# A Non-concept is Not a ¬Concept

Iaakov Exman
*Software Engineering Dept., The Jerusalem College of Engineering, Jerusalem, Israel*

Keywords: *Non-*, Concepts, Identity, Parts, Functionality, Pluggable Ontologies, Abstract Sockets.

Abstract: Often objects with removed parts or functionality also lose their identity. But, there are situations in which this is not the case: identity is preserved. We refer to such objects, by means of *Non-* concepts, *non-* implying partial negation and *concept* implying preserved identity. In this work *Non-* concepts are defined and pluggable ontologies are proposed for their representation. These ontologies are made pluggable by sockets, a novel kind of class. These are abstract place-holders for removed/added parts, functionalities or identities. The space of *Non-* concepts has been extensively explored. Pragmatic implications of *Non-* concepts include manageable design of products with a multitude of models. *Non-* concepts are also relevant to the formal controversy whether composition is/isn't identity. The resolution is not sharp. Identity is entangled with composition, such that identity is preserved to a certain extent, until further removal causes identity break-down.

## 1 INTRODUCTION

When an object is stripped of some of its component parts or loses functionality it reaches a point where it is not anymore recognized as such a kind of object: besides its utility, it loses its identity. But, there are intriguing situations in which a loss of functionality or parts does not lead to loss of identity.

We coin a concept assigned to an object in such situation a "*Non-*" concept. It is both *Non-* as it has lost some of its characteristics, and it is a *concept* as it is still easily recognizable as such.

This paper characterizes Non- concepts, proposes an ontological representation, explores the space of possible *Non-* concepts and deals with its pragmatic and philosophical implications.

### 1.1 Related Work

Identity is a widely discussed issue in the literature e.g. (Kripke, 1977). Literature relevant to this work relates identity to composition.

There are two roughly opposing positions with this respect. For one side identity *is* composition of parts. Some representative examples are e.g. (Lewis, 1993), (Merricks, 1999) and (Liao, 2005).

Lewis states that the opposite of identity is not non-identity, but distinctness in the sense of overlap, things with parts in common (Lewis, 1993) page 33. This is a suitable starting point for this work.

For the other camp a set of variations on the composition *is not* identity. See e.g. (Baker, 1997), (Elder, 2008). Inquiring deeper one finds that both camps have more in common than acknowledged.

Systems' functionality or behavior has been less under the focus of conceptual approaches.

A gentle introduction to formal ontologies – used in this work to represent Non- concepts – can be found in e.g. (Guarino, 1998).

Modular ontologies – composed of sub-ontologies – have been proposed and extensively discussed. A few representative pointers include (Rector, 2008) and (Schlicht, 2008).

Non- concepts do not imply malfunction, defective or broken objects, or incomplete and/or inconsistent ontologies. Design problems leading to incomplete and/or inconsistent ontologies have been dealt with in the literature, e.g. (Baumeister, 2005).

## 2 *NON*-CONCEPTS

Although dictionary-wise "non-" is a prefix indicating negation, we shall use it here as a noun, with the specific meaning of a kind of concept.

### 2.1 The *Non*-clock Example

The author of this paper has a non-clock hanging on

Exman I..
A Non-concept is Not a Concept.
DOI: 10.5220/0004149704010404
In *Proceedings of the International Conference on Knowledge Engineering and Ontology Development* (KEOD-2012), pages 401-404
ISBN: 978-989-8565-30-3

a wall in the kitchen. It is seen in Fig. 1. It is used to illustrate the idea of non-clock for guests. It can't be used to measure time.

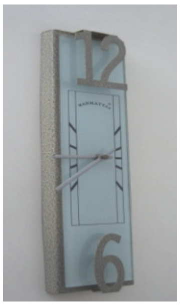


Figure 1: Photo of a Non-Clock – A non-clock as a concrete instance of its *Non-* concept. It has a visible scale – the numbers 6, 12, and marks for other hours. It is synchronizable by a mechanism in its back. It lacks periodicity since its battery was removed. Its identity is clearly recognized, but it is not useful for measuring time.

