



**HAL**  
open science

# Reflections on Identity Management in Smart Industry: The Paradox of Theseus' Ship and Beyond

Hans Wortmann, Wico Mulder

► **To cite this version:**

Hans Wortmann, Wico Mulder. Reflections on Identity Management in Smart Industry: The Paradox of Theseus' Ship and Beyond. IFIP International Conference on Advances in Production Management Systems (APMS), Sep 2016, Iguassu Falls, Brazil. pp.299-306, 10.1007/978-3-319-51133-7\_36 . hal-01615780

**HAL Id: hal-01615780**

**<https://inria.hal.science/hal-01615780v1>**

Submitted on 12 Oct 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# Reflections on Identity Management in Smart Industry: The Paradox of Theseus' Ship and Beyond

Hans Wortmann<sup>1,2</sup> and Wico Mulder <sup>1,2</sup>

<sup>1</sup>University of Groningen, Groningen, The Netherlands

<sup>2</sup>KPN consulting, Groningen, The Netherlands  
j.c.wortmann@rug.nl

**Abstract.** Sustainable digitization of factories (and especially IoT) requires all data related to “things” (products, equipment, but also informational objects) to be stored for later reference in a so-called *digital shadow*. This paper addresses an identity question: “which data should be stored as a so-called digital shadow of an object?” and identifies two opposing principles, Single Source of the Truth vs Local Autonomy. The first principle keeps data objects small and uses links to other data objects. It expands object’s life time. The second principle advises to keep objects large, reduces linkages but limits object’s life time. The paper analyses these two principles for physical objects as well as for related informational objects. Some guidelines for application are given.

**Keywords.** Smart factories · Sustainability · Object life cycle · Digital shadows

## 1 Introduction

The impact of data in the field of smart industry goes well beyond the level of optimizing production. Data is guiding the manufacturing processes. Whilst being designed, manufactured or shipped, the status of every product is kept in systems in the form of data. The same holds for related information that accompanies the product, such as the contracts, version information, ownership, user manuals or maintenance schedules. All of those are reflected by data. Once built, products leave the factory either serving as component of a different product, or starting to operate somewhere in our connected world. In a sustainable, circular economy, data that was collected during manufacturing of a product, will not only be used during the phase of manufacturing and operation, but also during refurbishment and reuse. Moreover, new data will be added. One can say that a modern data management platform<sup>1</sup> maintains information of objects during their full life cycle – from cradle to grave. However, such view is not straightforward. When using concepts such as cradle or grave in relation to the physical world of manufacturing, our language is metaphoric. Physical objects are not born in a cradle

---

<sup>1</sup> Data management platforms are computer infrastructures that provide the means for the storage, connectivity and analysis of data

and do not end in a grave: these are anthropomorphic notions. For example, if a component C disappears because it becomes part of an assembly A, should the life cycle of C be considered as ended (“grave”)? In many cases, this question would be denied, because the usage of C may be monitored as part of the assembly A. Accordingly, does the life cycle of C end when the assembly A is discarded and scrapped? Not necessarily, the component C may be refurbished and reused elsewhere. So when does the life cycle of C end? Who responsible for the data? What is implicit in our usage of the term ‘platform’? Maybe the lifecycle of C ends when C has no longer a necessity or purpose, although even this statement may be disputed.

Also, the notion of a connected world is problematic. Loosely speaking, it means that objects (representing e.g. things and humans) are ubiquitously connected to the internet. However, the implicit meaning is that objects have references to other objects’ data and that these can always be retrieved when needed. For example, manufacturing orders refer to products, to operations, to machines, and many other objects. Still, the assumption that such references can always be retrieved, is problematic. Objects are owned by applications and these applications have a finite life cycle. More fundamentally, relations between objects are temporary and references should therefore be time stamped. If component C is manufactured by supplier S according to manufacturing instructions I, how can we be sure that the right version of these instructions I are retrieved? In many cases, we cannot be sure, unless these instructions are stored together in a data management platform in the form of a so-called digital shadow (which we will explain in the next section). Accordingly, related to concepts such as smart manufacturing and the associated developments such as the internet of things (IoT) and cyber-physical systems (CPS), there are two questions which seem fundamental to be elaborated: (i) What is the life cycle of an object? Or more precisely: what are the criteria to create a digital shadow of an object or delete it? (ii) What is the boundary of an object? Or more precisely: what are the criteria to include related objects and their data in the digital shadow of an object?

