A HUB ARCHITECTURE FOR SERVICE ECOSYSTEMS

Towards Business-to-Business Automation with an Ontology-enabled Collaboration Platform

Alex Norta

Department of Computer Science, University of Helsinki
P.O. Box 68 (Gustaf Hällströmin katu 2b), FI-00014 Helsinki, Finland

Keywords: SOC, BPM, B2B, Broker, Hub, Collaboration, Trust, Reputation, Ontology.

Abstract: The management and coordination of business process collaboration experiences changes because of globalization, specialization, and innovation. Service-oriented computing (SOC) is a means towards business-process automation and recently, many industry standards emerged to become part of the service-oriented architecture (SOA) stack. In a globalized world, organizations face new challenges for setting up and carrying out collaborations in semi-automating ecosystems for business services. A need emerges for service Hubs that not only store service offers and requests together with their issuing organizations and assigned owners, but that also allow an evaluation of trust and reputation in an anonymized electronic service marketplace. In this paper, we explore the features of a semi-automating ecosystem in which business processes are expressed as services and where Hubs are essential for bringing together service offers and requests. The presented Hub architecture is designed so that business managers benefit from an interface that borrows concepts of social-networking sites while the complex computing machinery for matching service offers and requests remains hidden from the user. The partial implementation of service-Hub components demonstrate the feasibility of our approach.

1 INTRODUCTION

In a globalized business setting, enterprizes may run complex supply chains across several tiers that comprise many geographical regions. Original equipment manufacturers (OEM) maintain intricate in-house processes of which parts are outsourced to suppliers. With the emergence of SOC (Allen et al., 2006; E.A.Marks and Bell, 2006), such business-to-business (B2B) collaboration with business processes as a service (BPaaS), may be semi-, or fully automated.

The predicted increase in BPaaS leads to a considerable communication overhead for enterprises. A system model is missing for a BPaaS-Hub so that business- and logistics managers, industrial marketers and so on, engage in setting up cross-organizational B2B collaboration in a semi-automated way. Such a BPaaS-Hub should permit laymen to engage in service matching while the required computing complexity remains hidden. This paper fills the gap and presents a BPaaS-Hub architecture.

The structure of the paper is as follows. Section 2 presents high-level characteristics of B2B collaboration from which we deduct requirements for the BPaaS-Hub architecture that Section 3 comprises. Section 4 discusses preliminary implementations and what existing systems realize the BPaaS-Hub architecture. Section 5 concludes the paper and presents future work.

2 B2B CHARACTERISTICS

Observing B2B collaborations in the EU research project CrossWork (Mehandjiev and Grefen, 2010), particular features are characteristic. An OEM organizes an in-house business process that is decomposable into different perspectives, e.g., control flow of tasks, information flow, personnel management, allocation of production resources, and so on. A complex product of an OEM comprises many components of which several need to be acquired from suppliers. The reasons for acquiring parts externally are manifold, e.g., the OEM cannot produce with the same quality, or an equally low price per piece, the production capacity is not available, required special know-how is lacking, and so on.
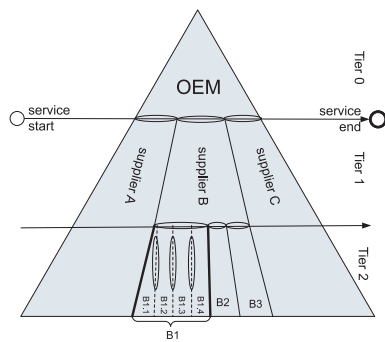
Figure 1: A collaboration pyramid in B2B.

In the scenario of Figure 1, the horizontal ellipses denote the client/server integration of ourtsourced parts of the in-house process to lower-level clients of the tier (Norta and Eshuis, 2009). The outsourced business process is refined by the supplier without the OEM's awareness while the supplier only has awareness of the OEM's outsourced process.

Vertical ellipses in Figure 1, depict a peer-to-peer (P2P) collaboration within a cluster of small and medium sized enterprises (SME). If several SMEs form a composed service in a P2P way, they become a supplier for a higher-level service consumer. For managing such P2P service collaboration (Kutvonen et al., 2007), their lifecycle needs to be managed (a) for business-community formation to compose services, (b) for the evolution of such business communities through epochs with changing members and modified services, and eventually (c) for the dissolution of P2P business communities.

