

Pragmatic Interoperability in the Enterprise

— A Research Agenda

Camlon H. Asuncion

Center for Telematics and Information Technology (CTIT), University of Twente
P.O. Box 217, 7500 AE, Enschede, The Netherlands
`c.h.asuncion@utwente.nl`

Abstract. Effective collaboration among today’s enterprises is indispensable. Such collaborative synergy is important to foster the creation of innovative value-added products and services that would have otherwise been difficult to achieve if enterprises work in isolation. However, it is a widely held belief that interoperability problems have been one of the perennial hurdles in achieving such collaboration. This research aims to improve the current state of the art in enterprise interoperability research by zeroing in on the notion of pragmatic interoperability (PI). When enterprise systems collaborate by exchanging information, PI goes beyond the compatibility between the structure and the meaning of shared information, it further ensures that the intended effect of the message exchange is realized. This paper outlines our research agenda to address the analysis, design, development and evaluation of a pragmatically interoperable solution for enterprise collaboration.

1 Introduction

In today’s day and age, enterprise collaboration is a must. Collaboration allows enterprises to explore one another’s core competencies, improve services to their customers, allow efficient use of resources, and increase information sharing [5]. Continued advances in networking, computing technologies and standards have stimulated this interest on collaboration. On the one hand, these advances help explore new business opportunities to add value to their products and services. On the other hand, these advances also provide opportunities for organizations to new enable partnerships in ways that were not previously possible [16].

However, although today’s enterprises want to leverage the benefits of their collaboration, interoperability problems between their enterprise systems prevent them from doing so effectively. For example, previous investments in equipment and software cause incompatibilities between data representation and application methods making integration of new systems difficult and costly [6].

Generally, interoperability allows “two or more systems to understand one another and to use the functionality of one another” [4]. From a technical perspective, heterogeneous software systems jointly function together and provide access to their resources. From a business perspective, enterprises collaborate through the exchange information and services achieving a common business objective [4].

In recent years, research interest is growing in terms of looking into the role of pragmatic interoperability (PI) to extend the current research on syntactic and semantic inteoperability solutions. While there is a need to agree on the *structure* and the *meaning* of the shared information, the intended *effect* of the collaboration is equally important as well. PI allows participants in a collaboration to mutually affect each other's state and behavior such that the produced effect in one participant is as expected and intended by other participants. PI can potentially help realize a truly effective collaboration by bringing the benefits of technological solutions closer to the business level.

This paper outlines our research agenda for the analysis, design, development and evaluation of a PI approach for enterprise collaborations. Our notion of PI is described in Section 2. The main research objective and questions are presented in Section 3. We distinguish our work from other related approach to PI in Section 4. Finally, we conclude by briefly discussing the current research progress and immediate challenges in Section 5.

2 On the Notion of Pragmatic Interoperability

In the philosophy of language, Charles Morris introduced the term *pragmatics* to study human interpretation of (non-)linguistic signs. Pragmatics is part of a larger study on the theory of signs, called Semiotics. Semiotics is comprised of three basic components: syntactics, semantics, and pragmatics. Syntax is that which acts as a sign, semantics is that which the sign refers to, and pragmatics is the effect of the sign on the interpreter which can be realized depending on the context where the sign is used [12].

Semiotics can also be used to explain the phenomena behind the interoperability of enterprise systems. A system *interacts* with its environment by exchanging messages which are made up of signs. These interactions produce an identifiable *effect*; e.g., a system provides information to or changes a state of its environment. This effect has a *value* to both the system and the environment where the interactions are taking place. The messages consist of data and behavior properties. The *data* properties represent values that describe what the message is all about. The *behavior* properties describe the message invocations between the system and its environment. Thus, to achieve interoperability, the system and environment should share a common understanding of the *data* in the message (achieved through syntactic and semantic interoperability) and a common expectation of the *effect* of the message (achieved through PI) [13].

The PI research area is still at its infancy as evidenced by the fragmented related work [2]. Varying solutions are proposed which are mostly specific to a particular research domain leading to the proliferation of various PI definitions. Our first step has therefore been to do a systematic review of the various PI definitions from extant literature to extract key concepts so as to understand and later harmonize the definitions (*c.f.* [3] for details). This leads us to our working definition of PI as *the compatibility between the intended versus the actual use of exchanged message within a relevant shared context*.