We have characterized (Exman, 2010) a clock as a device to measure time with three properties:

1. *Periodicity* - it has a periodic behavior, based upon a physical phenomenon;
2. *Adjustability* – it has a pre-defined scale of numbers, to which events are assigned;
3. *Synchronization* – it may send/receive messages, to synchronize with other clocks.

The non-clock in Fig. 1 has a scale as clearly seen: the numbers 6 and 12 and marks for other hours. It may be synchronized and is adjustable, since one can rotate the non-clock hands to any desired value in the scale, by a mechanism in its back.

It does not have periodicity, since its battery has been removed. Thus, it cannot be used to measure time. It lacks both a component part and its correspondent functionality.

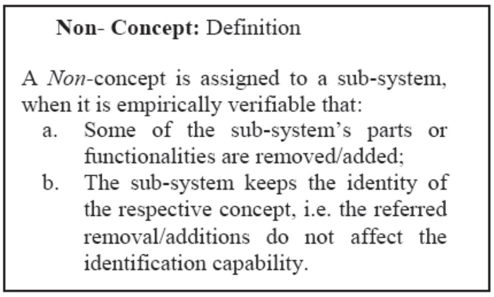
Nonetheless, one easily recognizes its identity. Ask any guest – what is hanging on the wall? – and one easily gets a "clock" reply. It takes some time to explain that it is not a real clock.

The object in Fig. 1 is not a ¬clock, where ¬ is the logical not sign. The referred object is not the complement of a clock in any chosen universe of objects. The very fact of its recognition implies that it is much closer to be a clock than whatever may be its complementary ¬clock.

## 2.2 *Non-* Concepts Defined

We define a *Non-* concept as follows.

Non- concepts do not refer gradual change. We mean discrete removal/addition of parts or functionalities, leading to a distinct entity of a new kind. A non- concept is neither a concept, nor a ¬concept.



**Non- Concept:** Definition

A *Non*-concept is assigned to a sub-system, when it is empirically verifiable that:

a. Some of the sub-system's parts or functionalities are removed/added;
b. The sub-system keeps the identity of the respective concept, i.e. the referred removal/additions do not affect the identification capability.

One can remove/add parts without affecting identification. In fact, there exist products explicitly designed to allow such removal/addition. Nevertheless there are essential parts that once removed prevent identification of the original object.

Functionalities are quite similar to component parts. Removal/addition of functionality does not necessarily prevent identification.

In the above definition there are four elementary undefined concepts: a) *Identity* – there may be several identities of an object, but there is a single identity in a given context; b) *Part* – a discrete structural component of the sub-system that may be added or removed; c) *Functionality* – a discrete behavior of the sub-system, associated with one or more of its parts; d) *Non-* - a noun serving as a kind of identity of a concept.

## 3 PLUGGABLE ONTOLOGIES

Here we propose sockets, a novel kind of class, to be added to ontologies in order to represent *Non-*concepts. Such ontologies are said to be pluggable, i.e. parts or functionalities may be plugged-in or out.

Sockets solve the following problem: -How to fully represent a sub-system's *Non-* concept displaying removable parts which have been actually removed/added?

### 3.1 Sockets

*Socket* is an abstract generic place-holder for any of the above concepts: identity, part, functionality. It allows dealing in a neat way with identities, removal and addition of sub-system parts and functionalities. Sockets are used as properties of classes. A Socket is itself a class. Each socket has one or more "plugged-in" properties, whose respective values are the respective pluggable part or functionality.

A plugged-in property has a cardinality

restriction whose value is Boolean. A part or functionality is either plugged-in with cardinality value 1, or not plugged-in, with value 0.

A sub-system should have more sockets than the sum of removable identities, parts or functionalities.

## 3.2 iSockets

An *iSocket*, standing for identity socket, is a sub-class of socket, specialized for *identity* removal/addition. The cardinality of the plugged-in property of an iSocket is omitted, as it is always 1.