For the BPaaS-Hub, we consider a stepwise "fuzzy matching" approach. The first step is a matching of service offers and requests based on extracted and ontologically clarified keywords contained in service descriptions. A matching of left over services subset requires on the next level machine-readable service-level agreements (SLA) with, e.g., WS-Agreement (Andrieux et al., 2007) or WSLA[1]. As an example for this matching type, in (Müller et al., 2009), matching templates and instantiations involve computing the adherence of the latter to templates.

For the BPaaS-Hub architecture in the sequel, we deduct requirements from the discussed B2B characteristics.

1. A BPaaS-Hub must allow laymen who have no or little SOC knowledge to engage in service discovery and matching.

2. Since the Hub is part of an anonymized service ecosystem, users must be able to check the trust-

---

[1]http://www.research.ibm.com/wsla/

worthiness and reputation of service offers and requests.

3. The Hub must support resolving ambiguities in the human-and machine readable service representations.

4. The Hub must support feasible service matching, i.e., we assume pragmatic fuzzy matching as explained above.

5. The user interaction with the Hub must be logged for extracting business intelligence.

3 A SERVICE-HUB SYSTEM

We specify a conceptual system architecture for the BPaaS-Hub. For the BPaaS-Hub architecture, we follow design principles, styles and patterns (Bengtsson, 2002; Gamma et al., 1995). Architectural styles comprise a description of component types and their topology, a description of the pattern of data and control interaction among the components, and an informal description of the benefits and drawbacks of using a particular style.

The conceptual architecture depicted in Figure 2 utilizes the principles of separation of concern, it follows a layer style, employs a pipes-and-filters pattern and pattern-based components for abstracting data repositories.

Separation of Concerns. In Figure 2, columns show the separations. *WHO*: refers to the business entities a user searches for, e.g., services in specific domains, organizations, or persons. *WITH*: refers to establishing on the fly the ontological infrastructure needed to resolve ambiguity issues in service descriptions. *WHAT*: refers to the need for pulling in additional service-related information from the Web cloud for a trust-enhancing mashups. We define a mashup as a Web page or application that combines data or functionality from two or more external sources to create a new service. *HOW*: refers to the application infrastructure necessary for the services to be matched and enacted. Additionally, social mining techniques analyze the logged user interaction with the Hub for extracting business intelligence.

Layer Style. A layer style separates vertically the BPaaS-Hub architecture with communication exchanges to the adjacent higher or lower layer to facilitate a decoupling and replacement with alternative components. In Figure 2, the top layer called *Views*, depicts all user-interface components. The middle
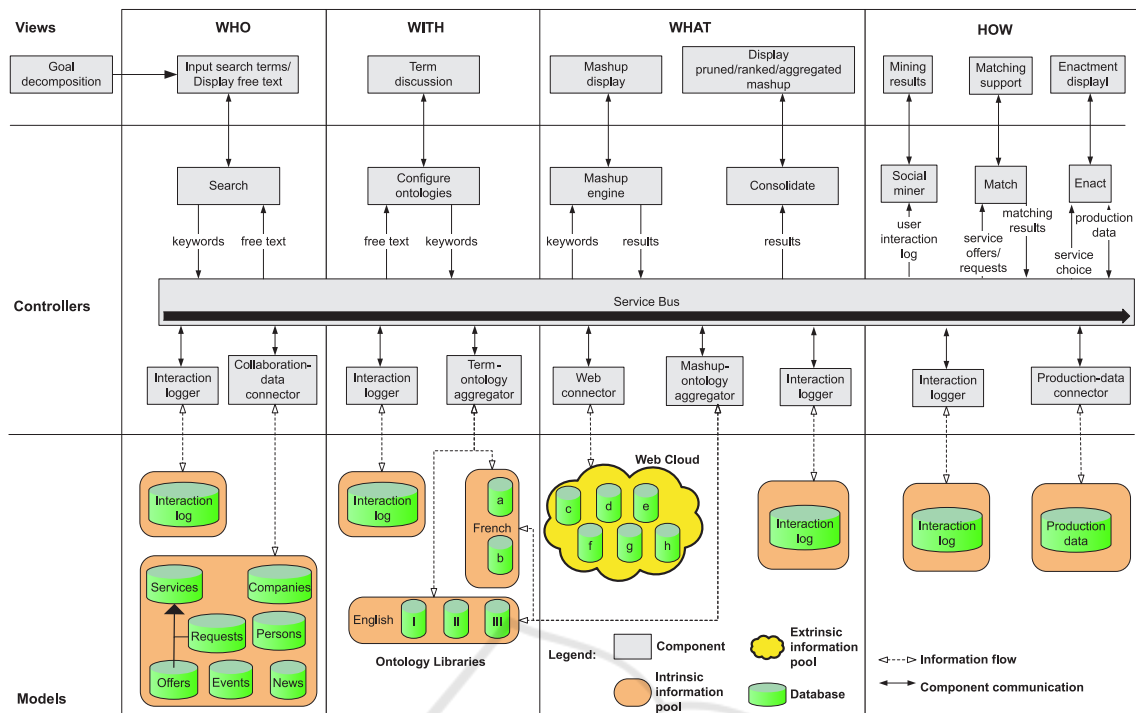
Figure 2: Architecture of a service Hub.

