IPSME- Idempotent Publish/Subscribe Messaging Environment [v0.3]

Kim Nevelsteen
kim.nevelsteen@mitm.se
MiTM AB

Martin Wehlou
martin.wehlou@mitm.se
MiTM AB

Abstract

The integration of disparate systems is required in the domain of Cyber Physical Systems, applied to Massive Multiuser Virtual Environments *e.g.,* computer video games and health-care applications. The problem of integrating systems that this article addresses is summarized as: 1) interoperability, getting highly disparate systems to communicate with minimal change to each system; 2) semantic mapping, designing communication so as to alleviate part of the complexity behind resolving semantic heterogeneity; and 3) scalability, finding a solution that allows for interoperability, while also being scalable. A publish/subscribe system is often cited as a communication infrastructure that enables data access and sharing over disparate systems, but various properties such as expressiveness or those related to quality of service often hamper interoperability, and in turn the scalability required for wide-area networks. A set of conventions describing an Idempotent Publish/Subscribe Messaging Environment (IPSME) are introduced to facilitate the communication avoiding the requirement of standardization, alleviating some complexity of resolving semantic heterogeneity, and without sacrificing scalability.

keywords: publish/subscribe; wide-area; integration; interoperability; scalability; semantic mapping; disparate; cyber-physical systems; medical; metaverse; virtual; MMVE.

1 Introduction

The integration of disparate systems is required if, for example, a viable Metaverse is to exist [Dionisio, Burns III, and Gilbert 2013] or the architecture enabling the Internet of Things (IoT) fragments [Nevelsteen, Kanter, and Rahmani 2016]. Both concepts are incorporated in the domain of Cyber Physical Systems [Rehm, Goel, and Crespi 2015], with the application thereof, in this article, initially aimed at games that make use of multiple virtual worlds. Given the strict definition of 'virtual world' [Nevelsteen 2018] and that the integration problems being faced were similar to those in the medical field [Barthell et al. 2004; Wehlou 2014], the domain was instead broadened to be applicable to Massive Multiuser Virtual Environments (MMVEs) [Nevelsteen 2018].

Publish/subscribe (pubsub) systems are increasingly popular in cyber-physical infra-structures and health-care applications [Uzunov 2016]. Wang et al. [2002] state that pubsub is "a communication infrastructure that enables data access and sharing over disparate systems and among inconsistent data models". Wang et al. [2002] reference Carzaniga, Rosenblum, and Wolf [2001], who introduce a pubsub system for distributed applications deployed over wide-area networks *e.g.,* The Internet. The problem with current pubsub systems is that various properties such as expressiveness or those related to quality of service often hamper interoperability, and also in turn the scalability required for wide-area networks.

To address this problem, this article provides – not a framework or standardized protocol, which are too restrictive – but, a set of conventions that enables integration by: facilitating the communication of highly disparate systems, alleviating some complexity of resolving semantic heterogeneity, and without sacrificing scalability. The set of conventions together form an Idempotent Publish/Subscribe Messaging Environment (IPSME).

2 Why Your Data Won't Mix

A common approach for getting two highly disparate systems to communicate is by having them speak the same protocol: either by having a standardized protocol with author specific implementations for each system, or by having a non-standard protocol with an SDK or framework implementing the same protocol for each system. The problem of integrating systems arises when many of these protocols or versions thereof come into existence (there will always be some feature lacking for those wanting to achieve something which was not initially envisioned or anticipated). Semantic mapping is then required to bridge protocols, with "different semantic and data representations" [Halevy 2005]. Halevy [2005] claims that "in a typical data integration scenario, more than half of the effort (and sometimes up to 80 percent) is spent on creating the mappings, and the process is labor-intensive and error-prone". Semantic mapping is an open topic of research.

The problem this article addresses is summarized as: 1) interoperability, getting highly disparate systems to communicate with minimal change to each system; 2) semantic mapping, designing communication so as to alleviate part of the complexity behind resolving semantic heterogeneity; and 3) scalability, finding a solution that allows for interoperability, while also being scalable.

3 Related Work

The integration of highly disparate systems is an open topic of research in many domains. If the architecture enabling IoT fragments, then "accessing IoT will require the interfacing with many platforms" [Nevelsteen, Kanter, and Rahmani 2016] *i.e.,* integration. Rehm, Goel, and Crespi [2015] state that IoT has been "recently been replaced by the broader, more tangible, concept of Cyber-Physical Systems" for which they "believe that virtual worlds can serve as platforms to facilitate integration".

