Get into the rhythm of Metapattern

Pieter Wisse

Metapattern is a method/language for conceptual modeling. A characteristic feature is so-called contextual differentiation. Letting contexts determine differences should not be mistaken as Metapattern's single feature, though.
How Metapattern simultaneously establishes differences as coordinated is particularly novel, and powerful. With global scope of information exchange, there is no escaping accepting as reality both differences of meanings and (their) interdependence.

As an application of discrete mathematics, Metapattern's approach to, say, analytical synthesis is given a formal description in part I of **Metapattern: context and time in information models** (Pieter Wisse, Addison-Wesley, 2001). Because the favored language mode for developing and communicating models is visual, in the book **Metapattern** the formalization is to a large extent expressed visually, too.[1]

It should not come as a surprise that the coordinating aspect of Metapattern is often missed by current practitioners and researchers. So far, modeling for digital information systems and services has been dominated by analysis. A paradigm shift is therefore at stake to balance analysis with synthesis.
As paradigms go, it doesn't help to regret the lack of immediate acceptance. All I can do about it in this case is to continue to try and explain Metapattern.

Let me briefly recapitulate how Metapattern views coordination of differences. Here, please read situation as a synonym of context.[2]

An object, let's call it O_n, is considered a set of mutually disjunct behaviors. A particular behavior occurs in an equally particular situation. A behavior cannot exist without a situation.
For unambiguous modeling of behavior, O_n is differentiated into situated objects. Each situated object, let's call it $o_{n,s}$, is uniquely identified. For coordination, all such situated identities refer to an overall identity, securing that each of an object's situated behaviors can be accessed from any other of its situated behaviors.
How an object's overall identity is positioned should preferably be no exception. For representing 'its' situation, a model is equipped with a horizon.

[1] Most of part I of the book Metapattern has been reproduced by the author as The pattern of metapattern: ontological formalization of context and time for open interconnection (in: **PrimaVera**, working paper 2004-01, Amsterdam University, 2004). For an earlier formalization, see a.o. Multicontextual paradigm for object orientation: a development of information modeling toward fifth behavioral form (in: **Informatiekundige ontwerpleer**, Ten Hagen Stam, 1999, translated from a text originally written in Dutch in 1991). A recent outline is sketched in Open conceptual modeling with Metapattern (2012) which contains further references available (in the English language) up to that moment.

[2] A formal distinction is made through the semiotic ennead; see Semiosis & Sign Exchange; design for a subjective situationism (Pieter Wisse, Information Dynamics, 2002, dissertation defended at Amsterdam University). Enneadically, then, a concept corresponds to a situated object's behavior. And a model does not contain concepts, but (re)presents them and as such also (re)presents situated behaviors.

I repeat that such an overall identity's sole purpose is to act as intermediary between all the object's 'real' situated behaviors. It follows that pertaining to an object's overall identity no (other) behavior is differentiated/specified.[3] An overall identity of/for an object is therefore called: nil identity.[4]
Figure 1 sketches in rough overview what the book **Metapattern** explains in a progression of detailed steps.

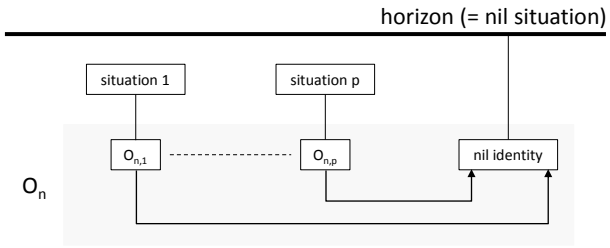

figure 1: coordination of contextual/situational differences through nil identity.

However succinctly described here, too, figure 2 shows how Metapattern directs decomposition of situation. In this case, situation p is taken to consist of situated object $o_{m,q}$ in situation q.
This so-called upward decomposition toward the horizon continues until the model only contains nodes that represent situated objects. In this sense, figure 2 serves to explain, as does figure 1, Metapattern's formalism of informational relativity.

