by using Secure Socket Layer (SSL) or Transport Layer Security (TLS). This allows us to comply with QR III and FR VIII. In combination with server certificates for signatures and encryption FR IX and FR X can be satisfied.

We understand the network that is built up by several peers as a directed graph. Every edge is a function of the source, target and the transformation between two of their exported schemas. Exported schemas are different to the internal conceptual schema to maintain there autonomy (FR II). The transformations between different peers are mappings of there exported schemas. We stress, that those schemas should resemble standards for data exchange. This minimizes the pair wise integration effort as the mappings may be reused (FR V). Those mappings can be defined in whatever language appropriate. We make use of existing XML Technologies as XSLT (W3C 1999) or XQuery (W3C 2007).

A peer orchestrates its interaction with neighboring peers. To minimize efforts, the syntactical interaction is standardized (FR VI). Moreover, only the components that implement the direct interaction with the integrated EIS must be implemented for every integration scenario. Components that provide cross-cutting functionalities such as the interaction, data quality algorithms or optimistic replication are designed to be reusable (QR I).

5.2 User Interaction

VIANA supports several interfaces for user interaction. Those divide into an eclipse editor for administrative purposes and functionalities for end-user interaction. The second can potentially be integrated with an EIS. The communication between all user interfaces and peers uses Web-Services facilitating alternative interfaces for different user groups or integration scenarios. It also enables a remote administration of the architecture (FR VII).

The administrative interface has several functionalities: The core view shows the topology of a network of peers. In this diagram (Figure 3) the integration of EIS is modeled as communication between exported views of peers. To every exported view exactly one peer is connected which contains its conceptual view. All edges represent transformations either between exported views of different peers or between an exported view and a conceptual view of one peer. An important aspect of this diagram is that to every peer its URI for further communication can be configured. This defines the endpoints for the communication of the administrative interface with the peers. This is especially important as we do not possess a single entry point to the architecture, due to the inherently distributed nature of VIANA. Another diagram we implemented is to edit schemas in UML notation. In this context, the possibility of annotating a schema with domain knowledge becomes important. This helps to create a common understanding on the intention of certain attributes and can help to create high data quality on an organizational level (Madnick and Zhu 2006). Currently, we are extending the administrative interface to gather information from peers on their run-time behavior.

We believe that interaction with the end-user is best accomplished in the context of the EIS. As a general extension to web-based systems, we plan to create portlets for two tasks: (i) The user should be notified, if the system has identified data quality concerns for the currently viewed object that could not be resolved automatically. This DQ-Servlet allows resolving conflicts in data quality. (ii) Likewise, the user should be asked for interaction if an update sequence failed. Those interfaces may be plugged in neatly in the EIS but may also be instantiated as standalone applications (FR IV).

6 DISCUSSION AND LIMITATIONS

The major contribution of VIANA compared to existing integration approaches is, that it does not depend on a central service, being a bus or a hub. Yet, it still provides the abstraction that is aimed to by implementing such a central service and does not loose itself in Spaghetti integration. By using peers to integrate EIS into a P2P architecture, the EIS stay autonomous at runtime and additionally the alteration of one EIS has only minor effects on the overall deployment (QR II, QR IV).