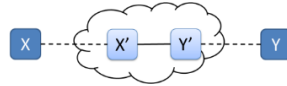
## 2 Terminology Background

Before discussing the above questions, it is worthwhile to present some working definitions. In the field of software development people use the word object to model elements from the real world into their design. We would like to use more conceptual but more precise notions, namely *digital shadows* and avatars. The Definition of A *digital shadow* is a digital representation of an object that manifests itself in the world around us.

A digital shadow refers to an entity inside a digital environment, e.g. in a database, that contains all the data of one particular object which exists in our real world. A digital shadow is tightly coupled to that physical object. In the context of internet of things, the digital shadow holds an ip-address, which allows for interaction with other objects during the phase of creation as well as during its operation.

As an example, one can think of any object, e.g. a bus, a plane or a cow, or one of their constituting components. Whenever we want to store or retrieve information from

such an object, we do this via its digital shadow. Anyone who wants to have information about a particular thing or product X can do so via its digital shadow X'. This also holds for any object Y which needs to exchange (retrieve or store) data with object X. As depicted in figure one, there is one important constraint to take in mind; namely that Y cannot interact with X' directly, but only via its own digital shadow Y'.



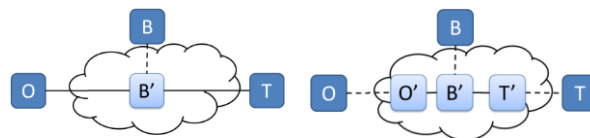
**Fig. 1.** Schematic picture of objects that exist in our physical world (denoted with X and Y) Together with their digital shadows (denoted with X' and Y'). The dashed lines show that X and Y are connected with their digital shadows. X and Y can communicate with each other via their digital shadows, as denoted with the solid line between X' and Y'.

Thus, there is an important design principle for digital shadows: it is not allowed to have a direct connection between an object Y and a digital shadow X'. As an example think of a bus B having a digital shadow B' and an organization O (owning the bus) that wants to have access to the maintenance schedule of this particular bus. Suppose also that a traveler T (taking that bus) is interested in information about the actual time schedule of that particular bus.

As we see in many contemporary IT architectures, B' will be accessed by both O and T. Such IT architectures require multiple interfaces and something like a role-based access system. But as the number of interactions increase (which will happen given the evolution of internet of things) maintenance and flexibility reasons imply to automate the management of those interfaces.

A possible way to do this is to make use of the concept of digital shadows and let the interaction with B' take place via two other digital shadows O' and T'. This allows for more flexibility from an IT maintenance perspective, and possibilities for time-dependent, automated managed interfaces.

An example in the area of smart factories could be the manufacturing of constituents of an airplane which are manufactured by multiple organizations. During the phase of creation a digital shadow of those constituents serves as a digital product dossier. Many vendors, customers, shipping companies can have access to this dossier. Also machines can access the products bill of materials via this digital dossier. Later on, when the plane is in operation, information about maintenance schedules may be used by various groups of users. E.g. flight carrier organizations or insurance companies, passengers that following that particular plane.



**Fig. 2.** Schematic picture stressing the design principle of digital shadows

It's a small step here to think of complete mimicked worlds of goods, plants and organizations that manifest themselves as a complete universe within a cloud environment. So far, we talk about a digital shadow as a passive set of data inside a platform that can provide access by means of a (web) interface. Now, if we extend this concept to an active entity, that is something that can be programmed to perform tasks we could use notion of avatar<sup>2</sup>. Avatars are quite similar to the concept of agents [3,4], but with one important restriction: they are uniquely connected to a (unique) object in our real world. The definition of an *avatar* is an active digital representation of an object that manifests itself in our physical world.