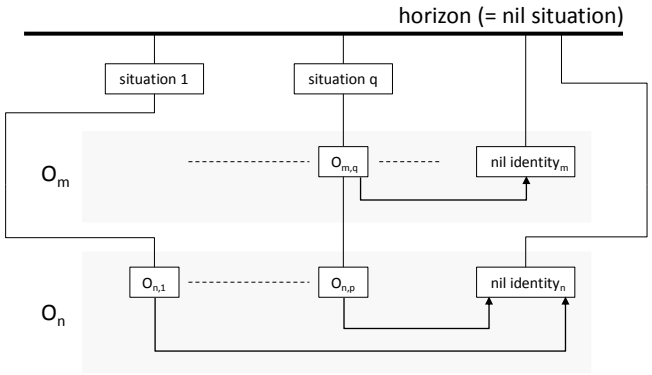

figure 2: conceptual model as network of situated objects.

The original assumption that an object's nil identity as a separate node should always be directly linked to the horizon is actually too strict. For some situated object, any other situated object may be taken as providing it with a nil identity, albeit of a relative nature. The recursive procedure indicated with figure 2 guarantees that eventually the horizon is 'reached.' This is illustrated in figure 3.

[3] The opposite approach, especially counterproductive when aggregate modeling of behavior gets stuck in incommensurability (which is inevitable beyond even a minimal scope of information exchange), is still followed for/with core components. For a critique, see How so-called core components are missing the point (2007).
[4] The horizon is also called nil object which consists of a nil identity, only. Metapattern's boundary conditions are fully specified in the book **Metapattern**. Properly chosen boundary conditions allow for compact, elegant formalism of regular method/language construct(s).

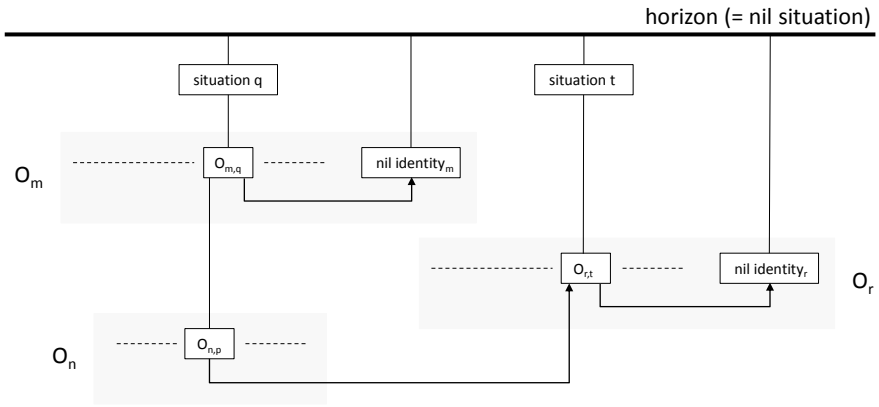

figure 3: relaxing the constraint of absolute nil identity, increasing conceptual relativity.

In the book **Metapattern** it wasn't much emphasized that a situated object need not refer to the object's nil identity taken in an absolute sense (within a horizon, that is). Taking an already situated object for an intermediary nil identity was presented as more or less obvious, calling it "derivation between contexts" and "chains of derivational relationships" (pp. xxvi-xxvii).[5]

Including an explicit nil identity would then only be required for situated objects with no other situational constituent than the horizon. Conceptually, however, separate nodes for nil identities (often) do not really add to understanding, making instead a model unnecessarily elaborate.
In the book **Metapattern** the practice was adopted for actual models, whenever a misunderstanding seems unlikely to occur, to conflate nil identity with a situated object that could be considered most 'primitive' within the horizon. Figure 3 is redrawn as figure 4, replacing nil identity with some, say, basic situated object (with basic relative to the horizon) including nil identity. Situations q and t are left for (upward) decomposition.

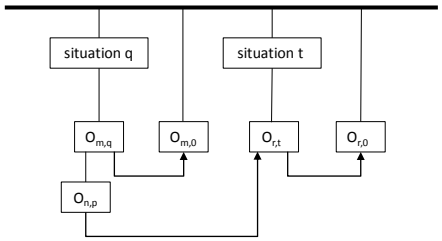

figure 4: hiding nil identity in primitive.

