

# 71.

## Notes, May – November 2019

Pieter Wisse

These notes originate for the most part from my email correspondence.

### 71.1

First of all I apologize for the delay in answering your message. I have just returned from two weeks absence, having the old-fashioned habit of not checking my email while away.

Of course I am most happy to offer “guidance.” However, it seems I need your guidance, too. From your brief introduction I do not yet understand what you are aiming at. Is there any relevant documentation publicly available that you could send me and/or refer me to? Meanwhile I shall be going over some of my other publications in English available on the web that might help you to an update on Metapattern. Its principle has remained unchanged, but since approx. 2002 I have adopted a more ‘rectangular’ notation. I’ll let you know soon.

What KnitbITs does, is implement what is modeled the Metapattern way. It is not a modeling tool. My idea is that modeling starts from a conceptual orientation. Again, KnitbITs is the technical platform. When your interest tends more in this technical direction, I’ll involve KnitbITs’ developer; with our company for r&d, Information Dynamics, Martijn Houtman is also building prototypes using KnitbITs.

### 71.2

Thank you for referring to your profile with some background. I see that you work for a health care organization. It certainly agrees with where I favor having Metapattern/KnitbITs applied, i.e., facilitating critically important processes in society for real people. And it is there where, in my opinion, a paradigm shift is mostly needed. If you allow me to come straight to at least my point, :-)) and also please make allowance for English not being my mother tongue, it will just not work to model and so on to deliver information from a particular organization out, and especially not enforcing so-called standardized meanings. Here in the Netherlands, I currently argue for the judiciary that for necessary and sufficient differentiation covering all stakeholders with their varied interests et cetera, first of all modeling should be done from the outside-in, with their organization participating equitably in information exchange at the much wider scale. More or less the same predicament from a self-centered perspective, I would say. I am having to admit, though, that I find decision makers in larger organizations hard to convince otherwise. Yet, an enlightened organization will take the lead for such a, say, infrastructural approach, it already on the short term very much being in its own interest, too (if not especially in its own interest for whatever period of time). You are quite right to suggest, at least that is how I came to understand your first message, that what is practically needed, therefore, are tools supporting variety of information, from modeling to delivering. When you get the hang of it, actually nothing really is more evident.

Am I making some sense to you? Is my assessment fair of what you would like to explore?

As far as what I feel is an improved modeling notation for Metapattern, see

[Metapattern, overview of notation](#) (a single sheet). Should you want to have this later notation compared with what you are seeing introduced in the Metapattern book, see [Metapattern, development of notation](#) (another single sheet). I strongly recommend [Open conceptual modeling with Metapattern](#) where you'll find Metapattern outlined in some three pages, including its semiotic foundation (an understanding of which I find is critical to appreciating and then unambiguously modeling variety). And then there is a succinct but wide-ranging model with some explanation of principles you might find helpful; see [Resident, designing a contextual-semantic diagram with Metapattern](#). Please note, it is just an example to suit registration of residents here in the Netherlands.

Speaking of context, I on my part am still far from understanding what you mean by “pre- and post-processing of metadata.” My idea of metadata is that it concerns typing (with, what is in a name, Metapattern as the method for recursively contextually differentiated typing). But I have heard of other interpretations, metadata meaning master data, or reference data, for example. Getting your meaning for metadata clear, could then help me to proceed to follow what there is about pre- and post-processing it. While I have already been blunt making a point of relevance, I am gladly using my privilege to come across as completely ignorant at early stages of our communication, too. :-)

I am also very much looking forward to studying what “examples” you feel free to share.

### 71.3

I'd first of all like to confirm receipt of your message. Thank you! I've now read it once. I'll read and reread it for better understanding, and subsequently come back to you trying making some sense, too. I am most happy to proceed on an informal, personal basis. I have never seen quality of design been achieved otherwise.

### 71.4

I agree with you on the need for a tool supporting, as a priority, being “free [...] to think about the problem.” Your approach makes it clear, however, that you are actually considering problems, that is, both in multitude and interdependent. Regretfully, that is not at all what business cases are usually about, instead merely aiming at solving a single so-called issue and be damned the consequences. So, I am very aware that a business case for Metapattern is extremely difficult, if not impossible, to make to currently fashionable management. I am all the more grateful for your interest, and hope to be able to honor it.

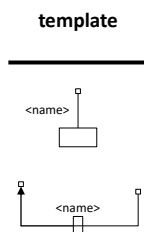


figure 1

Now, a full-fledged drawing tool I cannot offer for Metapattern. As there is just a single modeling construct, quite a simple one at that, you really cannot go wrong on syntax. I myself start modeling on the proverbial back of an envelope. When I want

the diagram to look more, say, official, I take to PowerPoint (in case I don't expect the model to be extensive). I do use some sort of template, see figure 1. I am also sending it as an attachment; please feel free to use it.

