Towards Pragmatic Interoperability in the New Enterprise - A Survey of Approaches

Camlon Asuncion, Marten Sinderen

HAL Id: hal-01572095
https://hal.inria.fr/hal-01572095

Submitted on 4 Aug 2017

Towards Pragmatic Interoperability
in the New Enterprise
— A Survey of Approaches

Camlon H. Asuncion and Marten van Sinderen

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

Abstract. Pragmatic interoperability (PI) is the compatibility between the intended versus the actual effect of message exchange. This paper advances PI as a new research agenda within the gamut of enterprise interoperability research. PI is timely in today's new enterprises as it is increasingly important that organizations are able to collectively add value to their products and services through effective collaboration. When enterprise systems exchange information, PI goes beyond the compatibility between the structure and the meaning of shared information. It also considers the use of information in a given context as an equally important aspect. As PI is a relatively new domain, this paper attempts to elucidate its notion by identifying and relating its key concepts from proposed definitions, and by reviewing extant approaches to identify critical knowledge gaps in PI research.

Key words: enterprise interoperability, pragmatic interoperability, survey of approaches

1 Introduction

The International Organization for Standardization (ISO) defines an enterprise to be "one or more organizations sharing a definite mission, goals, and objectives to offer an output such as a product or service" [11]. This definition implies that either *one* or a *group* of organizations is involved. However, today's organizations remain competitive if they collaborate with other organizations with the objective of adding more value to their products and services (e.g. supply chain networks [7]). This cross-organizational nature of enterprises gave way to two types: a *virtual enterprise* (VE) and an *extended enterprise* (EE). While a VE temporarily exists between organizations to exploit an immediate market opportunity, an EE requires long-term trust and mutually dependent relationships [5]. These *new* enterprises have given rise to innovative business models that drive today's global economy [6].

Although today's enterprises want to leverage the benefits of their collaboration, interoperability problems between enterprise systems prevents them from doing so to the full extent [28]. Previous investments in equipment and software

cause incompatibilities between data representation and application methods. New acquisitions can cause impracticable integration with third party applications [12].

Vernadat [29] argues that interoperability in the enterprise: "provides two or more business entities [...] with the ability of *exchanging or sharing information* (wherever it is and at any time) and of using functionality of one another in a distributed and heterogeneous environment." Since organizations are not only composed of software but of people as well, achieving enterprise interoperability is multi-faceted [8]. Chen, et al. [5] argue that enterprise interoperability is ensured if there is interaction at least in the following layers: data, services, process and business.

In recent years, a number of researchers are also looking into the role of pragmatic interoperability (PI) as another layer in the general research of enterprise interoperability. They argue that while there is a need to agree on the structure and the meaning of the shared information, use of information is important as well. To provide a working definition of PI, we borrow Pokraev's [20] definition as being:

Definition 1. *The compatibility between the intended versus the actual effect of message exchange*

Various motivations from different domains have been put forward to advance PI as a new, timely and important research agenda. Among them, for example, is in the domain of collaborative enteprise computing where Kutvonen, et al. [13] argue that cross-organizational collaboration not only involves technology but social aspects of integration as well (i.e., pragmatics). In the domain of Web service discovery, Lee [14] argues that pragmatics can be used to select the most appropriate service from among a set of syntactically and semantically similar, yet competing, services. This view is shared by de Moor [17] where stakeholders interpret the appropriateness, suitability and applicability of a set of Web services, a process they call pragmatic selection in the domain of virtual communities. Singh, among others, proposes the notion of the Pragmatic Web [25], which entails that the meaning of information should be interpreted in the context of its use.

We recently conducted a systematic literature review to elucidate important concepts that can be derived from published definitions of PI [2] — essentially, we wanted to understand *what* PI currently means. The contribution of this paper, on the other hand, builds on previous work by investigating *how* current approaches propose to achieve PI. We do this by proposing a set of criteria whereby we can position the approaches accordingly.

The rest of this paper is structured as follows. Section 2 provides an elaboration on the notion of PI. Section 3 provides a framework to compare PI approaches. Section 4 compares extant approaches using the framework. Section 5 draws insights from the comparison. Finally, Section 6 presents conclusions and a brief research roadmap to further advance PI research.