From, as it were,[6] hiding nil identity it follows that in modeling practice, given the right assumptions, all situated objects are differentiated from other situated objects, i.e. objects

[5] Also see Cascading nil nodes in Metapattern (2011).
[6] Nil identity has **not** been eliminated! It simply needs not be shown separately anymore, precisely because it is assumed for each situated object (!) that it refers to a (relative) nil identity.
When this omnipresence of nil identity is not understood in depth, Metapattern's synthetic quality is most certainly missed (and, as a consequence, models will not represent integrated variety).
In retrospect, the importance of nil identity may be already recognized for coordinating legacy systems; see Information strategy for information resources (2000). Adding explicit nil identity facilitates coordinating information sets developed and maintained separately. This concept is labeled information roundabout. Several papers in Dutch explain it, a.o. Stelselmatig overzicht via informatiesleutels (2013; see there for more

resulting from earlier differentiation.[7] Soon after publishing the book **Metapattern** it was found that a somewhat adjusted visual notation better fitted this modeling reality.[8]
In figure 5, part of figure 3/figure 4 is reproduced as figure 5.b. Figure 5.c is conceptually equivalent but adopts the changed notation, in use from 2002 on.[9]

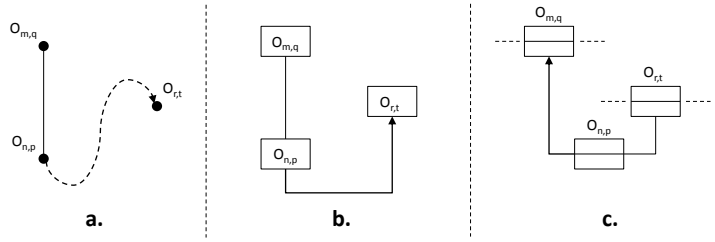

figure 5: optimized visual notation of additional differentiation.

In the book **Metapattern**, the symbol used for regular nodes was actually not rectangular, but a dot; see figure 5.a.[10] Conceptually, of course it also doesn't make any … difference.

For it is not a particular notation, distinctive as it may be, that effectively makes out Metapattern (or whatever method). Other symbols may be used, for example the rectangle changed by an ellipse,[11] but that doesn't detract in any way from Metapattern's focus on disambiguating behavior through additional differentiation by recursively situating objects.[12]

Please note that in figure 5.c the direction of the relationship has been reversed as compared with figures 5.a and 5.b.
In Metapattern's original notation, it was believed 'logical' to have the identity of the situated object point to that object's nil identity. By implication on account of the axiomatically dual nature of the relationship, the node at the relationship's other end would be equally rigorously indicated as the relevant context, i.e. its connecting situated object. The direction shown in figure 5.c is believed to correspond more to how a modeler practices

---

publications on the subject) written with Martijn Houtman. Jan van Til manages an English-language website dedicated to Information Roundabout.

[7] Indirections, when applied unequivocally, are equivalent in precision to a single pointer. Assuming nil identity in an absolute sense turns out as a step toward applying indirections as generally as possible.

[8] The modeling exercises in Metapattern are for the most part artificial, taken from other literature.

[9] For some applications of what still is the recommended visual notation for Metapattern, see (in Dutch) Conceptueel informatiemodel van GBA Startpakket+ (2002) and, written with Jan van Til, Multifocaal netwerkmodel (2006).

[10] For an overview, see Metapattern, development of notation (2012).

[11] This particular alternative is considered in Modifying Object-Role Modeling into Situated-Object-Behavior Modeling with Metapattern (2013).

[12] Recently, a(nother) notational equivalent was applied by Jan van Til on the most practical ground that it enabled him to at least get started with demonstrating Metapattern to his colleagues while continuing to use their prescribed computerized drawing tool with familiar symbols; see (in Dutch) his blog Architectuur van informatie (September 2013). Starting from two nodes, the additional node annex additionally differentiated situated object is represented using UML's notation for association (class); see figure 10-7, p. 145, in: **The Unified Modeling Language User Guide** (Addison-Wesley, second edition, 2005) by Booch, Jacobson and Rumbaugh. Despite borrowing notation from whatever other modeling method/approach, what results remain Metapattern models, of course, due to the characteristic constraints on the nodes thus 'associated.' This particular notational alternative uses a surface far from economically and is therefore unsuited for models as the number of nodes increases. For explaining Metapattern's principles to an audience of practitioners still attached to a more traditional modeling method/language, though, it might even be more effective.

differentiation. S/he starts from a situated object-to-be-additionally-differentiated, say dragging it to and then dropping it in a situated object-seen-as-representing-situation. It results in-between in yet another situated object, and so on.
Of course, the relationship being dual, one direction is equivalent to the other. The basic requirement is that some direction should be consistently applied within a single model.