On a scrap of paper I sketched how "member status codes" might be differentiated (and don't bother yet to set me right; I already know myself I must have missed the point miserably :-). I just refer it here to demonstrate how I would proceed for a PowerPoint rendering, see figures 2 up to 6, and 7.

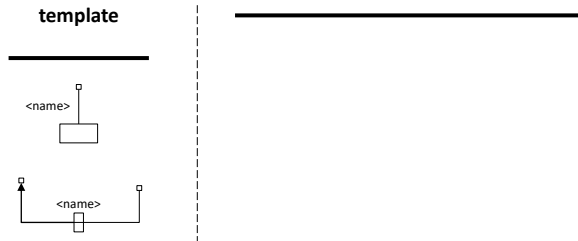


figure 2

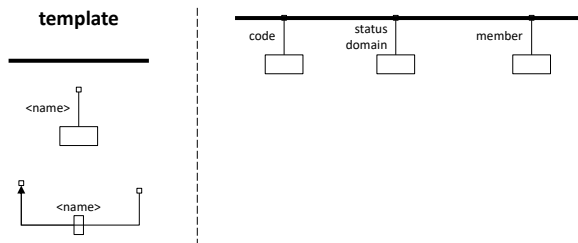


figure 3

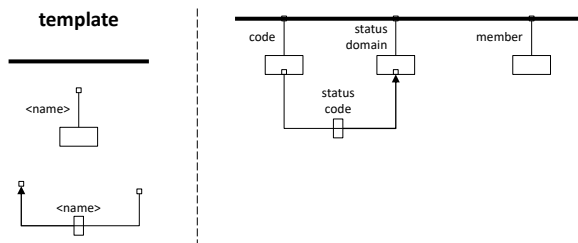


figure 4

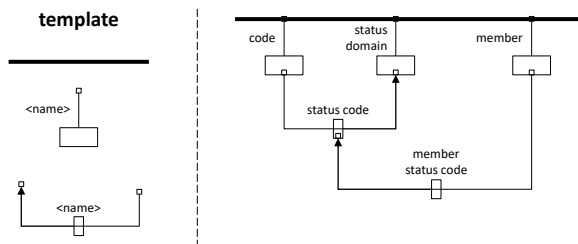


figure 5

It is aesthetically more pleasing to make the small squares used for lining up invisible, see figure 6.

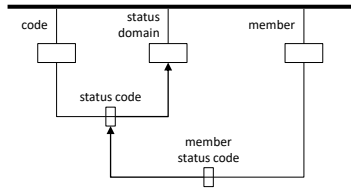


figure 6

To me, an alternative interpretation almost immediately presents itself, as shown in figure 7. Not really having a clue, I do prefer figure 6.

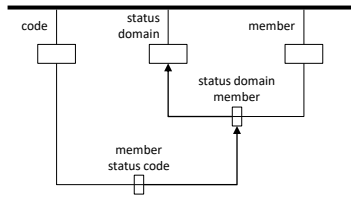


figure 7

For coming to better grips with a few concepts that seem confusing for lack of contextual differentiation, I find that a presentation tool such as PowerPoint is already more than sufficient. And of course leaving out the template, I advise you to present it as a developing model, i.e., a progressive series of models (as I have tried to do above). It makes eliciting relevant comments easier.

You may find, however, that for a more extensive conceptual design you run against limits of space with PowerPoint and the likes. I did, anyway. For an example of a far more elaborate model, see [Open system of systems' semantics, practice pattern: beyond central registers etc.](#) As it covers a large set of governmental responsibilities and activities, it might be interesting for you to go through it. Anyway, it practically demonstrates how powerfully far-ranging a model of interdependent concepts will turn out. Well, maybe later. I am sure you recognize I've used Visio to draw it. I've tried to translate the template from Dutch to English. You find it attached, too (and it is, of course, also available for your use). The Visio template includes as a construct the homogeneous hierarchy. And when for a modeling construct you already know in which direction the contextual differentiation should be drawn, it gives you the choice between left and right orientation. (I can easily add those to the PowerPoint template.) Of course, when forced to change a model you may find yourself adjusting directions, after all.

To be continued.

### 71.5

Differences usually occur for a reason. So, people are often extremely relieved when they feel they don't have to fight et cetera for their particular meanings to be taken seriously. Being reassured that their interests both are and remain supported, it right away makes them more open to do away with so many irrelevant differences they might counterproductively uphold. Anyway, that is the theory.

### 71.6

While admitting that my previous contextualization of a theme you have mentioned, that is, member status code, is pure and wild speculation, I nevertheless continue

with it. I am (mis)using it to illustrate some hopefully useful design guidelines for up-scaling information from independently operating applications/systems to an integrated order.