The main concepts in this definition consists of intention, use and context. Consider a sender sending a message to a receiver for the purpose of inter-

operation: *Intention* denotes the desired possible state of the world which a message sender can achieve through collaboration with a receiver (or receivers). *Use* means how the receiver acts upon the message to realize the intention of the sender. Finally, *context* helps in the correct interpretation of the meaning of the message request so that only the relevant information or relevant actions can be used to accomplish the sender's intention. In our terms, the notion of *effect* constitute message *use* and *context*. Thus to achieve PI, the sender's intention should be compatible with how the receiver uses relevant data from the message to perform relevant action under a shared relevant context. Actions on information depend on context to achieve the intended effect.

As a simple illustration, Figure 1 shows a scenario where a hospital sends a request for a lab test to a laboratory. After processing the request, the laboratory is expected to send back a lab report thereafter. To be syntactically interoperable, both use a compatible way of structuring their message (e.g., using XML to structure the message). To be semantically interoperable, both use standards (e.g., Health Level 7 or HL7) or ontologies to annotate the syntactic structure with meaning. To be pragmatically interoperable, the laboratory should have an understanding of the context in which the request for a lab test was made so that it can realize the intention of the hospital correctly.

In Figure 1.a, the hospital intends to receive the lab report as quickly as possible as it is in an emergency context. The laboratory, on the other hand, assumes that the request is made in the usual manner and thus performs the request like any other routine requests for a lab test. However, it may be the case that, the laboratory may use information and/or perform actions that vary between emergency and normal context (e.g., prioritizing lab tests that are immediately needed, implying also that payment information be asked later for emergency context). This could mean that the report will not be returned in due time as the hospital intends. Thus, the laboratory is not able to realize the intention of the hospital as the laboratory has a different understanding of the context that the hospital is operating in. In Figure 1.b, the laboratory is able to realize the intention of the hospital as now both have the same understanding of the prevailing relevant context. In essence, pragmatics allows the meaning of the hospital's request to be specialized in the context where the request was made.

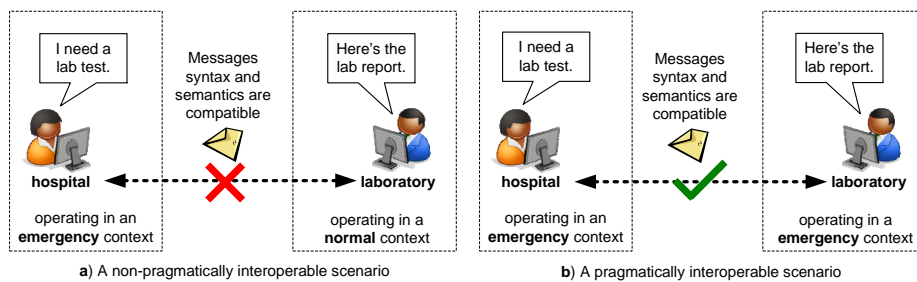


Fig. 1. A simple illustration of (non-)pragmatically interoperable scenarios.

3 Research Approach

This section describes the main research objective and questions that will be addressed, including the methodology to answer these questions. The main research objective of this PhD thesis is *to develop an approach that addresses the analysis, design, development and evaluation of a flexible and pragmatically interoperable solution for enterprise collaboration*. This thesis will also investigate and leverage goal-oriented approaches and service-oriented computing technologies to realize flexibility in the PI solution. Thus, the following research questions will be answered to address the research objective:

Q1 *How can Goal-Oriented Requirements Engineering (GORE) be used to elicit and specify requirements for PI?*

Essential in PI is the need for a message receiver to fully achieve the sender's desired effect. The sender's intention or goals must therefore be specified correctly. To do this, we investigate GORE as it has shown to be applicable in discovering and specifying requirements of a collaboration.

Q2 *How can Service Oriented Architecture (SOA) be leveraged to implement a PI solution for enterprise collaboration?*

SOA is a natural choice for developing interoperable solutions as heterogeneous systems can interact without changing the underlying implementation. Another aspect of PI is context-dependence; we will also explore how SOA can be used to achieve this.