layer termed *Controllers*, shows components with application logics while the lowest layer termed *Models*, contains all system intrinsic or third-party extrinsic information sources from the Web cloud for trust-building mashups.

Pipes and Filters Pattern.   The components of the controller layer instantiate a pipes-and-filters pattern enforced by a service bus. The ontology-supported *Goal decomposition* delivers input for what business entities are sought after in a fully or semi-automated way. A service search results both in human-readable text and optional machine-readable WS-* specifications. An analysis of search results culminates in a dynamically linked library of ontology libraries for resolving ambiguities in the service representations. A mashup engine performs automated searches in user-selected information pools of the Web cloud. The results require consolidation, i.e., pruning, ranking and aggregated. Mining the logged user interaction generates business intelligence. A matching of service offers with service requests precedes the enactment of machine-readable WS-* service representations.

Abstract Data Repository.   On the *controller* layer of Figure 2 are components of the architectural style *abstract data repository* (Klein and Kazman, 1999). This architectural style, on the one hand, keeps the producers and consumers of shared ontologies from having knowledge of each other's existence. On the other hand, abstract data repositories hide details of shared data-repositories. The abstract interfaces to the data repositories reduce the coupling between data producers and consumers.

4   FEASIBILITY EVALUATION

In the framework of the ContentFactory[2] (CF) research project, we implement the BPaaS-Hub architecture from Section 3 for conducting case studies with industry. For satisfying Requirement 1, we implement a business-service registry termed Collab[3] that links data of service offers and requests with service-responsible persons and service-issuing organizations. For keyword extraction, Collab sends the free-text description to the Likey (Paukkeri et al., 2008) application.

For Requirement 2, a mashup component is part of the BPaaS-Hub architecture. We consider the PULS (Yangarber and Steinberger, 2009) application for populating the mashup component. Ongoing PULS extensions cater for an in-depth exploration of

---

[2]ContentFactory, funded by Tekes (Finnish Funding Agency for Technology and Innovation), http://www.verkko-ope.net/cf/

[3]http://db.cs.helsinki.fi/~tkt_coll/collab/

business acquisition, investment, nomination, product release, ownership/stake; divestment/reduction of stake.

For Requirement 3, the BPaaS-Hub architecture includes components for creating ontology libraries. We use the TermFactory[4] application for allowing terminologists to define extracted keywords that enter ontology libraries for respective Hub-application contexts. The matching component in the Hub architecture satisfies Requirement 4. Currently we implement an application for realizing the matching heuristics in (Eshuis and Norta, 2009).

For Requirement 5, the Hub architecture includes logging components for several stages of user interaction and a social mining component for the extraction of business intelligence, e.g., ProM[5]. Many options exist for the *Enact* component, e.g., ActiveBPEL[6].