Immediately there is an instance where I struggle to find an equivalent term in English. Take system. With the term systeem available in Dutch, there does not seem to be a problem at all. However, in Dutch there is also a noun term, a luxury, for sure, and it reads stelsel, for something that is systematic at a yet larger, wider et cetera scope than ... system. Not for want of searching, I still fail to find in English an appropriate single term referring to that concept. I could translate it, and have indeed done so, with system of systems. But that still considers the constituting systems as its unchanged and unchangeable parts. What is at stake, I find I better call an integrated order. Do you have a terminological suggestion? Anyway, a system of systems, then, is only where we start from toward integration. I'll return later to – designing models – facilitating what I understand is what you refer to as such a process of “system integration.”

Before, what I have added to secure the disambiguation of codes is the concept of status domain. For such concepts are traditionally left implicit – and habitually even unconsciously so – for a traditional application (also read: information system) meant for separate operation. From identification of the application it is – supposed to be – clear what concept the codes entail. When there were no means to interconnect computers, no reproachable neglect was committed. All that has changed, to put it mildly. As I am continuing numbering the figures, it is in figure 7 that with Metapattern the relationship is sketched traditionally holding for a particular application.

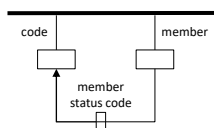


figure 8

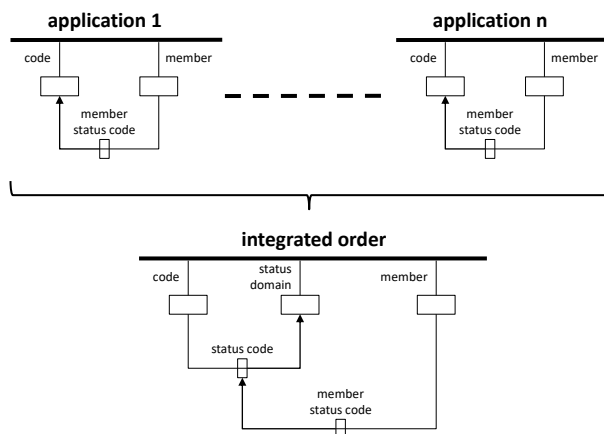


figure 9

When ‘mixing’ information from several applications, the implicit context for – the meaning of – codes provided by actually using a particular application, rather than any other, has been removed. At their encompassing scale, the relevant context

should now be made explicit. From the simplistic assumption that all separate applications share the same model, figure 9 ends up by repeating figure 6. Their integrated order only takes – please note, in this overspeculative example – one added concept for necessary and sufficient disambiguation.

An extension of this model could include – reference to – the original applications, see figure 10. Just experiment with positioning concepts you find relevant. From one contextualization, soon enough you'll get the idea for another, and so on until you are satisfied having exhausted possibilities (finding at some later time that you've gained some additional insight, and modifying a model accordingly).

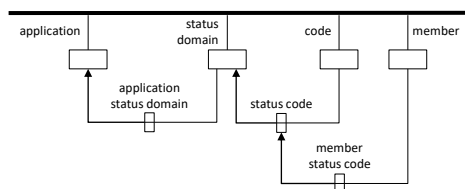


figure 10

If I may say so, :-) what really bothers me at the level/scale of an integrated order is the concept of member. I am happy that you right away recognized the contextualization of resident as “an excellent example of the type of problem we face internally at the corporate level and across all our health plans.” Being totally ignorant of health care structure in the United States, please take what follows as some suggestions for modeling the Metapattern way, too. I am only making use of the theme/case you mentioned, so you may more readily design the proper models yourself.

My guess is that being-a-member implies an insurance plan, with membership amounting to an insurance policy.

For now I'll suppress my inclination to follow a path to more and more extensive contextualization. Isn't membership possible of a sports club? And then, of what not? A citizen is also a member, isn't s/he? Such extension might seem irrelevant for the integration at hand. However, in my experience it never fails to yield a more flexible model, both suggesting additional opportunities with information services and better preparation for eventualities with respect to information needs. It certainly is worth the effort, an intellectually rewarding, at that. In fact, I have drawn up what I consider to offer a foundational model for limitlessly differentiating capacities of participation (also read here: modes of membership). I would say it even is a requisite for – support of – authorization regardless of scale, providing for audit trails, et cetera. For the scale, and with the content, you are aiming for with integration, I am certain that sooner rather than later I shall introduce you to it. (It does require more translation on my part.)

You cannot miss the modeling approach for arriving at figure 11.a being somewhat similar to the one taken for the resident contextualization. You'll also notice that according to the model in figure 11.a, only persons are eligible for insurance. Well, already there it is, I mean the requirement for differentiating between capacities of participation. My general term is: actor (luckily, same in Dutch :-). I am aware that in the U.S. the term party is often used. I prefer actor. For the moment, I am simply suggesting there is a through-and-through recursive model constituting it; you might get some structural impression from viewing the Dutch version: [Informatieverkeer op](#)

[stelselschaal](#). In figure 11.b I have changed person into actor, the latter being an intermediary node according to the underlying (sub)model. Actually, member could be brought under the concept of actor ...