2 Defining Pragmatic Interoperability

Our previous work focused on extracting dominant concepts from published definitions of PI as we have seen that there is a lack of canonical agreement as to how it is currently defined [2]. The following concepts from the proposed definitions of PI can be derived: *intention*, *exchange*, *use* and *context*.

A sender sends a message with some intention (also known as goal, need, preference, or intended effect). To realize this intention, the message must be delivered to the receiver, and hence, message exchange is part and parcel. On the receiver's end, message use deals with how the receiver acts to realize the intention of the message. However, a message must not be used arbitrarily but in a given context as its actual meaning varies according to the context in which it is used. Apart from this, context also changes over time which can affect the meaning of the message as well. Therefore, there must be a *shared* context which is *relevant* for the purpose of interoperation. Based on these concepts, we now propose a refinement of Definition 1 by stating PI as:

Definition 2. *The compatibility between the intended versus the actual use of received message within a relevant shared context.*

This is still consistent with Definition 1 in the sense that we relate *effect* to both *message use* and *context*. The notion of *message use* can be furthermore refined in terms of the *relevant subsequent use of information* and the *relevant subsequent actions using that information*. Essentially, we argue that the intended effect is achieved through a mutually dependent relationship between information, action, and context; i.e., *Actions on received information depend on context*.

– **Relevant subsequent use of information**: This denotes which relevant *information* (or part of the message) are to be used in a given context from the set of all data received during message exchange. We borrow this concept from Zeigler's notion of pragmatic frame [31].
– **Relevant subsequent actions**: This denotes which relevant subsequent *actions* the receiver can perform in a given context from the set of all possible actions. We borrow this from the Speech Act Theory (SAT) [23] and the Language/Action Perspective (LAP) [30], and from definitions given by such authors as Schade [22], Hofmann [10], Bazijanec [3], among others.

3 Criteria for Comparison

Seeing that the approaches are quite diverse in terms of their domain and implementation, a difficulty arises when they are compared with specific properties. Thus, we have resorted to devising a generalized set of criteria which is not too specific but still keeps the objective of this paper achievable. The approaches are compared in terms of the basic *architecture* it supports, the phase(s) of a development *life cycle* it contributes to, the role of *context*, the *adaptiveness* of

its solution, how it realizes *compatibility*, and where the *scope* of the effect is achieved. We now briefly describe each criterion.

Architecture. This criterion positions an approach according to the support it has in terms of the types of relations participants have in the collaboration (e.g., one-to-one, one-to-many or many-to-many), the distributed computing model of the architecture (e.g., client-server, peer-to-peer, n-tier, or space-based), and how the intended effect can be achieved through message exchange (e.g., single request or dialogue).

Life cycle. This criterion positions the approach in terms of which phase of an information systems development life cycle it contributes or applies to. We use the phases of the Systems Development Life Cycle (SDLC) [9]: planning, analysis, design, development, testing, and implementation. Although more elaborate frameworks can be used (e.g. ISO/IEC 12207, ISO/IEC 15288), we deem it sufficient to make a simplified and generalized comparison of the approaches using SDLC. The *planning* phase defines a development plan (e.g., project definition, scope, documentation, etc.) of the proposed system. The *analysis* phase involves understanding and documenting system requirements. The *design* phase involves creating a "blueprint" of how the proposed system will work. The *development* phase involves building the actual system according to the design documents. The *testing* phase involves verifying if the actual system meets the requirements of the analysis phase. The *implementation* phase begins when the system becomes operational by the end users. Finally, the *maintenance* phase involves monitoring the actual system to ensure its compliance with business goals.

Context. Our previous work [2] showed that a number of approaches consider context to be important in PI. This criterion asks if an approach is dependent on the notion of context to achieve PI. If so, we classify the role of context using the two refinements of message use described in Section 2; i.e, message use as the relevant subsequent use of information, or the relevant subsequent actions using that information as means to achieve the intended effect, in a given context.

Adaptiveness. A pressing requirement in today's new enterprises is to be adaptive to change. Adaptation compels an enterprise to respond to changes because of the conditions of its environment in an effort to survive [4]. This imposes on the PI solution a high degree of flexibility. We use this criterion to position an approach as to whether it has some mechanisms that allow the solution to swiftly adapt to new situations and still remain interoperable.