*Non-* is only used as a value of the plugged-in property of iSockets. There may be only a single *Non-* in the iSockets of an object. This is different from the logical not sign ¬ which can be added to each proposition, thus appear several times in the description of a single object.

## 3.3 Pluggable Ontology Examples

We start with the non-clock of sub-section 2.1. Its battery was removed, thus it has no periodicity. Since the adjustability and synchronization were not modified, they are not represented. The non-clock pluggable ontology is in Fig. 3.

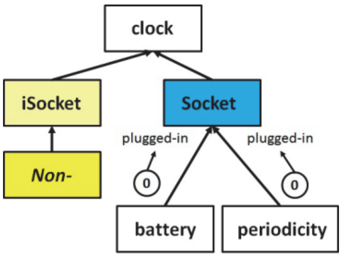


Figure 3: Non-clock partial pluggable ontology – It has one iSocket with a *Non-* value. It is a non-clock as its battery was removed: it lost its periodicity functionality. Cardinality values of the plugged-in properties are 0, as the respective part and functionality were removed.

Let us do a thought experiment. Suppose we add a new battery to our non-clock. We then synchronize and adjust the time shown to be the correct current time. So, now it is just a fine functioning clock.

Next we put an internet video camera in front of the revived clock. The image of the moving clock is transmitted through the internet, and seen in another computer – in a different country.

The image of clock through the internet is now an Internet-Video non-clock. It has a scale and periodicity. Its identity is easily recognized as an instrument to measure time and can be used to do so.

But the video itself cannot be synchronized. So, by the demand of the three properties above it is not a plain clock. Nonetheless it is a useful non-clock, as long as the actual clock which is the video image source works well. Its partial pluggable ontology is seen in Fig. 4. It differs from the previous ontology by a plugged-in addition.

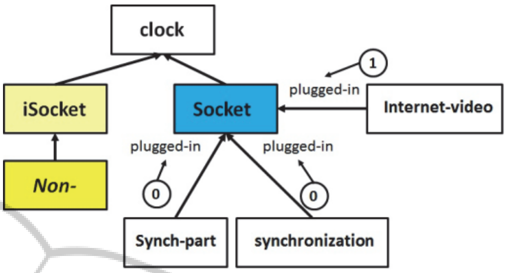


Figure 4: Internet-Video Non-clock partial pluggable ontology – It is a non-clock as we see its video through the internet: it lost its synchronization functionality. The cardinality values of the plugged-in synch-part and synchronization functionality are 0. On the other hand, an internet-video property has been added with cardinality 1.

Next, we do a second thought experiment. We keep the internet video camera, but again remove the clock's battery. The image now is static. This new non-clock is not very useful. It certainly has a scale, but no periodicity and no synchronization ability.

# 4 THE NON-CONCEPTS SPACE

## 4.1 *Non*-concepts by Design

There are products a priori designed to fit *Non-*concepts: a) *lacking parts*; b) *downgraded components* (see Fig. 5); c) *lacking functionality*.

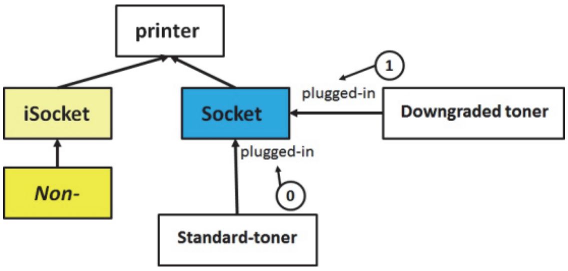


Figure 5: Non-printer with downgraded toner partial pluggable ontology – It is a non-printer since the standard toner was removed – plugged-in cardinality=0. It is sold with downgraded toner – plugged-in cardinality=1.

## 4.2 *Non*-concepts, Obsolescence and their Cemeteries

The most widespread object cemeteries are car cemeteries, see Fig. 6. They are impressive and have been the subject of literary works and a theatre play.