Q3 *How can a PI solution be modelled using the notion of a service as a first class concept?*

Are current modelling notations and languages (e.g., BPMN, BPEL) sufficient to model a PI solution? If so, how can they be applied to design PI solutions? If not, what modelling concepts are needed to design PI?

Q4 *How can a PI solution be assessed, evaluated, and measured with respect to the goals of the business collaboration?*

The quality of the PI solution needs to be assessed with respect to the collaboration goals. We will therefore explore some assessment frameworks (e.g., through Key Performance Indicators) to do this.

Q5 *How can a PI solution be designed and implemented to provide flexibility to changing requirements?*

Innovation requires constant change. Enterprises should be able to adapt to rapid demands in the business. How can a PI solution be made flexible to respond to such changes?

We follow the regulative cycle proposed by Wieringa to structure the research methodology [17]. Figure 2 shows a high level sketch of this cycle as applied to this PhD project, including which research activities are conducted and which research questions are answered. In the cycle, two types of research problems can be distinguished: *knowledge problems* (KP) and *practical problems* (PP). A KP exists when there is a desire to change our knowledge of the world from what we currently know about it. A PP exists when there is a desire to change our experience of the world based on some goal. This distinction is important as such problems are solved differently: solutions to PPs are solved by satisfying some

criteria based on some goals; whereas, solutions to KPs are evaluated against a truth value, independent of the goal’s criteria.

The *problem investigation* phase solves a KP where we seek to understand what PI means and how current approaches propose to achieve it. To do this, we conduct a literature review of proposed definitions and approaches, and by drawing requirements that specify properties of a PI solution (including flexibility criteria). As there are various definitions of PI, our goal is to harmonize these definitions and validate them through examples and counter examples to understand PI better. One of the outputs of this phase is a conceptual framework that should aid our understanding of PI.

The output of the problem investigation will form part of the *solution design* phase. We apply in this phase a “treatment” in an attempt to solve the requirements set in the problem investigation phase. This treatment takes a form of a solution framework which will comprise of a *methodology* and an *architecture*. GORE will be used to design a methodology including modelling mechanisms that will specify solution artifacts. SOA will be used to design an architecture to realize PI requirements. Both the methodology and architecture will constitute an integrated PI solution design. This phase solves a PP.

The *design validation* phase solves a KP where we ask if the solution design, if implemented correctly, will lead us closer to the realization of the design requirements without implementing the solution design yet. We aim to develop an evaluation framework specifically for PI which will take into account various proposals for interoperability measurement from literature to validate PI properties against the solution design. Some prototypes and tools will be developed to assist in the validation. Thereafter, some tradeoff and sensitivity analyses will be done to weigh solution design decisions.

In the *solution implementation* phase, the solution design will be implemented using two case studies from the healthcare domain. The goal is to use case studies that are complex enough to allow sufficient validation of the solution design. This phase solves a PP in that we attempt to change current healthcare practices by introducing PI into the scene.

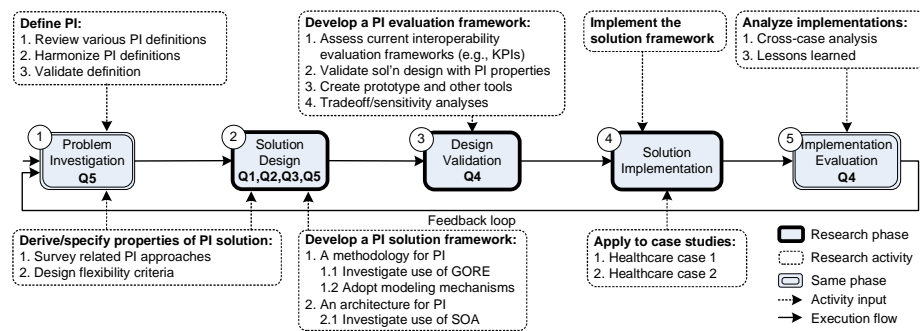


Fig. 2. The planned research methodology.

In the *implementation evaluation* phase, we solve a KP as we want to draw lessons from the implementation by doing a cross-case analysis of the two case studies. Additionally, the regulative cycle is iterative in that problem investigation can also occur during the implementation evaluation phase. Lessons learned in the implementation can improve problem understanding and solution design.