Compatibility. Recalling Definition 2, the key relation between the intended versus the actual effect is compatibility, which can be achieved either absolutely or relatively. By relative, we mean that achieving the intended effect is approximate in that, for example, one sender may be satisfied at a different level from another sender. By absolute, we mean that the intended effect is either present entirely or not; i.e, either PI is there or isn't there.

Scope. This criterion positions the approach as to where the intended effect is ultimately realized. We propose in our previous work [2] that message exchange can create a intended effect either at the application level or business level. For

each approach we ask: Does the intended effect remain at the digital world only (e.g., the update of a data item in a database column) or does the intended effect propagate to the business level to achieve a business objective; i.e, outside the digital world and into the real world?

4 Approaches to Pragmatic Interoperability

We now describe briefly and position accordingly each approach using the criteria. Of the 44 papers that were identified from our previous work [2], 9 are discussed here as they provide some level of detail as to how PI can be achieved; 24 papers, however, do not provide enough detail to warrant a comparison. The remaining 11 papers are in the domain of Multi-Agent Systems (MAS) which are deliberately excluded here.

Kutvonen et al. [13] The approach supports the collaboration of VEs, or eCommunities as the authors call them, where organizations can dynamically participate (i.e., they can join, leave, or be removed for various reasons) in global business environments based on open markets to achieve a certain business opportunity. An eCommunity is a venue for participants to be pragmatically interoperable as they can negotiate and agree on the compatibility of their business rules and policies.

The approach uses the concept of an eContract which not only encapsulates the rules and policies, but functional and non-functional properties of the collaboration as well. An eContract is implemented under an automated environment, called the webPilarcos B2B middleware, where facilities for finding, selecting and contracting relevant services are provided. The middleware is designed to follow a federated form of interoperation where services are developed independently by organizations and no meta-information is assumed to be shared allowing the approach to be adaptive. The middleware supports a many-to-many collaboration any organization can dynamically participate in the eCommunity, an n-tier distributed model is supported by the middleware, and dialogue is necessary during negotiation to agree on the eContract. The approach provides contribution to the analysis, design, and development of the eContract, and thereafter, maintenance and compliance monitoring using the eContract. It is oriented towards providing a PI solution at the business level achieving scope at that level. The role of context, however, is not explicitly explained.

Lee et al. [14] The approach proposes a solution for the automatic composition of Web services into a service workflow which combines simple services to form a complex one that meets user requirements. When a service requester is faced with a set of syntactically compatible, semantically equivalent, yet competing services, contextual knowledge can be used to select the most appropriate service. Pragmatic or contextual knowledge describes how and in what situations provided services can satisfy user requirements in a given context. Contextual knowledge is modelled using rule-annotated ontologies (e.g., RuleML and OWL).

An architecture uses a service composition agent to perform the necessary discovery and selection using pragmatic knowledge. We thus view interoperabil-

ity in their approach to be one-to-many and n-tier; however, it appears that once the selection of the service is completed, achieving the intended effect occurs using a single service request only. The approach seems to focus more on the analysis, design, and development of the PI solution rather than with other phases. Context is used as the key component in selecting the most appropriate service; in particular, the composition agent uses contextual information of the service requester during the service selection process. The approach is adaptive in the sense that once the contextual information of the service requester changes, the most appropriate service can then be selected to compose a new service workflow. Compatibility seems to be absolute since if the composition agent cannot find an appropriate service based on the context of the service requester, the service workflow cannot be generated as well. Finally, the approach is largely oriented towards achieving PI at the application level.

Liu et al. [16] To design service-oriented applications, the approach leverages pragmatics (a component of the Theory of Semiotics [19]), which is concerned with the use of information in relation to its intended purpose in a given context. The approach proposes a Pragmatic Web Service Framework for service request decomposition and aggregation. Here, a service broker decomposes a request into finer sub-requests that specify requirements that a concrete Web service should satisfy. A sub-request is annotated with its own semantic and pragmatic (i.e., purpose and context, or a pragmatic frame) definitions which are then represented as Web service abstracts. A workflow is then designed using business process requirements and the Web service abstracts. To implement the workflow, the service broker uses pragmatic knowledge to find concrete Web services using the Web service abstract and the pragmatic frame. How a concrete Web service meets the requirements of an abstract Web service in terms of the pragmatic frame is measured as the pragmatic distance.