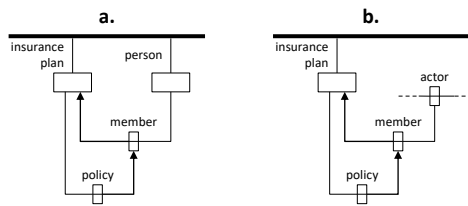


figure 11

Having included – the concept of – insurance plan, it makes more sense to me that a so-called status domain is contextualized by it, and subsequently the set of relevant (also read: fitting) status codes. Figure 12 outlines that an actor may be a member of an insurance plan. As such, that is, as a member s/he has been assigned a status. The constraint has not been specified that for status assignment only statuses are allowed that belong to the set for the insurance plan that s/he is – taken as – a member of. It is of course a matter for contextual differentiation, too. Adding it to the model, however, would clutter it and distract at this early stage from designing an integrated order for the concepts of major concern. At the relevant node an asterisk or so could be added to suggest such a constraint (or any remark).

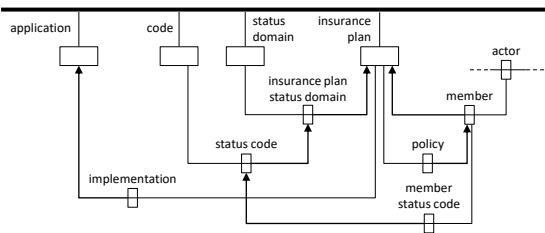


figure 12

Please let me know whether or not – be assured that I am not easily insulted in this respect; and of course I am only too happy to save me the trouble – you find being presented with such series of modeling exercises helpful. In my next message, I'll try to explain how distinguishing between two extreme uses for Metapattern provides guidance, sort of iterating between them. By way of preview, on the one side Metapattern facilitates modeling an integrated order – almost – purely conceptually. Such use might be characterized as practicing idealized design (an attitude coined by Russell Ackoff). Then, you try not to let any technology, or whatever, come in your way. It is practicing philosophy, really. On the other side there are, here and now, real “integration problems.” It means that you have to start from applications, systems and databases that actually exist and in most cases, if not all, will surely continue to be used for some time. While the result of idealized design may help to set a longer-term goal, and you would therefore be strategically disoriented without it, in real practice it is not what gets you off the mark today. For that, as you are only too well aware of, separate inventories of metadata need to be merged and information (I suppose you would call them: data) reported comprehensively.

Here again I have trouble finding an equivalent term in English. I find it has to sound right, too. The Dutch term is: rotonde. The emphasis is on the middle syllable. A 'rotonde' is where traffic enters and goes around until it exits. On the average, drivers benefit from the device for streamlining traffic flow. In an English dictionary I find the term traffic circle for it. Would you call it a roundabout in the U.S.? If so, it suggests other meanings that even run contrary to what the Dutch term provides, that is, the association with benefits from streamlining. (And roundabout also doesn't sound right to my ear. :-)

What I mean by 'informatierotonde,' is that operational databases can be connected to it on the basis of a model of identifiers for the concepts of interest for comprehensive reporting. To start with/from, having copies of databases available is fine, if not preferable. The idea is that nothing is changed for and/or in the databases involved. Of course, this is not at all a new idea. But supporting the interconnection with Metapattern/KnitBITS is, and it does appear to make a difference.

The Dutch judiciary was swamped by a host of databases when moving from decentralized to centralized financial management. Applying an 'informatierotonde' made comparisons between contents of more than a hundred different databases possible. The resulting overview pointed to which values for which attributes should be adjusted for the operationally still separate databases and their applications to subsequently operate in managed harmony. Of course, it is some way removed from, say, proper integration. But an 'informatierotonde' will get an organization at least already a surprisingly long way, even very quickly and at hardly any costs (of course saving much more in the process). Together with Martijn Houtman I've written a short introduction, yes, in Dutch. For my next message I plan to give you a summary in English.

As you are familiar with object-role modeling, I'd like to refer you to my short text [Modifying Object-Role Modeling into Situated-Object-Behavior Modeling with Metapattern](#). I hope it gives you yet another introduction to Metapattern's governing principle.

To be continued, regards.

## 71.7

Thank you for your terminological guidance. So, information roundabout it is. Indeed, a close friend of mine has straight away chosen this term for a translation; I have been hesitant.

As you see, models are technically not difficult to draw. Do you want me to send you files in Word-format, with figures in PowerPoint-format embedded?

Your colleague is of course also most welcome to deliver questions, comments.

I have made a note to elaborate on membership or, as I would like to put it more generally, actorship. Here is a brief introduction. When increasing the scale for integrated order I have come to especially appreciate the idea of conditioned actorship. For example, assume that a separate insurance plan exists for employees. First of all, the condition for being an employee is being a person, with being an employee subsequently a condition for being accepted as an insuree. I know insuree is not an English word, but that happens when trying to fit concepts at larger scales. I wouldn't dare to set a limit to the levels of conditioning for actorship. As a recursive construct, it always works. It turns out rather addictive, as I must warn you (now already being too late :-). You'll recognize conditioned actorships all over. What makes a person eligible to vote? What makes a person eligible to be on a



sports team? What makes a person eligible to occupy a job? There was no need to specify such conditions with applications each oriented at some particular, say, final actorship, for example employee, customer, et cetera. All conditions are then 'conveniently' left implicit in constituting each application. For integrated order, however, there is no way around making them explicit. First of all, a qualitatively different approach to modeling is required.