More specifically, interoperability was noted as a "well-known problem" for Pervasive Games [Nevelsteen 2015]; whatever solution is inevitably used to solve the interoperability problem must also be scalable, if it is to be applicable to distributed pervasive applications, incorporating IoT [Nevelsteen, Kanter, and Rahmani 2016]. If the IoT architecture fragments, then "the fragmented platform structure plus pervasive applications, aligned with virtual worlds, resembles the concept of the Metaverse" [Nevelsteen 2016] *i.e.,* a system of interconnected virtual worlds [Frey et al. 2008]. The underlying interoperability and scalability problems are corroborated by Dionisio, Burns III, and Gilbert [2013], by stating that both are central components of a viable Metaverse.

Pubsub schemes are usually topic-, content- or type-based [Eugster et al. 2003; Uzunov 2016]. In designing for pubsub over a wide-area, Carzaniga, Rosenblum, and Wolf [2001] have defined the term 'expressiveness' as the "power of the data model that is offered to publishers and subscribers of notifications", and declared that there is a trade-off between expressiveness and scalability. If maximum scalability, for possibly Internet-scale communication is desirable, then choosing the right data model for messages is vital. Eugster et al. [2003] states that string as keys in messages enforces interoperability, however even strings impose constraints *e.g.,* discrepancy in character encoding, byte order or string delimiters.

To achieve interoperability, data must be integrated from multiple sources, and "both semantic-level and data-level heterogeneity" must be handled [Halevy 2005]. Landman et al. [2011] state that if various sources offer open interfaces to their data, combined with standardized data formats, interoperability can be achieved. And, there are those [Morgan 2009; Branton, Carver, and Ullmer 2011; Barthell et al. 2004] that "argue that the way to resolve semantic heterogeneity is through standard schemas". But, standardization only has limited success [Halevy 2005] with the possibility that a proposed standard can be outdated before it is standardized. Precisely the reason integration is needed is due to the fact that various sources speak different standard protocols or various versions thereof *e.g.,* Barthell et al. [2004] suggest having standards for multiple specialized usages. If multiple standards are established, then semantic mapping must be used for integrating and "there is not necessarily a single correct semantic mapping" [Halevy 2005]. In addition, when standards are updated, any existing semantic mappings must also be updated exaggerating the problem *e.g.,* in a component-based architecture, Barthell et al. [2004] require that "each component should be able to accept real-time standardized messages" as input and output from various sources.

Sometimes a centralized architecture is used to enable certain properties (*e.g.,* allowing for asynchrony and persistence), but such a centralized architecture remains a bottleneck and single point of failure. In other words, "scalability also often conflicts with other desirable properties" [Eugster et al. 2003]. This is valid for pubsub subscriptions as well where "highly expressive and selective subscriptions require complex and expensive filtering and routing algorithms, and thus limit scalability" [Eugster et al. 2003]. Besides pubsub, other communication paradigms exist (*e.g.,* message passing, RPC, shared spaces, etc.), but many of these paradigms have space, time or synchronization coupling hampering scalability [Eugster et al. 2003].

4 Introducing IPSME

IPSME is introduced to enable the integration of highly disparate systems. Rather than a framework or standardized protocol, IPSME is a set of conventions that allows any participant to talk to any other participant, without a central authority and without the need for standardization, provided groups of participants speak the same protocol.

IPSME defines the following conventions:

- A messaging environment (ME) is defined as:
  - A pubsub system which relays messages to listening participants.
  - Messages must be idempotent or identifiable as duplicates.
  - Messages are simply ignored if not understood by any participant.
  - Transformation of messages is done by having a participant listen to received messages and sending out transformed versions.
- Each participant sends and receives messages in a (local) ME.
- Communication across ME boundaries is through reflectors; a participant listener with a counterpart in another ME, which resends messages there. Communication between pairs is left unspecified and completely up to the author of those participants, as is the selection of which messages to resend.

An author of a participant is free to implement their own message protocol as long as these conventions are met. The set of conventions is purposely kept non-restrictive for easy adoption. Together these conventions define a 'dataspace' [Halevy 2005] consisting of one or more MEs, each with various participants and independent relationships to other participants. The

IPSME conventions are explained in detail pertaining to interoperability, semantic mapping and scalability, in the following sections.