The interaction is one-to-many where one request can be satisfied by many providers, mediated by a service broker. Once an appropriate concrete Web service is selected, a single request seems to achieve the intended effect. The approach can be positioned in terms of the analysis, design, and development phases of the life cycle. Context is important, and the approach makes use of illocutionary acts from the SAT to achieve intended effect. The notion of pragmatic distance leads us to believe that compatibility can be achieved relatively; e.g., requesters may select concrete Web services with varying pragmatic distances. Since the approach is preliminary, we cannot sufficiently assess the adaptiveness of the solution. Finally, we view the scope of the approach to be at the application level.

de Moor et al. [17] The approach proposes a solution to Web service selection in virtual communities (VC) for the communication of their members using collaborative tools. Particularly challenging in VCs is that tacit knowledge cannot be completely and explicitly represented. In this respect, the authors leverage pragmatic concepts to provide a context-dependent mechanism so that service selection can be tied to the context of its intended use. This is important as VCs use services in novel and unique ways depending on their context. Guided

by a methodology called RENISYS [18], relevant stakeholders of a VC interpret the usefulness of a set of syntactically and semantically compatible candidate Web services based on certain criteria. This is called as the process of pragmatic selection.

We cannot assess the architectural aspect of the solution as no architecture is used, but rather a methodology. RENISYS can be positioned mostly during the analysis stage when a VC selects the most pragmatically compatible set of Web services according to the VC's context of use. Context is critical as it will determine what sort of services a VC requires; however, it is not clear if context is described in terms of relevant information or relevant actions. Compatibility is relative as a virtual community may have different and unique requirements from another. Finally, the methodology is focused at achieving the intended effect ultimately at the business level.

Pokraev [20] The approach uses service-oriented and model-driven principles to provide solutions in the domain of Enterprise Application Integration (EAI) where multiple, heterogenous, autonomous and complex distributed systems need a flexible interoperability solution to collaborate effectively. It implements a service mediator that resolves data and process mismatches between collaborating systems with fixed service descriptions. 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.

We view the compatibility of the approach to be rather absolute as the specific order of service invocation must be observed. The approach proposes a five-step methodology for designing integration solutions which covers analysis, design, development, and testing. However, it does not explicitly consider the role of context in PI. As the approach leverages model-driven techniques, a key benefit is abstraction; i.e., with the same abstract solution depicted as platform-independent models, new implementations can be automatically generated, allowing adaptiveness. Furthermore, there is also work that extends the approach whereby business rules are used to separate the dynamic aspects of the requirements making the approach even more flexible [1]. We position the approach as having a business level scope.

Rukanova [21] The use of a standard can help organizations be interoperable. However, the challenge lies in determining if whether or not a standard is sufficient to meet the communication requirements of an organization's digital business transaction (DBT), where a DBT is an electronic exchange of products and/or services for some remuneration. Thus, when organizations choose to adopt a standard to perform DBTs with others, the "fit" or the capabilities of a standard vis-a-vis the requirements of a DBT should first be evaluated. To do this, the approach proposes a meta-model that can be used as basis for the evaluation, with an accompanying methodology for doing such evaluation. The

theoretical foundations of the meta-model are derived from theories related to the pragmatics of communication, where context plays an important role. The approach argues that when a standard fits a DBT, PI is satisfied.

The architectural aspect of the approach cannot be positioned as no architecture is used, but rather a methodology. The methodology can be positioned at the analysis phase of the life cycle where it helps assess the fit of a standard for a given DBT. In terms of context, the meta-model uses the communicative acts of the SAT and LAP to describe the pragmatic aspect of the communication. Compatibility is therefore absolute as a selected standard should be able to address all communicative patterns of LAP; otherwise, pragmatic interoperability problems may occur. On the other hand, it is not clear how the methodology would allow the solution to be adaptive to business changes. Finally, we position the scope at the business level as the ultimate effect is realized through an effective DBT.

Tamani et al. [26] Motivated by the argument that collaborating parties use Web services differently according to their own context, the approach proposes a solution levaraging principles of the Pragmatic Web [25] for automated Web service discovery, where context is used to select the most appropriate service above and beyond semantically equivalent services. The approach uses XML to capture and XPath to query the personal, functional, and contextual information of collaborating parties stored in profile repositories. These XML profiles specify each requester's and provider's identity and role (the "who" part), the purpose and goal (the "why" part that specifies the context of the collaborating parties), the input and output parameters annotated by some ontology structure (the "what" part). A matching architecture is used to semantically match the "what" part of the request/offer XML and pragmatically match the "why" part of the request/offer XML. The Web service of the offer XML that is semantically and pragmatically closest to the request XML, through string tokenization algorithms, is selected.