Avatars can be useful in situations where information about status and quality needs to be exchanged frequently or under certain restricted conditions. They play a role in local decision making and in the exchange of information with various stakeholders inside and across organizational boundaries, e.g. in production chains or when interacting in an Internet of Things context. Avatars can also be useful for simulation purposes and the analysis of what-if scenarios in situations where information is only locally available. A data management platform, as we mentioned in the previous section, can be regarded as a habitat for avatars.

### 3 Theseus' Ship and Beyond

#### 3.1 Physical Objects

With these two concepts in mind, it is possible to elaborate on the questions mentioned in the introduction of this paper. The first question, related to the life cycle and identity of (physical) objects raised interest from philosophers in ancient times. In the form of Theseus Ship paradox, it was introduced by Plutarch in the 1st century, and it remained in philosophers interest until modern times. However, a related question was earlier posed by Heraclitus (6th century BC), who discussed the possibility to enter twice in the same river. Theseus Ship paradox goes as follows:

“The ship wherein Theseus and the youth of Athens returned from Crete had thirty oars, and was preserved by the Athenians down even to the time of Demetrius Phalereus. They took away the old planks as they decayed, putting in new and stronger timber in their places, in so much that this ship became a standing example among the philosophers, for the logical question of things that grow; one side holding that the ship remained the same, and the other contending that it was not the same [1]. Plutarch thus questions whether the ship would remain the same if it were entirely replaced, piece by piece” [2].

In modern times, the same issue emerges for our industrial society. Basically the same question emerges for many industrial products, viz. when to introduce a new identification for changing products. For reasons of conciseness, we describe here the solution adopted for standard assembled products, known as the form, fit, function rule. In many industries the so called Form-Fit-Function rule is applied to the problem at hand

---

<sup>2</sup> The concept of avatars is known in the field of computer gaming, but the name of the concept originates from ancient Hinduism beliefs in reincarnated bodies.

[5]. This rule states that companies will define a new item replacing an old one, if: (i) The item has a new form and/or; (ii) The item has a new fit (e.g. new tolerances) and/or, (iii) The item has a new function.

Therefore, new raw material to be used for an item, for example, is in itself not a reason for defining a new item. Also, if the item would get a new price, if it were stored in another warehouse, or if documentation of the item would be changed, this is not a reason for defining a new item or for issuing a new item code number.

But how can companies then keep track of significant changes that do not change the form, fit or function? For example, how can companies reflect the fact that the weight or the carbon footprint of a product has become much less due to the use of other materials? For these purposes, the notion of versions exist in manufacturing industries. The FFF-rule goes together with a versioning policy. This policy states, that items can have versions, and that a new version will be issued for all changes in the item documentation, which don't violate the FFF-conditions.

If something changes in the definition of an item O inside a product P, but there is no new form, fit, or function regarding P, then there is no reason to define an entirely new item of product P. In this case, a new version of O will be created.

### **3.2 Informational Objects**

The notion of versions is crucial in the FFF-rule. However, what is a version? First of all, it should be acknowledged that the FFF-rule does not refer really to physical products, as we are tempted to believe, but to so-called informational objects. Informational objects are objects that describe physical objects. Versions are no exception. Versions do not refer to physical instances, but to the informational description of these instances.

The question, what a version really is, can now be answered. A version is a new (or updated) description of a standard physical product. What makes a version different from other informational objects? Obviously, the fact that a (new) version refers to a previous version, and carries forward some attribute values or other informational elements (e.g. methods) from the previous version. Because the FFF-rule stems from the pre-digital era, it is not exactly defined which informational elements are carried forward.