5 CONCLUSIONS

In this paper we explore the characteristics of B2B collaboration and present a framework for automating the matching of service offers and service requests. Based on extracted requirements for service matching in the setting of business-to-business collaboration, we present a Hub architecture for brokering business processes as services. The Hub permits business managers to explore with free text service offers and requests, their issuing organizations and service-managing persons. Ontology engines resolve ambiguity issues in the text and to establish trust and explore the reputation of services and their affiliated organizations and persons, Hub users employ mashups comprising news feeds, blogs, wikis, and so on.

For future work, we pursue the integration of identified applications for implementing the Hub architecture and plan to conduct case studies with industry using the Hub for discovery and matching of service offers and requests. Furthermore, we explore Hub extensions for integrating a service-tendering procedure that allows users to place negotiable bids.

REFERENCES

Allen, P., Higgins, S., McRae, P., and Schlamann, H., editors (2006). *Service Orientation: Winning Strategies and Best Practices*. CAMBRIDGE UNIVERSITY PRESS.

Andrieux, A., Czajkowski, K., Dan, A., Keahey, K., Ludwig, H., Nakata, T., Pruyne, J., Rofrano, J., Tuecke, S., and Xu, M. (2007). *Web Services Agreement Specification (WS-Agreement)*. http://www.ogf.org/documents/GFD.107.pdf.

Bengtsson, P. (2002). *Architecture-Level Modifiability Analysis*. PhD thesis, Department of Software Engineering and Computer Science, Blekinge Institute of Technology, Sweden.

E.A.Marks and Bell, M., editors (2006). *Service-Oriented Architecture (SOA): A Planning and Implementation Guide for Business and Technology*. John Wiley & Sons, Inc., Hoboken, New Jersey.

Eshuis, R. and Norta, A. (2009). Constructing process views for service outsourcing. In S.Y. Shin, S. O., editor, *Symposium Proc. of the 2009 ACM Symposium on Applied Computing (SAC)*, pages 1615–1616. ACM.

Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995). *Design Patterns: Elements of Reusable Object-Oriented Software*. Professional Computing Series. Addison Wesley, Reading, MA, USA.

Klein, M. and Kazman, R. (1999). Attribute-based architectural styles. *TECHNICAL REPORT CMU/SEI-99-TR-022 ESC-TR-99-022*.

Kutvonen, L., Ruokolainen, T., and Metso, J. (2007). Interoperability middleware for federated business services in web-Pilarcos. *International Journal of Enterprise Information Systems, Special issue on Interoperability of Enterprise Systems and Applications*, 3(1):1–21.

Mehandjiev, N. and Grefen, P. (2010). Crosswork: Internet-based support for process-oriented instant virtual enterprises. *IEEE Internet Computing*. to appear.

Müller, C., Resinas, M., and Ruiz-Cortés, A. (2009). Explaining the non-compliance between templates and agreement offers in ws-agreement. In Baresi, L., Chi, C., and Suzuki, J., editors, *Service-Oriented Computing - ICSOC 2009, 7th International Conference*, volume 5900 of *Lecture Notes in Computer Science*, pages 237–252, Stockholm, Sweden. LNCS Springer.

Norta, A. and Eshuis, R. (2009). Specification and Verification of Harmonized Business-Process Collaborations - Preprint Version. Accepted for publication in the journal of Information Systems Frontier, University of Helsinki, Finland.

Paukkeri, M., Nieminen, I., Pöllä, M., and Honkela, T. (2008). Language-independent approach to keyphrase extraction and evaluation. In Scott, D. and Uszkoreit, H., editors, *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 237–252, Manchester, UK.

Yangarber, R. and Steinberger, R. (2009). Automatic epidemiological surveillance from on-line news in medisys and puls. In *Proceedings of IMED-2009: International Meeting on Emerging Diseases and Surveillance*, Vienna, Austria.

---

[4]http://www.helsinki.fi/∼lcarlson/CF/doc/TFManual.html

[5]http://prom.win.tue.nl/tools/prom/

[6]http://sourceforge.net/projects/activebpel/