

A Coupling Metric for B2B e-Commerce Systems

Anthony M. Orme

University of Alabama in Huntsville
Huntsville, Alabama 35899
United States of America
256.824.6244
aorme@cs.uah.edu

Letha Etzkorn

University of Alabama in Huntsville
Huntsville, Alabama 35899
United States of America
256.824.6291
aorme@cs.uah.edu

ABSTRACT

Recent research on distributed Internet based B2B e-commerce has focused on long running distributed transactions and the possibility that this characteristic may lead to extended record locking. New standards developed define the terms and condition necessary for businesses to interact in an electronic contract, such as a Trading Partner Agreement (TPA) and therefore do not require record locking. Researchers suggest that B2B e-commerce systems developed using TPAs show a lower degree of coupling between each partner's systems. Although the proceeding statement seems intuitively true, currently there are no metrics that have been shown to theoretically or empirically prove this relationship. The focus of this paper is to provide a set of metrics to measure coupling in B2B architectures.

Keywords

B2B, e-commerce, Trading Partner Agreements, Coupling

1.0 INTRODUCTION

The Internet boom of the last decade has led to the realization that business automation is possible on a wider scale than ever before. The result of early B2B automation was the development of several proprietary B2B frameworks that defined a set of services to provide pair wise interoperability between two companies for integration of their enterprise applications [2]. However, the pair wise interoperability was limited by the development of many competing frameworks and a lack of interoperability between the competing frameworks.

There are many examples of B2B e-commerce transactions that require coordination of

multiple interactions between many trading partners. These B2B e-commerce transactions require distributed transaction support not provided in these B2B frameworks. Therefore, front-end applications provide the distributed transaction support required for many B2B and logistics management problems. If these front-end applications use traditional distributed transaction techniques the possibility of extended record locking across organizational boundaries exists. In long running transactions these extended record locks may not be held by an integrated application but rather by a partner outside of this organization, and therefore these systems exhibit a higher degree of coupling. Thus the autonomy of individual trading partners may be compromised [8], which may lead to a potential loss of sales under certain circumstances.

Some have suggested easing the restrictions of the ACID properties of standard transaction-based systems as a solution to the problem of extended record locking [8]. The relaxation of the ACID properties is achieved by using a TPA to ensure complete independence for both parties and also by reducing the amount of system coupling. However, relaxing the ACID properties can lead to several well known data anomalies, thus retaining the ACID properties is desirable. The main problem with retaining the ACID properties is the extended record locking and potential rollback of work that has been accomplished in a transaction when one system fails, when in some cases preserving the work would be preferable [6,7].

The remainder of this paper is organized as follows: Section 2 provides the basic information required to understand software coupling metrics as applied to software modules and background information on TPA as . In Section 3 we outline a coupling metrics that may be used to measure coupling in B2B architectures. In Section 4 we provide a conclusion to our research and topics for further investigation.

2.0 BACKGROUND

The background section is divided into the following subsections: Section 2.1 explores current coupling metrics that provide a measure for software modules of a given system. Section 2.2 provides detailed background information on TPAs and their role and relationship to other components in a B2B transaction.

2.1 Software Coupling Metrics

One measure of a modular software design is coupling between modules in the system. Coupling measures the “connectedness” of a module to other modules, global data, and the outside environment. The goal is to have modules that exhibit low coupling so that we minimize the reliance from one module on another. Page-Jones [ref] provides 3 reasons why low coupling is desirable between modules:

1. Fewer interconnections between modules reduce the chance that a fault in one module will cause a failure in other modules
2. Fewer interconnections between modules reduce the chance that changes in one module will cause problems in other modules (which enhances reusability)
3. Fewer interconnections between modules reduce programmer time in understanding the details of the other modules.

2.2 Trading Partner Agreements

In a B2B e-commerce transaction contractual agreement there is need to agree not only in the traditional contractual sense, but e-commerce also requires agreement from the perspective of the information technology used to carry out the agreement. An electronic TPA will facilitate the electronic negotiation by accurately embodying terms and conditions agreed upon by each trading partner. A TPA allows a transaction to take place without requiring or providing a way to enforce the ACID properties of traditional transaction-based systems.

Figure 1 shows the basic structural elements of a TPA. The TPA information section allows a TPA to have a specific name, provides generic role information and specifies the names of the

organizations and contact information of the trading partners. The Communication Properties section allows for the specification of communication protocols (e.g. VAN-EDI, HTTP). The security properties section allows the specification of authentication protocols and certificate information. The action section allows a server to specify a set of interaction requests that can be made by the other partner. The Sequencing Rules section specifies the order actions that can be called on a server. Finally, the Error Handling section specifies conditions relating to errors that may occur.

TPA Information
Communication Properties
Security Properties
Actions
Sequencing Rules
Error Handling

Figure 1 Structural Elements of

An electronic TPA is an XML-based document. [ref] defines the XML-based layered structure of a TPA in the following way.

```
<TPA>
<TPAInfo>
...<!--TPAname, role definitions,participants, etc.-->
</TPAInfo>
<Transport>
<!--communication & transport security info-->
</Transport>
<DocExchange>
<Security>
...<!--document-exchange and message security info-->
</Security>
<BusinessProtocol>
<ServiceInterface> <!-- for each provider-->
...<!--Action definitions etc.-->
</ServiceInterface>
</BusinessProtocol>
</DocExchange>
</TPA>
```

[ref] also defines the following the three layers of an electronic TPA in similar fashion to the layers of the OSI model. These three layers allow trading partners

to define in a static manner the ways in which they interact, the manner in which they exchange documents and the transports available for exchanging the documents.