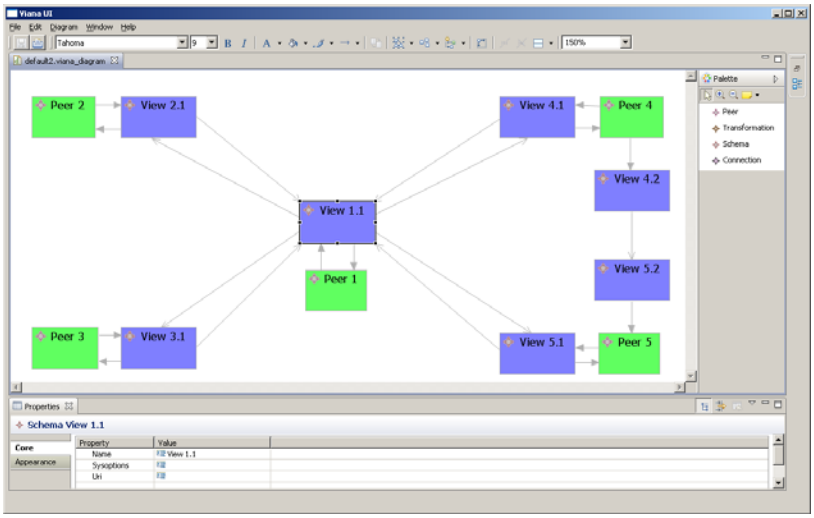

Figure 3: Eclipse editor for topology modelling

Materializing integration has some characteristics that need special discussion. Due to the fact that data is replicated between several systems objects are stored simultaneously in several places. Those should not enforce write locks. Therefore, it is essential that strategies are developed to cope with concurrent updates on one distributed object. As

these are very technical we did not discuss them here.

Additionally, diverging representations of one real-world object may exist. This is generally a data quality concern that has several dimensions (Wang and Strong 1996). One reason that may lower data quality is due to outdated information, which in comparison to virtual integration is possible, as not everywhere the same representation is accessed. VIANA does not claim to solve all possible drawbacks in data quality – including the above. Yet, it provides mechanisms to address several severe data quality concerns involving reference reconciliation that are not as easy solved in virtual integration. Moreover, it incorporates common mechanism based on high quality schema mappings.

Materializing integration requests that every replica is stored separately, which yields to an increased demand for storage. Nevertheless, while it has costs associated it brings the advantage of not depending on external systems at runtime. Undoubtedly, the data quality will lower over time when the connection and therefore the integration of external systems is interrupted. Still, the data stays accessible (QR VI, QR VII). Another characteristic is that data moves physically out of the realm of its owner. VIANA could address this trust issue by providing a data purge mechanism. We plan this as future work.

While the amount of interconnections in the graph is potentially of the order $n(n-1)$ we expect it to follow a power law as known from network theory (Barabási and Albert 1999). From the business perspective this seems reasonable, as not every business partner maintains direct relations with every other possible partner. Especially, if one partner is added to the infrastructure this is probably because a relationship with *one* of the already existent partners is formed. Furthermore, VIANA does not depend on any centralized component that would form a central point of failure or is used by every transaction. Following Wang et al. (2002) we therefore believe that VIANA scales well (QR IV). Certainly, we will have to show this in an experiment.

By using multiple databases and related peers as shown in Figure 2, a service provider is able to sell a well defined share – consisting of storage, processing power, network traffic, etc. – of its service. Additionally, data of competing business parties can be stored physically separated. We believe that this can serve as an enabler both to a well-founded business model as well as to address trust issues.

7 CONCLUSION

We started with a presentation of the business case of independent sales agencies that have the need for EIS integration. We then augmented that business case to generate an overall Win-Win situation which left us with three independent scenarios. From the business view we derived requirements of an integration architecture. We discussed the advantages of materialized over virtual integration for that particular scenario. The architecture of VIANA uses peers as their core components to build a network for materialized data integration without the inherent need for a super node which nevertheless may be implemented in a particular instantiation. Those peers act as wrappers to Enterprise Information Systems. They amend their functionality without the necessity to alter those systems themselves. Data integration is accomplished by optimistic replication that distributes operations following semantic mapping paths.

Following the design science approach we first analyzed the problem domain in Kokemüller et al. (2008). We then presented in this paper the concept of VIANA's architecture and evaluated it against the formulated requirements, which is according to Hevner et al. (2004) a suitable descriptive method for the evaluation of an IS artifact.

The chosen approach reflects the organizational structure of independent autonomous enterprises. It does not interfere with operational activities and has the potential for a cost effective realization. We therefore believe, that VIANA addresses specific integration needs better, than existing approaches.