4 Related Work

This section provides a brief overview of some related approaches to PI. We note again that these approaches are positioned in various domains so that their interpretation of PI vary slightly from ours (*c.f.* [2] for a more detailed treatment).

In Kutvonen et al. (e.g., [7]), enterprises that participate in a collaboration are pragmatically interoperable if they have compatible business intentions, rules and organizational policies in order to perform digital business transactions. An eContract, or a collaboration contract, binds such business rules and policies including functional and non-functional properties of the collaboration. A B2B middleware, called webPilarcos, implements the eContract where facilities for finding, selecting and contracting relevant services are provided.

In Lee et al. (e.g., [8]), a so-called pragmatic or contextual knowledge is used to select the most appropriate service from among syntactically compatible, semantically equivalent, yet competing services for automatic service composition that meets the contextual requirements of the users. Rule-annotated ontologies (i.e. RuleML and OWL) specify in what situations a service can be used. PI is realized if a compositional agent is able to find a service with respect to the contextual requirements of a user.

In Liu et al. [9], the approach proposes a Pragmatic Web Service Framework for service request decomposition and aggregation where a service broker decomposes a request into finer sub-requests. A Web service abstract annotates a sub-request with the purpose and context of the request using a pragmatic frame. Achieving PI means having a service broker find the most appropriate concrete Web service than can satisfy the Web service abstract to form a business process workflow. How a concrete Web service meets the requirements of an abstract Web service is measured as the pragmatic distance.

In de Moor et al. [10], the approach proposes a solution to Web service selection in virtual communities (VC) for the communication of their members using collaborative tools. Guided by a methodology called RENISYS [11], relevant stakeholders of a VC interpret the usefulness of a set of syntactically and semantically compatible candidate Web services based on context of the VC's intended use, a process called pragmatic selection.

In Pokraev [13], the approach uses a service mediator that resolves data and process mismatches for enterprise application integration. A data mismatch occurs when systems have different denotations of the same message element in the real world. A process mismatch occurs when systems have a different understanding of message interaction protocols (i.e., the order of message exchange). PI is achieved when the sender and receiver of the message have the same expectation of the effect of message exchange which can be realized by ensuring that the proper order and execution of message invocations are followed.

In Tamani et al. [14], contextual information is used to select the most appropriate Web service above and beyond semantically equivalent services. Separate sender's and receiver's service profiles, specified in an ontology, describe their identity ("who part"), the purpose or goal ("why part") which specifies the context of the collaborating parties, and the input and output parameters ("what part"). A matching approach (e.g., through tokenization algorithms) thereafter matches the why part of the profiles to achieve PI.

In Tolk et al. (e.g., [15]), a framework is proposed for assessing the degree of conceptual representation between interoperable systems, known as the Levels of Conceptual Interoperability Model (LCIM). LCIM has seven levels of interoperability: no interoperability, technical, syntactic, semantic, pragmatic, dynamic, and conceptual. Focusing on the PI level, PI is reached when interoperating systems are aware of each other's methods and procedures; i.e., the data's context of use is understood.

In summary, we contribute in the following respects: There is not an approach that uses GORE to specify collaboration goals; we thus aim to contribute here (*c.f.* [1] for our initial work). The notion of context, though important in PI, has not been fully demonstrated; we aim to leverage SOA to enable this. Most approaches do not have flexibility in mind as a criteria to develop PI solutions; we aim to bridge this gap (*ibid*). SOA has indeed been used as the technology for enterprise collaboration, we aim to apply it here as well. Our solution combines both a methodological and architectural approach; most current solutions focus only on either of them. There is not an approach that provides a mature way of evaluating PI solutions; we aim to develop one. Finally, current solutions remain at the technical level; we aim to extend the solution to the business level.

5 Current Progress and Immediate Challenges

Our current work and contribution so far have largely been devoted to the problem analysis phase where we seek to explicate the notion of PI through an exhaustive review of extant literature. This was a necessary yet a difficult task as the research area on PI is still at its infancy. We have so far arrived at a working definition of PI as presented in Section 2. Our immediate next step is to validate the definition by applying it to several example and counter examples in the healthcare domain, and by gathering comments and suggestion (e.g., through the doctoral consortium). We have also conducted a literature review of related approach which shall serve as one of the sources for drawing requirements and properties of a PI solution [2].