## **4 Single-version-of-the-truth vs Self-containedness**

In practical discussions about smart manufacturing, there are two principles which can be encountered. These principles are also underlying much academic work, although these principles are (to our best knowledge) not elaborated in the context of smart manufacturing. These principles are: (i) The principle of *single version of the truth* which states that data should always be retrieved from their source. Data kept by the digital shadow is often a reference to other digital shadows; (ii) The principle of *autonomy or self-containedness*. This principle states that all data describing an object should reside in its digital shadow or under control of its avatar.

The principle of single-version-of-the-truth tries to avoid redundancy, because redundant data are difficult to maintain: there are always updates which get lost. Therefore, this principle states that copying data or documents should be avoided. Data in e.g. digital shadows of objects should refer to data in other objects whenever such sources exist. When applied to versions of physical products, the principle states that a new version should contain only information in which the new version differs from the old version and all other information should be retrieved from the older version. When applied to components that are part of assemblies, the principle states that digital shadows of assemblies should not contain the information about the components, but rather refer to the digital shadows of these components.

For versions of software products the principle of single-version-of-the-truth leads to software architectures based on Software-as-a-Service (SaaS) principles. Therefore, the principle assumes that ownership of data is well organized and remains available over a long period of time. Accordingly, informational objects should have a long life cycle in this view, and hardly ever be deleted, because there may always remain links which require information from these objects. Also, the platform on which avatars or digital shadows reside, are assumed to have a long life cycle.

The principle of autonomy aims at robustness. Physical objects should have digital counterparts which are self-contained and proactive. Physical objects as components, parcels, tools, or machines should be enabled to take initiative, communicate, negotiate and get decisions taken. Advocates of the principle of autonomy are not primarily interested in avoiding redundancy. They do not argue against copying of data. They acknowledge the fact that objects may need updates from other objects, but they claim that e.g. publish-and-subscribe protocols are sufficient means for this purpose. They also do not have SaaS architectures in mind. Rather, they see smart manufacturing as consisting of decision-making applications where avatars interact with each other. In addition, they see analytical applications, where digital shadows of objects of the same type are analyzed by human professionals or by artificial intelligence applications. These applications typically take a broad view on the objects analyzed, and therefore many data are taken into the digital shadow.

Because the necessary data are carried in the digital shadow of the object, there is less interest in the longer life time of objects. Digital shadows and other digital objects are seen as created for a certain purpose, and when the purpose vanishes, the digital object or the digital shadow may vanish as well.

## **5 Discussion**

### **5.1 Digital Assemblies**

It is interesting to note that advocates of both principles see information objects largely as being constructed from other information objects. The term digital assembly can be coined for this process. This leads to the question, if the Theseus ship paradox is also applicable for digital assemblies. The answer is yes, there is an issue with the identity of digital assemblies, but there is also an important difference: if a digital component is taken into a digital assembly, the original component does not disappear. It can either

be copied into the new object, according to the principle of autonomy or it can be referenced in the new object, according to the principle of single-version-of-the-truth.

It is also interesting that advocates of both principles refer to earlier experience with integration problems. Advocates of the first principle are well aware of the difficulties of heterogeneous systems and the advantages of open standards, web services and common databases. They claim that redundancy should be avoided. Advocates of the second principle are also well aware of integration problems. They claim that integration should be avoided.

## **5.2 Ownership**

It is obvious that the advocates of the first principle (single-source-of-the-truth) must assume that the smart manufacturing organization has implemented proper ownership of digital shadows and of other digital documents. If data are not properly maintained, or if there is no ownership of data management platforms, the smart factory will not survive. This is not only required for the own organization, but also for partner organizations in New Product Development, in Supply Chain, in Distribution, in the product use phase and in recycling or refurbishment.

However, also the advocates of the second principle (autonomy) must assume that their digital shadows are informed if related objects change. For example, if an assembly is dismantled, the components must re-appear. If new components failure modes are detected, the assembly digital shadow must become aware. As said, this will be implemented in autonomous systems by e.g. publish-and-subscribe protocols. However, these protocols also assume that ownership for platforms, applications and data is well established in the smart manufacturing organizations and with related stakeholders.