4.1 Interoperability

A local ME implies the usage of a readily available pubsub resource. On the various OS platforms (*e.g.,* macOS, Windows, Linux, etc.), there is usually a platform specific messaging system to do inter-process communication where a pubsub system can be utilized *e.g.,* NSNotificationCenter on macOS. Participants are usually processes running on the platform. If a platform specific messaging system is not available, any other available pubsub system can be used, as long as all participants that are to be local to a ME, know how to access the system. For example, it is possible to use an instance of pubsub running on an external server, as long as all participants that are local to each other use that instance. If more than one pubsub system is utilized on a platform, they must be interconnected by a pair of reflectors, if they are to interact. Participants can be both publishers and/or subscribers in an ME. The pubsub system in an ME serves no other purpose than to relay published messages by broadcasting them to subscribers.

Because the pubsub system is of the broadcast type, messages in IPSME are required to be idempotent or be identified as duplicates so that they can be eliminated, to promote asynchronous communication by reducing the amount of acknowledges required. Idempotent messages can be processed multiple times by a participant, but the execution of each message must give the same result after the application of the initial message.

IPSME does not specify any limitations on message content. Participants speak their own protocol and only those participants that understand the message will be able to process it *i.e.,* partitioning the semantic and syntactic space into any number of separate spaces. This means adhering to the IPSME conventions is minimally invasive for existing systems, since they can continue to speak the protocol that was previously implemented, but over a IPSME dataspace instead. A vital part of IPSME is that if a participant doesn't understand a message, it simply drops the message and continues processing. This can lead to scenarios where certain participants might want affirmation that another participant has received the message. It is possible to send acknowledge messages through IPSME, but how such an acknowledgment is defined is beyond the IPSME specification.

4.2 Semantic Mapping

If each participant speaks their own protocol, communication is limited by the number of participants that understand the sent messages. To broaden the set of participants that understand a message, specialized participants that translate messages can be inserted in to an ME. These translators listen for messages that adhere to a certain protocol, translate them to a different protocol and send out the translation. This means that a translating participant can be the mediator between various participants. It is possible to have a participant that understands multiple protocols *i.e.,* a translator is built into the participant.

After some initial communication, participants can negotiate to communicate directly rather than through the ME. It is the responsibility of the participants to organize such a communication *i.e.,* it is beyond the IPSME specification. Direct communication can still make use of a translator *i.e.,* the translator can also be communicated with directly.

Any authentication or message security is up to the authors of participants *i.e.,* security is left as peripheral to this discussion [Uzunov 2016], see future work in Section 6. If standard and/or central authentication services are required, such a service can be provided through the use of a translator.

4.3 Scalability

Expecting all prospective participants to be connected to the same ME is impractical; not all prospective participants would easily route to a single ME and a single ME would certainly be overloaded. To avoid a centralized architecture and provide scalability, IPSME specifies communication across ME boundaries *i.e.,* cross ME communication. IPSME conventions specify participants should connect to a local ME, but places no limitation on the number of MEs or the 'topology organization' [Uzunov 2016] thereof. If participants in two different MEs want to communicate, such communication is achieved through a pair of reflectors (proxies), one proxy in each of the MEs.

A proxy object listens and filters for particular messages that should be routed to another ME. The proxy object communicates directly with the proxy object in the other ME (its counterpart). How two reflectors communicate is left undefined and is completely up to the author of those objects. The counterpart then publishes received messages to its local ME; participants of that local ME will receive messages from a distant ME transparently *i.e.,* without being aware of any communication complexity thereof. A reply message is routed back through the reflectors in a reverse fashion. Similar to how adding translating participants adds functionality without changing the existing system, reflectors can be inserted into an ME without changing existing objects; the ME is simply expanded.

5 An Interoperable Scalable Solution

5.1 Demonstration

To demonstrate the interoperability and semantic mapping of IPSME, two computer video games, Doom3 and Minecraft, were integrated. The source code of each computer game was available, each in a different programming language. The ability to teleport from Doom3 to Minecraft, and *vice versa*, was implemented through indirect communication using the IPSME conventions. Semantic mapping was used to resolve the schema heterogeneity of each game's player profile allowing for items to be translated from one game world to the other.