A model is, so to speak, bootstrapped from the horizon. Next, one or more situated object – for practical purposes taken as performing the service of nil identity, too – are attached directly to the horizon.
For each such 'primitive' situated object, the horizon acts as both situation and object-to-be-differentiated. Applying Metapattern's visual notation throughout would yield figure 6.a. A simplified notation is preferred for such primitives; see figure 6.b.

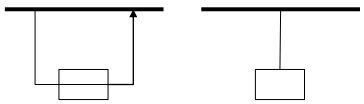

figure 6: keeping notation as simple as possible.

In fact, a sign for direction may be omitted from any line 'hitting' the horizon. When the horizon is not both situation and object-to-be-differentiated, as in figure 6, it can only constitute situation for the additionally differentiated situated object.

In 2008 the Standardisation Forum, a Dutch government institution, recognized in Metapattern a modeling method/language uniquely positioned to facilitate so-called semantic interoperability. As a testimony to the Forum's broad-minded approach to standards, successful modeling cases with Metapattern helped to create awareness of real-world semantic variety including the need to be able to interrelate differences in meaning.[13]
In order to develop a stronger case for acceptance, Standardisation Forum subsequently commissioned two independent evaluations of Metapattern.
The first evaluation was performed in 2009 by Novay (previously known as: Telematica Instituut). Novay's researcher quite rightly focused on additional differentiation as Metapattern's distinguishing, even single modeling construct (shown here as figure 5.c).[14]
In 2010 followed a second evaluation, drawn up by Rand institute. It also highlighted what is shown here as figure 5.c as exemplary for Metapattern's methodical expression of integrated variety.[15]

The conciseness of Metapattern's regular modeling construct is precisely what makes possible combining them with unlimited variety. Whatever the scope of the configuration, every node carries an unambiguous meaning (concept).
With Metapattern, there are always these two traditionally separately applied perspectives

[13] Available in the English language is Open system of systems' semantics: practice pattern, beyond central registers etc. (July 2008).
[14] Novay's evaluation report has been published (in Dutch) as Attachment C, titled **Contextuele verbijzondering: inspiratie door Metapattern**, to Semantiek op stelselsschaal (Standardisation Forum of the Netherlands, June 2009, pp. 34-53).
[15] For the evaluation report from Rand Institute, see Metapattern in Context (Standardisation Forum of the Netherlands, May 2010).

jointly at work. It helps to call them angles. One is analytical, differentiating. What stands for this angle is the object-to-be-differentiated. The other angle derives from synthesis, integrating. It originates from situation. From any two nodes seen from such angles, yet another node originates, that is, where both angles 'meet.' A horizon provides the necessary boundary condition in order for the modeling construct to be generally valid.

How much of this admittedly dense theory should a practicing modeler command? But then, is it really so difficult? I would say it presents an even insurmountable difficulty for researchers and practitioners clinging to a strictly analytical paradigm.

On the other hand, when you are open to try, it shouldn't be all that difficult. Get into the rhythm of Metapattern's idea of situatedness. One-two-three, relate 1. object-to-be-differentiated with 2. situation, to constitute 3. situated object, and so on, one-two-three, and so on. Decompose both up- and downward as required, until you are satisfied about having captured relevant behaviors unambiguously (according to your interests and especially those of relevant stakeholders). And if you want to make sure, every now and then, that it is not just any rhythm, but is thoroughly grounded, consult this paper and other literature on Metapattern. For, indeed, there is far more to modeling integrated conceptual variety than meets the eye that is not adjusted yet to a synthetic view.