Architecturally, the approach uses a one-to-many interaction as one requester may need to discover from one among many offers from the profile repository. A matcher sits in between the requester and provider that resolves semantic and pragmatic differences between the two. A single request, however, seems to be sufficient to achieve the intended effect. The approach falls within the design and development stages of the life cycle. Context is important by treating only the similar contextual information specified as "why" part of the request/offer XMLs. On the other hand, it is difficult to assess the adaptiveness of the approach as this criterion is not well described. Compatibility is rather absolute in the sense that if a matcher cannot find similarities between the "why" part of the request/offer XMLs, PI cannot be achieved. Finally, the intended effect is only realized at the application level through the successful matching of contextual information.

Tolk et al. [27] The approach proposes a framework for measuring the degree of conceptual representation between interoperable systems, known as the Levels of Conceptual Interoperability Model (LCIM), which is highly influenced

by the Modelling and Simulation community. LCIM has seven levels of interoperability: no interoperability, technical, syntactic, semantic, pragmatic, dynamic, and conceptual. Focusing on the PI level, PI is reached when "when the interoperating systems are aware of each others methods and procedures. In other words, the use of the data — or the context of its application — is understood by the participating systems; the context in which the information is exchanged is unambiguously defined". LCIM takes two roles: descriptive and prescriptive, Descriptively, LCIM can be used as a documentation which serves as a maturity model whereby properties of an interoperating system is measured against an LCIM level. Prescriptively, LCIM has a set of methods for achieving a target interoperability level.

As LCIM is largely an assessment framework, no architecture is proposed. Furthermore, as LCIM can be used descriptively and prescriptively, we position it in terms of the analysis, design and testing phases of the life cycle. Context relates to how information is used in a given state and specification of the system. LCIM does not clearly discuss if reaching the PI level entails adaptiveness of the solution. We view compatibility as absolute because the meaning of information depends on the state and specification of the system. Finally, we view the scope as that which is confined only to the interactions of the systems.

Zeigler et al. [31][24] In the Modelling and Simulation domain, the approaches positions PI in one of three levels: syntactic, semantic, and pragmatic. At the pragmatic level, focus is on the receivers' interpretation of messages in the context of its application relative to the senders intent. Thus, at the pragmatic level, there is a shared agreement between the sender and receiver of a message as to how information is to be used; i.e, the receiver reacts to the communicated message in a manner that the sender originally anticipated (with the assumption that there is non-hostility in the collaboration).

The approach introduces the notion of a pragmatic frame. Information exchange involves reporting or requesting changes in the state of the world between a sender and a receiver. When a sender produces information by capturing the state of the world (through some form of ontology, derived from his own perspective of the world), the pragmatic frame characterizes how a receiver would use that information (which maybe the full or part of that ontology). Thus, the pragmatics of use describes how data will be subsequently processed.

Architecturally, the approach supports a one-to-many type of interaction, n-tier, and single request. We position the approach in terms of the analysis, design, development phases of the life cycle. Context is an important concept that emphasizes on the subsequent use of information. The approach is not clear as to whether adaptiveness is reached when PI is achieved. Compatibility is relative in the sense that the subsequent use of relevant information may depend on the receiver and its context. We position the scope to be at the business level as the ultimate effect is realized by the receiver of the message when it uses the acquired relevant information.