## 71.8

The metaphor of a roundabout I find quite apt since digital technologies are no longer only used for processing information locally, i.e., separately facilitating single applications/systems, but more and more for information exchange, too. In an order(ing) sense, information therefore has also very much become a matter of – regulating – traffic.

I do hope you are not getting bored by me going on and on about conceptual aspects of – the use of – language. In Dutch, and in German, for that ... matter, the traditional term for, say, people being actively engaged in their community is: *verkeer* (German: *Verkehr*). Subsequently using bicycles, cars, airplanes, or whatever for going about our ways is also called 'verkeer,' but is understood as being-a-part-of the far more general concept of – well, why shouldn't I try an English neologism? – *communiting*.

Digital technologies offer modalities of *communiting*. What we do with them differs from moving ourselves and/or material goods about. Instead, we ourselves can stay put, while information goes about (with data inevitably being a material good, too, which here I'll conveniently ignore :-). In principle, though, it is nothing new. People since time immemorial have used smoke signals, sent letters, et cetera. As it is, we now conduct our lives occupying ourselves increasingly with information *communiting* (Dutch: *informatieverkeer*) as an aspect of *communiting* (pleonasm, but perhaps helping to make the point: social *communiting*). However, what have been conceived of as separate applications are simply not sufficiently prepared for seamless – support of – participation in information *communiting*. Let's say that nobody ever dreamt of the communication potential of digital technologies. So-called professionals, however, are still putting processing first, making an afterthought at best of interconnection. For some time now, that is the wrong way around (as "integration problems" are trying to 'tell' us).

It is because of such "integration problems" that stand-alone systems are classified as legacy. For all sorts of reasons, I would say the most important being of a psychological nature, those applications cannot be changed, replaced and so on overnight. My idea of an information roundabout is to start with 'information traffic.' From its principle it is non-invasive regarding the applications – rather, their respective database instances – being thus interconnected. The maxim reads: Don't change anything! I find it goes down well when qualifying a ... change proposal. In practice there is no paradox experienced. For what stakeholders take away from it, is that their stakes are being respected, served and so on. So, each user can continue to use her/his particular application just as s/he is ... used to. What really does change, is that through the information roundabout each user is helped with suggestions for improving the quality of information in her/his application. I was actually surprised by how fast it showed results with the Dutch judiciary. With just one knowledgeable employee 'operating' the information roundabout, from – copies of – over a hundred databases he could offer specific suggestions to specific users.

They soon recognized their own advantage. No more problems which are in fact impossible to solve in isolation as all too often occurs when, for example, an external actor unwittingly, but often unnecessarily and thereby confusing coordination, provides different information to different applications. Of course, what most importantly did change was trust. Local users came to rely on the information roundabout for the quality of their work, acknowledging mutual influences operating on what they now recognized as the larger scale of real ... communiting. So, also very soon, thus being freed from frustrating work, they were even happy to let the information critical for coordination be managed from the 'site' of information roundabout.

I can only strongly recommend starting with an information roundabout. Of course, since decades already we are trying to align differences by making translation explicit, that is, putting something like a value-here : value-there table in between. Thereby solving one problem, however, creates the next when such tables et cetera take up definitive positions, that is, effectively adding to the number of applications/systems. A solution of value-translation should be temporary.

Now, an information roundabout is really not different from a set of such tables, at least to begin with. In fact, it also should't, as that is what usually counts as the immediate problem. Yes, it aims to generalize (which also has been done before).

I would say the main feature of an information roundabout as I favor it is its orientation at contextualization. At the start, that is, interconnecting databases-as-they-are, Metapattern is only crudely relevant. You might ask, why bother?

Gradually, on the side of the information roundabout contextual differentiation at the scale of integrated order may be enhanced, with corresponding modifications made to connected applications (thereby more and more changing into mutually interdependent constituting elements for the integrated order for which an idealized design serves as its goal to aim at). It is of course contingent what the optimal order (in time) is for approaching the relevant integrated order (as a result). My general idea of an optimally controlled change process facilitating information communiting/traffic is one of morphing an information roundabout. By the way, there is no rule that says you can only start with one information roundabout.

In my next message I'll attempt to give a brief structural explanation of an information roundabout. You should find it trivial.

To be continued.

## **71.9**

Intermezzo. About thirty years ago I happened to be talking to the publishing director of the leading magazine in the Netherlands on soccer. Actually, he was talking to me. He told me his idea was to set up a database giving readers the opportunity to look up facts. I managed to ask him what he wanted such a database to cover. His reply was both immediate and short. Everything!