To be more precise, if a new digital object is made with its own life cycle, there should always be an owner. This owner either relies on references to existing data (e.g. for versions, following the first principle) or relies on copies (following the second principle), but ownership which guarantees the quality of the data cannot be avoided in smart manufacturing. In the end, the owner determines the solution chosen for the Theusis paradox in the case of digital documents.

Ownership means, that a party in the real world takes responsibility for the correctness of data ([8],[9]). The difference between the two principles resides in the nature of this responsibility. With single-source-of-the-truth, updates and changes are carried forward. With local autonomy, the data are frozen at the moment of creation. It is interesting that here the notion of shared ownership of frozen data becomes possible due to blockchain technology, a distributed public ledger of all transactions or digital events that have been executed and shared among participating parties [7].

## **5.3 Can Digital Objects Have Digital Shadows and Avatars?**

Initially, our discussion has been primarily focusing on the situation where there is a digital shadow and/or avatar which represents a physical product. However, the example of the FFF rule showed, that there are important informational objects in the digital



world which may claim a similar status as physical objects. For example, an invoice may take initiative to be paid, a contract may urge to be signed. There is no reason to restrict the notion of avatar to representatives of physical objects.

However, the digital shadow of a digital object may technically be realized in the same environment as the digital object itself. Conceptually, however, there is no reason to make a distinction between physical and informational objects in this respect.

## 6 Conclusion

In this paper, we posed two questions, related to sustainable smart manufacturing. As for the first question, we concluded that ownership of data objects is crucial in (smart) manufacturing. The life cycle of physical objects is ultimately related to a company's strategy in circular economy (including partnerships) but this can only be realized with properly assigned responsibility for data.

As for the life cycle of informational objects, our analysis revealed that informational objects are usually assemblies of other informational elements. The paper discussed two principles: (i) The principle of *single-source-of-the-truth*, which tends to keep longer life cycles and to keep distinct objects, leading to SaaS architectures; (ii) The principle of *autonomous objects*, which tends to have shorter object life cycles and allows overlapping objects, leading to Data Warehousing architectures.

These principles also provide conflicting guidelines for the second question. The first principle would create objects as small as possible with references to other objects, whereas the second principle would do the opposite. In practice, one sees combinations of these principles, still a guideline is to apply the first principle for objects which are used in applications that require real-time relational flexibility, whereas the second principle can be used for objects in applications that require robustness or historical compliancy.

## References

1. Plutarch: "Theseus". The Internet Classics Archive (2008)
2. Wikipedia, [https://en.wikipedia.org/wiki/Ship\\_of\\_Theseus](https://en.wikipedia.org/wiki/Ship_of_Theseus)
3. Jennings, N.R., Bussmann, S.: Agent-based Control Systems: Why Are They Suited to Engineering Complex Systems? IEEE Control Systems Magazine, 23(3), 61-73 (1999)
4. Wooldridge, M.: MultiAgent Systems. John Wiley and Sons (2002)
5. Romero Rojo, F. J., Roy, R. and Shehab, E.: Obsolescence Management for Long-life Contracts. Int. J. Advanced Manufacturing Technology, 49(9-12), 1235-1250 (2009)
6. Erkoyuncu, J., González Muñoz, R., Shehab, E., Weinitzke, M., Bence, R., Fowler, C., Tohill, S. and Baguley, P.: Understanding the Life Cycle Implications of Manufacturing. Procedia CIRP, 37, 24-29 (2015)
7. Zuidam, R van: Whitepaper Government as a Service (Dutch), <http://www.dutchchain.nl>
8. Loshin, D.: Who Owns Information? In Enterprise Knowledge Management: The Data Quality Approach. Morgan Kaufmann (2001)
9. Wende, K. A Model for Data Governance – Organising Accountabilities for Data Quality Management. In 18th Australasian Conference on Information Systems, 417–425 (2007)