References

Barabási, A. and Albert, R. (1999) Emergence of Scaling in Random Networks, Science, 286 (5439), 509-512.

Bernstein, P. A. and Goodman, N. (1983) The failure and recovery problem for replicated databases, *PODC '83: Proceedings of the second annual ACM symposium on Principles of distributed computing*, New York, NY, USA, 114-122.

Bernstein, P. A. and Haas, L. M. (2008) Information integration in the enterprise, Commun. ACM, 51 (9), 72-79.

Chandra, B., Dahlin, M., Gao, L. and Nayate, A. (2001) End-to-end WAN service availability, *USITS'01: Proceedings of the 3rd conference on USENIX Symposium on Internet Technologies and Systems*, Berkeley, CA, USA, 9-9.

Chari, K. and Seshadri, S. (2004) Demystifying integration, Commun. ACM, 47 (7), 58-63.

Chen, Z., Kalashnikov, D. V. and Mehrotra, S. (2005) Exploiting relationships for object consolidation, *IQIS '05: Proceedings of the 2nd international workshop on Information quality in information systems*, New York, NY, USA, 47-58.

Conway, M. E. (1968) How do committees invent? Datamation, 14 (4), 28-31.

Demers, A., Greene, D., Hauser, C., Irish, W., Larson, J., Shenker, S., Sturgis, H., Swinehart, D. and Terry, D. (1987) Epidemic algorithms for replicated database maintenance, *PODC '87: Proceedings of the sixth annual ACM Symposium on Principles of distributed computing*, New York, NY, USA, 1-12.

Dolmetsch, R. (2000) eProcurement, Addison-Wesley, München.

Elmagarmid, A. K., Ipeirotis, P. G. and Verykios, V. S. (2007) Duplicate Record Detection: A Survey, IEEE Transactions on Knowledge and Data Engineering, 19 (1), 1-16.

Erasala, N., Yen, D. C. and Rajkumar, T. M. (2003) Enterprise Application Integration in the electronic commerce world, Computer Standards und Interfaces, 25 (2), 69 - 82.

Frazier, G. L., Spekman, R. E. and O'Neal, C. R. (1988) Just-in-time exchange relationships in industrial markets, The Journal of Marketing, 52-67.

Ghosh, A. K. and Swaminatha, T. M. (2001) Software security and privacy risks in mobile e-commerce, Commun. ACM, 44 (2), 51-57.

Haas, L. (2006) Beauty and the Beast: The Theory and Practice of Information Integration, Database Theory – ICDT 2007, 28-43.

Halevy, A. Y., Ives, Z. G., Madhavan, J., Mork, P., Suciu, D. and Tatarinov, I. (2004) The Piazza Peer Data Management System, IEEE Transactions on Knowledge and Data Engineering, 16 (7), 787-798.

Halevy, A., Rajaraman, A. and Ordille, J. (2006) Data integration: the teenage years, *Proceedings of the 32nd VLDB Conference*, 9-16.

Hasselbring, W. (2000) Information system integration, Commun. ACM, 43 (6), 32-38.

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

Huebsch, R., Chun, B., Hellerstein, J. M., Loo, B. T., Maniatis, P., Roscoe, T., Shenker, S., Stoica, I. and Yumerefendi, A. R. (2005) The Architecture of PIER: an Internet-Scale Query Processor, *Conference on Innovative Data Systems Research*, Jan. 2005.

Ives, Z., Khandelwal, N., Kapur, A. and Cakir, M. (2005) Orchestra: Rapid, collaborative sharing of dynamic data, *Conference on Innovative Data Systems Research*, Jan. 2005.

Kesh, S. and Ratnasingam, P. (2007) A knowledge architecture for IT security, Commun. ACM, 50 (7), 103-108.

Kett, H., Höß, O. and Kokemüller, J. (2008) Mobile Multilieferanten-Vertriebsinformationssysteme für Handelsvertretungen und -vermittlungen, Fraunhofer IRB, Stuttgart.