It was to prove a life-turning event for me. :-) Of course, he didn't really mean it. But then, he was certainly right, if that's what he meant, anyway, that it is impossible to draw a line for what hard-core fans might be(come) interested in about their idols (and also about, I am afraid, their supposed enemies).

It was the apparent practical absurdity of his demand for "everything" that I decided to take seriously, at least for some time when trying to come up with principles for, say, open-ended integrated order for information.

No, a database on soccer and related facts never materialized. For the director, our

conversation had merely been a fleeting one, soon forgotten about. Almost right away, however, I did document what I believed to constitute limitlessly practicable modeling principles, indeed, taking a cue from object orientation and pointing out my departure. About ten years later, the book you got hold of, that is, **Metapattern: context and time in information models**, was still written presenting Metapattern as an extension of sorts of object orientation.

After that book was published, I set to work more seriously to try and fit Metapattern with axioms proper. One of my ideas was that, if information is always a sign, vice versa, semiotics should provide the frame of reference. Yes and no. By way of a resounding yes, two sentences by Charles Peirce especially caught my attention. They read: "A sign is something which stands to somebody for something in some respect or capacity [...] which I have sometimes called the ground." Now I had already been reading Arthur Schopenhauer elaborating on will as ... ground. And was Metapattern's concept of context not some ... ground, too?

Unaware of Peirce's work having been declared sacrosanct in academic circles, and therefore not being disciplined, limited, et cetera, I felt free to extend his triad-plus-ground first of all into a hexad and subsequently into an ennead. I had already established, well, to be honest, just for myself, that Metapattern provides for requisite variety in conceptual modeling. But when it is so-called requisite, there should be 'other' varieties it agrees with as such. Or? What Peirce's triad-plus-ground doesn't yet suggest, is now covered by the semiotic ennead, i.e. full correspondence of varieties between Peirce's originally assumed triadic elements (developed into the three enneadic dimensions/moments). No, at least not as far as I believe, Peirce's semiotic scheme is not yet both necessary and sufficient for 'grounding' integrated order through modeling.

Especially the last twenty years I have been going over many writers with their theories, and continue to do so, in attempts to discover earlier methods for conceptually modeling variety. In some work or other, and at some time or other, really from Plato onwards all relevant axiomatic concepts seem to have been mentioned and put in some perspective. Only lacking was yet another synthesis. At least, I have so far not come across an earlier version of my proposal. It should not be surprising, though, that it happens to us here and now. For we are only quite recently confronted with digital technologies pervasively facilitating 'communiting,' resulting in problems of integrated order for information that require a qualitatively different method for their practical solution. I hope I don't offend you to suggest semiotics to fundamentally change, too. And enneadically speaking I don't see how to set semiotics apart from philosophy, linguistics ...

### **71.10**

After having introduced an information roundabout as a practical tool for getting underway toward an ever more tightly integrated order for conceptual variety at whatever scale, from a short Dutch-language text I am now reworking my presentation of a general model into English. Please allow me some time to do so. I therefore sent you my previous message by way of intermezzo. I did not want to burden you with additional references. However, you did succeed in recovering [Multicontextual paradigm for object orientation: a development of information modeling toward fifth behavioral form](#). As indicated in the intermezzo – the previous one, that is, for this is yet another, :-)) – that article is my very first documentation of ... what only several years later I named Metapattern. And what you are reading is

my translation, also done several years later, from the Dutch original. As an introduction to Metapattern I wouldn't now recommend it; I find it more historically relevant as I am still taking my main cue from object orientation while trying to – playfully? – mix it with ideas inspired by relational normalization. Nowadays I prefer the reverse presentation, with Metapattern explained on its own – terms for – concepts, and only subsequently showing how it covers earlier modeling methods/languages through posing particular constraints (as you have seen me do with object-role modeling in another article; as you have brought up ORM again, see more on it below).

Of course I cannot help notice you have spelled the first word of its title of **Multicontextual paradigm for object orientation** as “multicultural.” Yes, you are right about that, too! :-)

Am I right to understand from your comment about one of those earliest of modeling examples, i.e., about a person depending on circumstances behaving either in an extrovert or an introvert manner, that you are contrasting “technical advantages” with the persistent backwardness of still being unable to “model [...] in context-specific ways”? When, indeed, you are voicing surprise about the gap between current needs and available tools, I believe a new theory including paradigm is the missing link (as I have briefly tried to put forward in [Innovation dynamics across theory, technology and tool](#)). Or are you making another point?