Table 1. A summary of the comparison among pragmatic interoperability approaches

	Architecture[α]	Life cycle[φ]	Context[φ]	Adaptiveness	Compatibility[ξ]	Scope[δ]
Kutvonen et al. [13]	M2M/NT/D	A, Des, Dev, M	No	Yes	R	B
Lee et al. [14]	O2M/NT/S	A, Des, Dev	Yes, I	Yes	A	A
Liu et al. [16]	O2M/NT/S	A, Des, Dev	Yes, A	?	R	A
de Moor et al. [17]	none	A	Yes, ?	?	R	B
Pokraev [20]	O2M/NT/D	A, Des, Dev, T	No	Yes	A	B
Rukanova [21]	none	A	Yes, A	?	A	B
Tamani et al. [26]	O2M/NT/S	Des, Dev	Yes, I	?	A	A
Tolk et al. [27]	none	A, Des, T	Yes, I	?	A	A
Zeigler et al. [31]	O2M/NT/S	A, Des, Dev	Yes, I	?	R	B

[α] *O2O*: one-to-one, *O2M*: one-to-many, *M2M*: many to many / *P2P*: peer-to-peer, *NT*: *n*-tier, *CS*: client-server / *S*: single request, *D*: dialogue

[φ] *P*: planning, *A*: analysis, *Des*: design, *Dev*: development, *T*: testing, *I*: implementation, *M*: maintenance.

[φ] *I*: relevant subsequent use of information, *A*: relevant subsequent actions

[ξ] *A*: absolute, *R*: relative

[δ] *B*: business level, *A*: application level

5 Discussion

A summary of the comparison among PI approaches with respect to the previously described criteria is given in Table 1. In this section, we first summarize and compare approaches per criterion to ascertain their differences and similarities. Thereafter, we attempt to identify critical knowledge gaps in PI research based on this comparison. Summarizing the approaches per criterion:

Architecture. Although there are more architectural approaches to PI, either a methodology or an assessment framework is also used. Among the architecture-based approaches, a one-to-many type of relation is mostly supported. It is often the case that an intermediary software is used to coordinate participants in the collaboration (i.e., a middleware, mediator, matcher, or a Web service broker). Particularly, among service-oriented approaches, a single request is often the type of message exchange used to achieve the intended effect. We also notice that the "one-to-many : single request" pair occurs more frequently.

Life cycle. Most of the approaches can be positioned in terms of the analysis, design and development phases of the life cycle. No approach adequately supports the planning phase. Furthermore, there are very few approaches that support the testing and maintenance phases. Furthermore, it is interesting to

note that non-architectural approaches contribute more to the analysis phase where the focus is on requirements elicitation; e.g., pragmatics is seen as a technique during service selection where the most appropriate service is chosen based on contextual information.

Context. A large number of approaches value context as an integral part of achieving PI. This is due to the fact that these approaches leverage pragmatic theories such as Semiotics, SAT, and LAP as foundations for designing their solutions. The difference lies, however, in their interpretation of context: either the focus is more on the relevant subsequent information use or relevant subsequent actions performed; we observe, however, that neither approach supports these types simultaneously. Approaches that do not use pragmatic theories are not explicit about their support for context.

Adaptiveness. Most of the approaches are not clear as to how adaptive their solutions are, and hence we cannot position them accordingly. For approaches which we have positioned as adaptive, we cannot directly attribute their adaptiveness to their use of pragmatics. An exception, however, is the one of Lee [14] where the change in context can be used to dynamically recreate a service workflow to meet the service requester's requirements by selecting the most appropriate service relative to the requester's context.

Compatibility. The approaches vary greatly in terms of achieving the intended effect either relatively or absolutely. There also seems to be a pattern: approaches whose scope is at the application level (i.e., that is the focus of the approach is rather technical, and, hence, the effect remains in the digital world) achieve PI absolutely. On the other hand, approaches whose scope is at the business level, achieving the intended effect is relative.

Scope. The approaches also vary in terms of scope. Some approaches, which are mostly technical (and, hence, the compatibility is absolute), achieve the intended effect at the application level; whereas, approaches with a relative compatibility achieve the intended effect at a business level scope. It is also interesting to note that single-request approaches have the scope of application level; whereas, dialogues are common in approaches whose scope is at the business level.

A number of critical knowledge gaps in PI research can be observed. We find that there is a need to bridge the gap among approaches that provide only architectural or methodological solutions to PI. We argue that to advance PI research, an architectural *and* methodological approach, which are both complementary, should be devised.

Currently, no approach seems available that supports the development of PI solutions in each phase of a software development life cycle (i.e., from planning to maintenance). Challenges are yet to be identified when pragmatic principles are considered in every phase of a system development life cycle specific to the goal of developing PI solutions. One promising approach is called MEASUR (Methods for Eliciting, Analysing and Specifying Users' Requirements)[15] that takes a semiotic approach to information systems development.