- **Business-Protocol Layer:** The Business-Protocol layer defines the heart of the business agreement between the trading partners: the services (actions) which parties to the TPA can request of each other and the sequencing rules that determine the order of requests. The Business-Protocol layer is the interface between the TPA-defined actions and the business-application functions that actually perform the actions.
- **Document-Exchange layer:** The Document Exchange layer accepts a business document from the Business Protocol layer, optionally encrypts it, optionally adds a digital signature for nonrepudiation, and passes it to the transport layer for transmission to the other party.
- **Transport layer:** The transport layer is responsible for message delivery using the selected communication protocol. Transport security (encryption and authentication) definitions are also provided.

3.0 COUPLING DEFINITIONS FOR B2B

Page-Jones reasoning about desiring low-coupling between modules may be modified to work at a system level in the following way:

1. Fewer interconnections between *systems* reduce the chance that a fault in one *system* will cause a failure in other *systems*.
2. Fewer interconnections between *systems* reduce the chance that changes in one *system* will cause problems in other *systems*.
3. Fewer interconnections between *systems* reduce programmer time in understanding the details of the other *systems*.

Using the modified Page-Jones definitions we are able to define metrics that measure the coupling in B2B systems by the number of interconnections that exists between business partners. Specifically, we can measure coupling based upon TPA, which contains the

interconnection information discussed in the sections defined above.

3.1 B2B Coupling Metric

The TPA provides several options for counting interconnections between systems in the Business-Protocol Layer, Document-Exchange layer or the Transport Layer. One example of a metric taken from the Business-Protocol Layer would be Number Of Actions (NOA). NOA counts the number of actions defined, which describes the requests each party can issue to the other. An action is analogous to making a remote procedure call to the other parties' system, which is an interconnection.

NOA can be represented mathematically, if A represent distinct action in a TPA, as follows:

$$NOA(tpa) = \sum A_j \text{ for all } 1 \leq j \leq n \text{ (number service interfaces in the tpa)}$$

In the following TPA, NOA(tpa) = 1 which corresponds to the action tag highlighted below. We could easily count these types of metrics with a standard XML parser.

```
<ServiceInterface InterfaceId="interface01">
  <OrgName Partyname="_LargeCo"/>
  <Client>
    <OrgName Partyname="_PensAreWe"/>
    </Client>
    <ActionMenu>
      <Action ActionId="action01" Type="basic">
        <Request>
          <RequestName>putOPOR</RequestName>
          <RequestMessage>OBIPOR</RequestMessage>
        </Request>
        <Response>
          <ResponseName>getOPO</ResponseName>
          <ResponseMessage>OBIPO</ResponseMessage>
          <ResponseServiceTime>
            <ServiceTime>3600</ServiceTime>
            <Presume>fail</Presume>
          </ResponseServiceTime>
        </Response>
      </Action>
    </ActionMenu>
    <ServerServiceTime>
      <ServiceTime>3660</ServiceTime>
      <Presume>fail</Presume>
    </ServerServiceTime>
    <StartEnabled>
      <RequestName>putOPOR</RequestName>
    </StartEnabled>
```

</ServiceInterface>

4.0 CONCLUSIONS

Our research has shown that metrics for measuring coupling in B2B systems can be achieved using standard TPAs. We calculated NOA by examining the Business-Protocol layer of TPA, which defines the interface between TPA-defined actions and the business-application functions that actually perform the actions. Other metrics could be calculated in a similar manner using the Document-Exchange Layer or the Transport Layer. We feel that further research is needed in this area to further the understanding of coupling in B2B systems. Reduced coupling would allow systems to be more autonomous, which is desirable for B2B applications.

5.0 REFERENCES

- [1] ANSI X3.135-1992 American National Standards for Information Systems- Database Language-SQL, November, 1992
- [2] Ariba Inc., "cXML Version 1.0 Specification", Aug. 2003,<http://www.cxml.org>
- [3] A. Dan, D. Dias, R. Kearny, T. Lau, T. Nguyen, F. Parr, M. Sachs, H. Shaikh, Business-to-business intergration with tpaML and a business-to-business protocol framework, IBM Systems Journal Vol. 40 No.1, pp 68-90, 2001
- [4] H. Dombrowska, The Low-Level Support and Logging for Flexible Transaction, Advances in Database and Information Systems, ADBIS, 1994
- [5] Gartner (2001) The economic downturn is no excuse to retrench B2B Efforts. <http://www.gartner.com>
- [6] B Medjahed, B Benatallah, A Bouguettaya , A. Ngu , A K Elmagarmid, "Business-to-business interactions: issues and enabling technologies", The VLDB Journal, Vol. 12, No. 1, May 2003, pp 59- 85
- [7] C. Mohan, B. Lindsay, R. Obermarck, "Transaction Management in the R* Distributed Database Management System", ACM Transactions on Database Systems, Vol. 11, No. 4, December 1986, pp. 378-396
- [8] H. Weigand, E.M. Verharen, F. Dignum, Dynamic Business Models as a Basis for Interoperable Transaction Design, Information Systems Vol. 22 No.2/3 pp.139-154 1997
- [9] H. Weigand, W..J. Heuvel, F. Dignum, Modeling Electronic Commerce Transactions A Layer Approach, Proceedings of the Third International Workshop on the Language Perspective on Communication Modeling, 1998
- [10] M. J. Wooldridge., N. R. Jennings, "Intelligent Agents: Theory and Practice", Knowledge Engineering Review, Vol. 10, Issue 2