Are you perhaps saying that a new difficulty crops up? If so, you are right. However, there is no avoiding it. If anything, Metapattern helps you to recognize and address it. Take whatever number of non-overlapping contextually differentiated descriptions. Doesn't non-overlapping mean that it has become impossible to tell which descriptions pertain to which object? Yes. Luckily, though, objects change from one behavior to the next. That is, from earlier behavior(s) we can – almost always – infer the object a subsequent behavior should be attributed to. And then we create ‘behaviors’ especially for that purpose of attribution, for example ‘fitting’ objects with unique identifiers. Now applications with, in terms of Metapattern, each a limited horizon have issued particular sets of identifiers. Let's take persons as an example. How can we tell for establishing an integrated order that id-x from database A refers to the same person as id-y from database B? Often, there are some common properties registered in both. The non-overlapping criterion does not hold for so-called legacy databases. No, what actually does overlap never offers “certainty,” at least no absolute certainty (which is, of course, a main reason for increasing integrated order). If the lack of “certainty” is your point, not just theoretically but especially in practice, I even strongly agree. However, comparing contents – when persons are relevant ‘objects,’ usually name and address are available from databases – may already go a long way (and, by the way, that is precisely what an information roundabout at first helps to analyze et cetera). Then, in many cases you can be certain enough to proceed (and some mistakes will creep through; negative effects can be minimized when the persons involved are made part of the quality feed-back loop, and you may have to come up with something to get them to respond; and, anyway, such negative effects are far less than would continue to occur through use of strictly separate databases). Inevitably there will be cases, though, where you have to contact people and make further inquiries. Miss-spelling of names, whatever. It is what they did at the Dutch judiciary, make phone calls. People are glad to hear from you, that you take care. In fact, they still make such

calls, as users make mistakes. But mistakes are fewer and fewer, so they need to make fewer and fewer calls. And as a matter of “certainty,” too, :-) there is no alternative, and as soon as an id-for-integrated-order adequately connects the contextually differentiated behavioral descriptions for ‘its’ object, changing and/or adding such descriptions continues ‘normally.’

The three enneadic dimensions are derived from Peirce’s triadic elements. He must have struggled with the choice between – the philosophies – of realism (simplified: about objects) versus idealism (simplified: about concepts or, as Peirce called those, interpretants). His stroke of genius, I find, was to decline to choose. What he did was to mount an element mediating. He put sign between object and interpretant. All I have done, really, is to suggest corresponding structures for each of the Peircean triadic elements. Doing so, I believe I have taken Peirce’s qualification with ground more seriously than he himself did. At the time, there was of course no problem that needed to be solved by moving beyond taking one grounded whatever at a time. Interconnecting databases without practical limit requires us to be able to collect different meanings while both being able to unambiguously tell them apart and likewise interconnect them. Anyway, I credit Peirce for founding a semiotics allowing us to manoeuvre between the otherwise irreconcilable opposites of realism and idealism. We can, and should, move on, though.

As far as I understand ORM, it remains fixed on objects. Compared with, say, classic OO, ORM is already far more flexible. For the role-aspect makes it other-object oriented, too. That way, ORM is about the role of one object as related to the role of one other object, or the respective roles of more other objects.

What may cause confusion is the term ontology. What I take ontology to mean in a philosophical sense, and that is also how I consider the axioms aka ontology for Metapattern, is the most fundamental idea of what constitutes reality (and as such simultaneously offering a general concept of reality). As such, ORM’s ontology is still atomist, which severely limits flexibility. For objects nevertheless come first. And each object ‘has’ roles for relating to other objects (with those other objects ‘in return’ fitted with roles, too). Based on that ontology you can draw up models. Does Halpin perhaps call – some of those – models ontologies? Again, that is not what I understand by ontology. To me, they are also just models.

As you have gathered from my comparison of Metapattern with ORM, and I believe I have been quite fair, the respective ontologies – in the philosophical sense – differ. Yet, ORM comes quite close, as I have tried to demonstrate through proposing constraints. What ORM forces upon the modeler, however, is specifying all objects pertaining to mutually relevant roles. With n-ary relationships, I find overview is soon lost. A model therefore doesn’t scale easily. ORM has not purposely been designed for use at the scale of integrated order.

Instead, Metapattern principle of differentiation is binary. For behavior it is object-in-situation, corresponding in a model with signature-in-context for – identifying – description. On this binary principle rests the possibility for limitless recursion. Thus, with Metapattern a model scales almost naturally.

I am happy to hear you are going for hands-on experience. I would say that you have read, at least for the moment, more than enough about Metapattern. There is no substitute for practicing it, stimulating your “thought process” as you go along, on what is of course your own example.

### 71.11

What the Dutch government was attempting at the time, and I am afraid still is, was to organize what it thought are master data in separate (!) databases each with national scope. So, one database about residents, another for residences, and yet another for legal entities. Well, that's about it. The idea being that all organizations at all levels of government would use such uniquely so-called authentic information. Contextual differentiation was not considered. The "gist" of my model is therefore to start by cataloguing the separate models from the simple if not rather primitive but at least more or less generally valid principle of whole-part division, with when required a part subsequently considered as a whole with further parts, and so on. Anyway, it then becomes possible to relate thus differentiated 'data elements' between databases. For example, in the residents' database an address is included 'about' the resident. As a type it should correspond with a set of elements in the residences database. A step toward an already somewhat integrated set-up would then be to refer from one database to – an entry in – another database, i.e., avoiding redundancy. It is in principle really no different from what I later called an information roundabout. However, [Interoperabiliteitscatalogus informatietypen](#) stops at cataloguing and thereby only roughly suggests how to proceed and with databases still to be designed et cetera, at that. An information roundabout is also, indeed especially so, about establishing and managing practical interconnection between legacy databases. I am on it to document such a roundabout's "gist" for you in English.