We still need to deepen our understanding in terms of which form of message exchange is necessary and sufficient to achieve the intended effect in PI: is it through a single request or through a dialogue? In human linguistics, pragmatic effects of communication are usually achieved through a dialogue that depends significantly on the prevailing context; how do we translate this when enterprise systems communicate pragmatically?

Our notion of context in PI needs further exploration as various approaches characterize context differently. This seems to be the most important aspect in PI, yet the most difficult to get to grips with. The two refinements of message use we described in Section 2 (i.e., relevant subsequent information use and relevant subsequent action in achieving PI) need consolidation as neither of the approaches treat both as important concerns together. Finally, it has been argued that not everything about context can be formalized [17]. Questions thus remain: how much of context can or should we automate for the purpose of PI? How do we handle context that is difficult to automate but is nevertheless important for achieving PI?

As we have mentioned, adaptiveness and flexibility are critical in today's new enterprises. There is a need to create approaches that are adaptive to change. Already we find that this has been one of the goals of more recent PI approaches; however, evidence is yet to be had to ascertain whether the adaptiveness of the solution can be directed attributed to its being pragmatic. Demonstrating adaptiveness may allow PI research rapid adoption in the industry.

PI is a new and still less-developed research area as evidenced by the lack of implementation and validation in real life scenarios and the diversity of the terminologies used. Application of pragmatics-based solution remains to be seen in real life settings. Already we find that PI has been applied in various domains, is it possible to conceptualize PI so that it becomes applicable in various domains, similar to how semantic interoperability is understood in whatever domain it is applied?

Finally, no approach offers a mature way to *measure* the intended effect especially in both cases when the compatibility is either absolute or relative, or the scope is either at the business level or application level. Measuring PI especially becomes difficult when the business level is considered. One promising measure is proposed by Liu [16] called the pragmatic distance at the application level; however, implementation details remain to be seen.

6 Conclusion and Future Work

Pragmatic interoperability (PI) is the compatibility between the intended versus the actual use of message exchange within a relevant shared context. The interest in PI has been increasing recently as more and more researchers argue that agreeing on the syntax and semantics during message exchange is not enough, how information is used in a given context is important as well to reap the benefits of a truly effective interoperability solution. The application of pragmatics in information system research, in general, and PI research, in particular, still

remains preliminary. This paper attempts to advance this field by harmonizing different concepts related to the notion of PI from among currently diverse interpretations of its concept. We also position PI approaches according to a set of criteria so that critical gaps in knowledge, related to PI research, can be elucidated.

In summary, some of these knowledge gaps include developing an approach where both architectural and methodological aspects are considered, studying the impact when pragmatic principles are considered in every phase of a system development life cycle, exploring the role of context in PI, and validating approaches in real life settings.

As we have described earlier, there are several other approaches to PI specific to the MAS domain. We have not considered them in this paper deliberately as we believe that they deserve a separate review. We intend to do this in the future, and perhaps perform a further comparison between MAS and non-MAS approaches to PI. Finally, it is our aim to envision a future solution where service-oriented principles will be used to achieve PI particularly in the healthcare domain.

References

1. Asuncion, C.H., Iacob, M.E., van Sinderen, M.J.: Towards a Flexible Service Integration through Separation of Business Rules. In: 14th International Enterprise Computing Conference, Brazil. pp. 184–193. IEEE Computer Society (2010)
2. Asuncion, C.H., van Sinderen, M.J.: Pragmatic Interoperability: A Systematic Review of Published Definitions. In: IFIP Enterprise Architecture, Integration and Interoperability (EAI2N), vol. 326, pp. 164–175. Springer Boston (2010)
3. Bazijanec, B., Zaha, J.M., Albani, A., Turowski, K.: Establishing Interoperability of Coordination Protocols in Ad Hoc Inter-Organizational Collaborations. In: Interoperability of Enterprise Software and Applications, pp. 123–133. Springer London (2006)
4. Chakravarthy, B.S.: Adaptation: A Promising Metaphor for Strategic Management. The Academy of Management Review 7(1), pp. 35–44 (1982)
5. Chen, D., Doumeingts, G., Vernadat, F.: Architectures for Enterprise Integration and Interoperability: Past, Present and Future. Computers in Industry 59(7), 647 – 659 (2008)
6. Chung, W.W.C., Yam, A.Y.K., Chan, M.F.S.: Networked Enterprise: A New Business Model for Global Sourcing. International Journal of Production Economics 87(3), 267 – 280 (2004)
7. Davis, E., Spekman, R.: The Extended Enterprise: Gaining Competitive Advantage through Collaborative Supply Chains. Financial Times Prentice Hall Books, Financial Times/Prentice Hall (2004)
8. Doumeingts, G., Müller, J.P., Morel, G., Vallespir, B.: Enterprise Interoperability: New Challenges and Approaches. Springer (2007)
9. Haag, S., C.M., McCubbrey, D., Pinsonneault, A., Donovan, R.: Management Information Systems for the Information Age. McGraw-Hill (2006)
10. Hofmann, M.: Challenges of Model Interoperation in Military Simulations. Simulation 80(12), 659 (2004)