5.2 Evaluation

IPSME does not specify any limitations on message content (no predefined structure or data model) *i.e.,* it is possible to use strings, binary or any of the topic-, content- or type-based publish/subscribe schemes, if so required. By not specifying message content IPSME avoids having a predetermined 'expressiveness' [Carzaniga, Rosenblum, and Wolf 2001], perhaps having many simultaneously. Eugster et al. [2003] states platform interoperability is enforced by having strings as keys in messages. In contrast (similar to the flexibility granted by XML [Halevy 2005], where uninterpretable parts of the format can be ignored), the authors argue that one of the main tenets that enables interoperability for IPSME is that receiving participants are able to drop any uninterpretable message. Any attempt to force all participants to adhere to a particular message scheme violates the IPSME conventions.

Since IPSME does not put a limitation on message content, there is nothing impeding the use of standardized protocols *i.e.,* messages can be as custom or standard as required. If a particular participant does not speak a required standard a translating participant can be inserted in the dataspace mediating communication to the required standard *i.e.,* translating participants provide for semantic mapping. If communicating over a pubsub is problematic for a protocol, then initial contact can be made via an ME, followed by direct contact between participants. Resolving schema heterogeneity through semantic mapping is an open topic of research [Halevy

Figure 1: Demonstration of the Doom3/Minecraft integration[1]

2005]; IPSME alleviates the problem through its decomposition. For each source, there can be zero or more translating participants each with a different semantic mapping of a schema in the dataspace. The translators can collectively divide the problem vertically, horizontally [Uzunov 2016] or even incrementally. Because messages in the dataspace are broadcast using pubsub, multiple participants can answer a sender's request. This does add more complexity to the sender, since the sender must handle multiple responses *e.g.,* pick the best response or use all the responses.

Through the use of pubsub, IPSME decouples the production and consumption of messages so as to increase scalability, by removing all dependencies (*i.e.,* time, space or synchronization) between interacting participants. A single ME takes advantage of a centralized architecture to allow for message asynchrony *i.e.,* decoupling synchronization. But, a centralized architecture (*i.e.,* a bottleneck and single point of failure) or even hierarchical topology (*i.e.,* with possible performance problems) [Carzaniga, Rosenblum, and Wolf 2001] is avoided so as to promote scalability. A centralized architecture is avoided by using cross ME communication, allowing for a general graph topology of MEs. IPSME is not fixed to specific properties such as expressiveness or those related to quality of service [Eugster et al. 2003], that affect the scalability of an architecture.

With respect to interconnecting virtual worlds, the dataspace determined by the IPSME conventions resembles that of dividing participants into 'regions' [Nevelsteen, Kanter, and Rahmani 2016; Nevelsteen 2018] using multiple MEs and 'interest management' [Nevelsteen, Kanter, and Rahmani 2016] via each participant understanding and selecting the messages to process. In the domain of wireless sensor and actuator networks, the construct of 'clustering' [Nevelsteen, Kanter, and Rahmani 2016] also allows for abstract regions, where 'cluster heads' act as decentralized authorities for each cluster'. A single ME can be consider similar to such a cluster head, but with its only authoritative function being the relaying of messages to participants.

[1]Video of the demonstration: `https://www.youtube.com/watch?v=knKZd15rhJE`

6 Conclusion

A set of conventions has been introduced, under the name IPSME, so as to facilitate the communication of highly disparate systems. IPSME avoids the requirement of standardization or that all systems must accept the same SDK as a dependency. IPSME has been designed to be minimally invasive to existing systems; a system can continue to speak its implemented protocol within the IPSME dataspace. To help resolve schema heterogeneity through semantic mapping IPSME allows for decomposition of the problem either vertically, horizontally or even incrementally.

If IPSME is to be accepted as a wide-area solution, scalability as well as interoperability is required. IPSME avoids a centralized architecture or hierarchical topology, and scalability is not fixed to specific properties such as expressiveness or those related to quality of service.

The intent of IPSME is to be broadly accepted for the integration of highly disparate systems, but there are environments which have specialized requirements *e.g.,* various security mechanism for the medical field. The set of conventions defined here as IPSME is only the primary layer of a multilayered system; additional layers (*e.g.,* to address service discovery and security) will be presented in subsequent publications.