Several new challenges have also risen from the literature review. One of the most challenging is distinguishing the role of context in achieving PI. Among such issues include the dynamicity of context which can affect the meaning of the request over time (thus, the effects of meaning evolution should be understood further). Other questions regarding context include: How much of context should be sufficient to achieve PI, seeing that contextual elements can accumulate infinitely? How do we decide which part of context to formalize (and hence, automate) and which part to leave out and still keep PI achievable, seeing that context can also bear tacit knowledge? What role, if any, does context have

in measuring the compatibility between the intended and the actual effect of message exchange to achieve PI?

Author's note: This research is part of the V-Care project supervised by dr. Marten van Sinderen (m.j.vansinderen@utwente.nl) and prof. Roel J. Wieringa (roelw@cs.utwente.nl) of the Information Systems Group, University of Twente, The Netherlands.

References

1. Asuncion, C.H., Jacob, M.E., van Sinderen, M.J.: Towards a Flexible Service Integration through Separation of Business Rules. In: 14th IEEE Int. Enterprise Computing Conf. (EDOC), Brazil. pp. 184–193. IEEE Comp. Soc. (2010)
2. Asuncion, C.H., Sinderen, M.: Towards Pragmatic Interoperability in the New Enterprise — A Survey of Approaches. In: 3rd IFIP Int. Working Conf. on Enterprise Interoperability (IWEI). LNBIP, vol. 76, pp. 132–145. Springer (2011)
3. Asuncion, C.H., van Sinderen, M.J.: Pragmatic Interoperability: A Systematic Review of Published Definitions. In: IFIP Enterprise Architecture, Integration, Interoperability and Networking (EAI2N), LNBIP, vol. 326, pp. 164–175. Springer Boston (2010)
4. Chen, D., Doumeingts, G., Vernadat, F.: Architectures for Enterprise Integration and Interoperability: Past, Present and Future. *CI* 59(7), 647 – 659 (2008)
5. Daugherty, P.J., et al.: Is Collaboration Paying Off for Firms? *Business Horizons* 49(1), 61 – 70 (2006)
6. Jardim-Goncalves, R., Grilo, A., Steiger-Garcia, A.: Challenging the Interoperability between Computers in Industry with MDA and SOA. *Computers in Industry* 57(8-9), 679–689 (2006)
7. Kutvonen, L., Ruohomaa, S., Metso, J.: Automating Decisions for Inter-enterprise Collaboration Management. In: *Pervasive Collaborative Networks*, pp. 127–134. Springer Boston (2008)
8. Lee, J., Lee, Y., Shah, S., Geller, J.: HIS-KCWater: Context-Aware Geospatial Data and Service Integration. In: *ACM-SAC*. pp. 24–29. Seoul, Korea (2007)
9. Liu, K.: Pragmatic Computing – A Semiotic Perspective to Web Services. In: *E-business and Telecommunications*, vol. 23, pp. 3–15. Springer Berlin (2009)
10. de Moor, A., van den Heuvel, W.J.: Web Service Selection in Virtual Communities. *37th Annual Hawaii Int. Conf. on System Sciences* 7, 70197 (2004)
11. de Moor, A., Jeusfeld, M.A.: Making Workflow Change Acceptable. *Requirements Engineering* 6, 75–96 (2001)
12. Morris, C.: *Foundations of the Theory of Signs*. University of Chicago Press (1938)
13. Pokraev, S.V.: *Model-Driven Semantic Integration of Service-Oriented Applications*. Phd thesis, University of Twente (2009)
14. Tamani, E., Evripidou, P.: A Pragmatic Methodology to Web Service Discovery. In: *IEEE Int. Conf. on Web Services (ICWS)*. pp. 1168–1171 (2007)
15. Tolk, A., Turnitsa, C., Diallo, S.: Implied Ontological Representation within the Levels of Conceptual Interoperability Model. *Intel. Decision Tech.* 2(1), 3–19 (2008)
16. van Sinderen, M.J.: Challenges and Solutions in Enterprise Computing. *Enterprise Information Systems* 2(4), 341–346 (2008)
17. Wieringa, R.: Design Science as Nested Problem Solving. In: *4th DESRIST*. pp. 8:1–8:12. ACM, New York, NY, USA (2009)