11. ISO: ISO 15704: Industrial Automation Systems — Requirements for Enterprise Reference Architectures and Methodologies (1999), `http://www.mel.nist.gov/sc5wg1/gera-std/15704fds.htm`
12. Jardim-Goncalves, R., Grilo, A., Steiger-Garcao, A.: Challenging the Interoperability between Computers in Industry with MDA and SOA. Computers in Industry 57(8-9), 679–689 (2006)
13. Kutvonen, L., Ruohomaa, S., Metso, J.: Automating Decisions for Inter-enterprise Collaboration Management. In: Pervasive Collaborative Networks, pp. 127–134. Springer Boston (2008)
14. Lee, J., Lee, Y., Shah, S., Geller, J.: HIS-KCWater: Context-Aware Geospatial Data and Service Integration. In: 2007 ACM Symposium on Applied computing. pp. 24–29. ACM, Seoul, Korea (2007)
15. Liu, K.: Semiotics in Information Systems Engineering. Cambridge Univ Pr (2000)
16. Liu, K.: Pragmatic Computing — A Semiotic Perspective to Web Services. In: Communications in Computer and Information Science, vol. 23, pp. 3–15. Springer Berlin Heidelberg (2009)
17. 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)
18. de Moor, A., Jeusfeld, M.A.: Making Workflow Change Acceptable. Requirements Engineering 6, 75–96 (2001)
19. Morris, C.: Foundations of the Theory of Signs. University of Chicago Press (1938)
20. Pokraev, S.V.: Model-Driven Semantic Integration of Service-Oriented Applications. Phd thesis, University of Twente (2009)
21. Rukanova, B.: Business Transactions and Standards: Towards a System of Concepts and a Method for Early Problem Identification in Standard Implementation Projects. Phd Thesis, University of Twente (2005)
22. Schade, U.: Towards the Edge and Beyond: The Role of Interoperability. In: 10th International Command and Control Research and Technology Symposium (2005)
23. Searle, J.: Speech Acts: An Essay in the Philosophy of Language. Cambridge University Press (1970)
24. Seo, C., Zeigler, B.: DEVS Namespace for Interoperable DEVS/SOA. In: 2009 Winter Simulation Conference (WSC). pp. 1311 –1322 (2009)
25. Singh, M.P.: The Pragmatic Web. IEEE Internet Computing 6, 4–5 (2002)
26. Tamani, E., Evripidou, P.: A Pragmatic Methodology to Web Service Discovery. In: IEEE Int. Conf. on Web Services (ICWS). pp. 1168–1171 (2007)
27. Tolk, A., Diallo, S., King, R., Turnitsa, C.: A Layered Approach to Composition and Interoperation in Complex Systems. In: Complex Systems in Knowledge-based Environments: Theory, Models and Applications, vol. 168/2009, pp. 41–74. Springer Berlin (2009)
28. van Sinderen, M.J.: Challenges and Solutions in Enterprise Computing. Enterprise Information Systems 2(4), 341–346 (2008)
29. Vernadat, F.: Interoperable Enterprise Systems: Principles, Concepts, and Methods. Annual Reviews in Control 31(1), 137–145 (2007)
30. Weigand, H.: Two Decades of the Language-Action Perspective. Communications of the ACM 49(5), 45 (2006)
31. Zeigler, B.P., Hammonds, P.E.: Modeling & Simulation-Based Data Engineering: Introducing Pragmatics into Ontologies for Net-Centric Information Exchange. Academic Press (2007)