References

Barthell, Edward N et al. (2004). "Disparate Systems, Disparate Data: Integration, Interfaces, and Standards in Emergency Medicine Information Technology." In: *Academic Emergency Medicine* 11.11, pp. 1142–1148. DOI: 10.1197/j.aem.2004.08.008.

Branton, Chris, Doris Carver, and Brygg Ullmer (2011). "Interoperability standards for pervasive games." In: *Proceedings of the 1st International Workshop on Games and Software Engineering.* New York, NY, USA: ACM, pp. 40–43. DOI: 10.1145/1984674.1984689.

Carzaniga, Antonio, David S Rosenblum, and Alexander L Wolf (2001). "Design and Evaluation of a Wide-Area Event Notification Service." In: *ACM Transactions on Computer Systems (TOCS)* 19.3, pp. 332–383. DOI: 10.1145/380749.380767.

Dionisio, John David n., William G. Burns III, and Richard Gilbert (2013). "3D Virtual Worlds and the Metaverse: Current Status and Future Possibilities." In: *ACM Computing Surveys* 45.3, 34:1 –34:38. DOI: 10.1145/2480741.2480751.

Eugster, Patrick Th. et al. (2003). "The Many Faces of Publish/Subscribe." In: *ACM Computing Surveys (CSUR)* 35.2, pp. 114–131. DOI: 10.1145/857076.857078.

Frey, Davide et al. (2008). "Solipsis: A decentralized architecture for virtual environments." In: *1st International Workshop on Massively Multiuser Virtual Environments*, pp. 29–33. URL: https://hal.inria.fr/inria-00337057.

Halevy, Alon (2005). "Why Your Data Won't Mix." In: *Queue* 3.8, pp. 50–58. DOI: 10.1145/1103822.1103836.

Landman, Adam B. et al. (2011). "An Open, Interoperable, and Scalable Prehospital Information Technology Network Architecture." In: *Prehospital Emergency Care* 15.2, pp. 149–157. DOI: 10.3109/10903127.2010.534235.

Morgan, Graham (2009). "Challenges of online game development: A review." In: *Simulation & Gaming* 40.5, pp. 688–710. DOI: 10.1177/1046878109340295.

Nevelsteen, Kim, Theo Kanter, and Rahim Rahmani (2016). "Comparing Properties of Massively Multiplayer Online Worlds and the Internet of Things." In: *2016 IEEE Symposium on Computers and Communication (ISCC).* Digital Entertainment, Networked Virtual Environments and Creative Technology (DENVECT). Messina, Italy: IEEE, pp. 1–5. DOI: 10.1109/ISCC.2016.7543704.

Nevelsteen, Kim J. L. (2015). *A Survey of Characteristic Engine Features for Technology-Sustained Pervasive Games.* SpringerBriefs in Computer Science. Switzerland: Springer International Publishing. DOI: 10.1007/978-3-319-17632-1.

Nevelsteen, Kim J. L. (2016). "Distributed Technology-Sustained Pervasive Applications." PhD thesis. Stockholm University, Department of Computer and System Sciences. URL: http://urn.kb.se/resolve?urn=urn:nbn:se:su:diva-129151.

— (2018). "Virtual World, Defined from a Technological Perspective, and Applied to Video Games, Mixed Reality and the Metaverse." In: *computer animation & virtual worlds* 29.1, e1752. ISSN: 1546-427X. DOI: 10.1002/cav.1752. URL: http://arxiv.org/abs/1511.08464.

Rehm, Sven-Volker, Lakshmi Goel, and Mattia Crespi (2015). "The Metaverse as Mediator between Technology, Trends, and the Digital Transformation of Society and Business." In: *Journal For Virtual Worlds Research* 8.2. DOI: 10.4101/jvwr.v8i2.7149.

Uzunov, Anton V (2016). "A survey of security solutions for distributed publish/subscribe systems." In: *Computers & Security* 61, pp. 94–129. DOI: 10.1016/j.cose.2016.04.008.

Wang, Chenxi et al. (2002). "Security issues and requirements for Internet-scale publish-subscribe systems." In: *System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on.* IEEE, pp. 3940–3947. DOI: 10.1109/HICSS.2002.994531.

Wehlou, J. Martin (2014). *Rethinking the Electronic Healthcare Record.* Man in the Middle books. ISBN: 978-91-981706-0-3.