Figure 6: Yellow non-car in Car cemetery photo – All the cars in a cemetery are identifiable as such. Although one cannot tell that the yellow one is for sure a non-car, its overall condition leads us to think so.
Photo: Norbert Aepli, published under the license "Creative Commons Attribution 3.0".

A non-car's ontology for a car rescued from the cemetery – say the bright yellow little car – to be a collector's item is shown in Fig. 7.

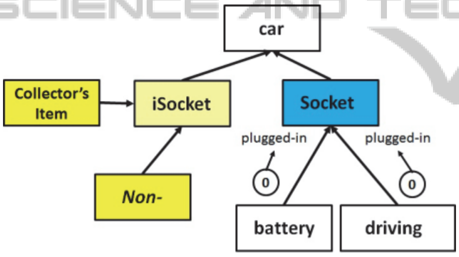


Figure 7: Collector's Non-car partial pluggable ontology – It is a non-car since the driving functionality was removed, perhaps by removing the battery – plugged-in cardinality=0. It serves only as a collector's item – as shown plugged-in in the iSocket.

# 5 DISCUSSION

The pragmatic implications of *non-* concepts refer to design of systems of a few kinds: a) *variety of models*; b) *removable parts*. In such cases, one could use abstract sockets to explicitly manipulate parts with differing status, viz. to label the respective parts along design, manufacturing and delivery stages.

The formulation of Non- concepts and sockets in this work and the examples given lead us to a unique position about whether identity *is*/*isn't* composition.

We say that identity and composition are entangled. To a certain extent, composition changes by parts' removal/addition do not affect identity.

Beyond further removal/addition of parts, identity breaks down. This is not marked by a fixed quantitative limit; it depends on the part types and order of removal/addition.

## 5.1 Future Work

Among the open questions regarding Non- concepts:

Do we need additional accessories to characterize *non-* concepts? While it is satisfactory that with a minimal set of generic classes – Sockets and iSockets – one still needs a more comprehensive investigation to provide a more definitive answer.

Are pluggable ontologies completely equivalent to modular ontologies? Sockets seem to be the natural mechanism to attach ontology modules.

# REFERENCES

Bacon, J., Campbell, K. and Reinhardt, L., 1993. *Ontology, Causality and Mind*, Cambridge University Press, Cambridge, UK.

Baker, L. R., 1997. "Why Constitution is Not Identity", *The Journal of Philosophy*, Vol. 94, (12), pp. 599-621.

Baumeister, J. and Seipel, D., 2005. "Smelly Owls – Design Anomalies in Ontologies", in *Proc. 18th Int. Florida Artificial Intelligence Research Society Conf.*, AAAI Press, pp. 215-220.

Elder, C., 2008. "Against Universal Mereological Composition", Dialectica, Vol. 62, (4), pp. 433-454.

Exman, I., 2010. "Software is Runnable and Composable Ideas – 1. Persistent Systems", in *Proc. SKY'2010 International Workshop on Software Knowledge*, Herzlia, Israel, pp. 29-35.

Guarino, N., 1998. "Formal Ontology and Information Systems", in *Proc. Of FOIS'98*, Amsterdam, IOS Press, pp. 3-15.

Kripke, S., 1977. "Identity and Necessity", pp. 66-101, in Schwartz, S.P. (ed.) *Naming, Necessity and Natural Kinds*, Cornell University Press, Ithaca, NY, USA.

Lewis, D., 1993. "Many, But Almost One", in ref. (Bacon, 1993), pp.23-37.

Liao, Shen-yi, 2005. "Things are Their Parts", Logos, Vol. II, Issue 2, pp. 44-61 (Spring 2005).

Merricks, T., 1999. "Composition as Identity, Mereological Essentialism, and Counterpart Theory", *Australasian Journal of Philosophy*, 77, pp. 192-195.

Rector, A., Horridge, M., Iannone, L. and Drummond, N., 2008. "Use Cases for Building OWL Ontologies as Modules: Localizing, Ontology and Programming Interfaces & Extensions", in *Proc. 4th Int. Conf. SWESE Semantic Web Enabled Software Engineering*.