Kett, H., Kokemüller, J., Höß, O., Engelbach, W. and Weisbecker, A. (2008) A Mobile Multi-Supplier Sales Information System for Micro-sized Commercial Agenciesin P. Cunningham and M. Cunningham (Eds.) , *Collaboration and the Knowledge Economy*, 1240 - 1247.

Kokemüller, J., Kett, H., Höß, O. and Weisbecker, A. (2008) A Mobile Support System for Collaborative Multi-Vendor Sales Processes, *Proceedings of the Fourteenth Americas Conference on Information Systems*, August 14th-17th, Toronto, ON, Canada.

Kokemüller, J. and Weisbecker, A. (2009) Master Data Management: Products and Research, *Fourteenth International Conference on Information Quality*, November 7-8, 2009, Potsdam, 8-18.

Madnick, S. and Zhu, H. (2006) Improving data quality through effective use of data semantics, Data & Knowledge Engineering, 59, 460-475.

Merritt, N. J. and Newell, S. J. (2001) The Extent and Formality of Sales Agency Evaluations of Principals, Industrial Marketing Management, 30 (1), 37-49.

Ng, W. S., Ooi, B. C. and Tan, K. L. (2003) PeerDB: a P2P-based system for distributed data sharing, *Proceedings of the 19th International Conference on Data Engineering*, 633-644.

Nunamaker Jr, J. F., Chen, M. and Purdin, T. D. M. (1991) Systems development in information systems research, Journal of Management Information Systems, 7 (3), 89-106.

Pfitzmann, A. (2001) Multilateral security: Enabling technologies and their evaluation, Lecture Notes in Computer Science, 50-62.

Pohl, K. (2008) Requirements Engineering, dpunkt-Verlag, Heidelberg.

Puschmann, T. and Alt, R. (2004) Enterprise application integration systems and architecture - the case of the Robert Bosch Group, Journal of Enterprise Information Management, 17 (2), 105-116.

Rahm, E. and Do, H. H. (2000) Data Cleaning: Problems and Current Approaches, IEEE Data Engineering Bulletin, 23 (4), 3-13.

Roth, M. T. and Schwarz, P. (1997) Don't Scrap It, Wrap It! A Wrapper Architecture for Legacy Data Sources, *Proceedings of the 23rd VLDB Conference*, Athens, Greece, 266-275.

Saito, Y. and Shapiro, M. (2005) Optimistic replication, ACM Comput. Surv., 37 (1), 42-81.

Schwinn, A. and Schelp, J. (2005) Design patterns for data integration, Journal of Enterprise Information Management, 18 (4), 471-482.

Schwinn, A. and Winter, R. (2005) Entwicklung von Zielen und Messgrößen zur Steuerung der Applikationsintegration,.

Swanson, M. (2001) Security self-assessment guide for information technology systems, Special Publication, 800-26.

W3C (1999) XSL Transformations (XSLT) Version 1.0, http://www.w3.org/TR/xslt.

W3C (2007) XQuery 1.0: An XML Query Language, http://www.w3.org/TR/xquery/.

Walter, P., Werth, D. and Loos, P. (2006) Peer-to-Peer-Based Model-Management for Cross-Organizational Business Processes, June, Los Alamitos, CA, USA, 255-260.

Wang, A., Reiher, P. L., Bagrodia, R. and Kuenning, G. H. (2002) Understanding the Behavior of the Conflict-Rate Metric in Optimistic Peer Replication, *DEXA '02: Proceedings of the 13th International Workshop on Database and Expert Systems Applications*, Washington, DC, USA, 757-764.

Wang, R. Y. and Strong, D. M. (1996) Beyond accuracy: what data quality means to data consumers, Journal of Management Information Systems, 12 (4), 5-33.

Wiederhold, G. (1992) Mediators in the Architecture of Future Information Systems, IEEE Computer, 25 (3), 38-49.