### 71.12

I have asked Martijn Houtman to comment on the requirement for support of "connect[ing] to everything we need to." As I have explained, I have taken the demand for "everything" seriously in the conceptual sense with Metapattern. :-)  
I know Martijn has gone to great lengths to develop KnitbITs technically for matching flexibility. So, more to follow as soon as possible.

You are right, an information roundabout should not be 'about' metadata, with the conceptual exception of course being identifier as a type (values of which constitute the necessary and sufficient conditions for – establishing – actual connections). Is that what you are saying? That is where regression stops. Do you agree with the term identifier? Key, perhaps? Or?

### 71.13

There is an earlier message of you I still want to respond to. In **Ontological Modeling: Part 1** Terry Halpin right away confirms that

the information systems community typically uses "ontology" for a conceptual model of some business domain[. p. 1]

Halpin goes on to emphasize that

An ontology used in information systems is typically restricted to a single business domain. So you have different ontologies for different domains[. p. 1]

In a pervasively interconnected world, such a double restriction is of course nonsense, if by now not outright counterproductive. I really don't have a clue how to classify businesses as mutually disjunct in order to arrive at equally disjunct domain models. And why just for does business? Isn't, for example, a single person domain a domain just as well? Et cetera.

I apologize. I am all getting wound up again at the sheer short-sightedness. :-) From the impossibility of conceptually separating domains, it follows we need a methodical approach to model – towards – integrated order, period. Preferably, a single modeling method is adequate to the task. And underlying such a method can only be an ontology in the philosophical sense. It has therefore been an act of stupidity that, here's Halpin again,

[m]ore recently, the informatics community [has] adopted the term "ontology" for a different purpose.[p. 1]

Well, I didn't. About this, I am expecting especially enthusiastic support. :-) However, the modeling method that Halpin champions might very well, if not by design, then perhaps by lucky accident, be perfectly suitable for modeling beyond a separate business or whatever. I certainly agree that ORM improves upon classic object orientation by insisting on roles and thereby allowing for already a large degree of relativity (accompanied by differentiation). Its philosophical ontology remains atomist, though. And for that reason I also don't see how ORM really differs from entity-relationship modeling with no attributes allowed.

I merely glanced at some other papers you were so kind to send me. Most importantly, I want to get on with describing the information roundabout. In addition, nowhere Halpin seems to recognize he has to go back to change axioms first. I am immediately getting lost in technicalities beyond my conceptual focus. To me, they appear irrelevant for bringing integrated order closer. To put it bluntly, there simply is no recovering from a counterproductive turn taken right at the start. There are practical problems to be solved, and the relevant scale is for some time already that of an integrated order.

Thank you for explaining what your "sigh" was about. I hope it is soon turning into a sigh a relief at feeling confident about gaining control over, as you have called them, "system integration problems." Speaking for myself, I have actually come to enjoy solving such problems.

At the risk of misunderstanding what you have brought up referring to category, I would argue that in that sense category is equivalent with type. And conceptually, every node in a Metapattern model denotes a type where two 'previous' types constitute the next, and so on and across until you feel you have done enough differentiating. And there might be one category with different persons entertaining different prototypes, i.e., typical instances. Therefore, I would distinguish between category and prototype, each with relevant contextualization. At least to start with, I believe that is the safe option.

#### **71.14**

This brief conceptual introduction in English to an information roundabout (Dutch: informatierotonde) is based on, and doing so an abbreviation of, the Dutch-language paper [Stelselmatig overzicht via informatiesleutels](#) by Martijn Houtman and Pieter Wisse.

The conceptual model of an information roundabout is straightforward, if not trivially simple (and therefore hardly original). As you have right away inquired after, it is the tool set that critically counts to make it work. I believe it is nevertheless helpful to recognize what it conceptually is, and especially is not, too. You may find my terminology contrived. I don't see how it can be avoided when potentially dealing with "everything."

For practical orientation, start by imagining a boundary for what you want to include in – having an initial go at – an integrated order. Metapattern establishes such a boundary as a model's horizon.

You are interested in getting an overview of information, also call it data. There is 'something' about which information might be distributed across a variety of digital storages, each of such databases most likely having been set up independently at some time and place or other.

You need to identify such a 'something,' or 'whatever,' at the scale of what you are aiming at for integrated order. Actually, there may be different types of 'something' you are after for improved coordination.

For example, an airline should want to track engine parts. When, and here I am just thinking of 'something' off the top of my head, the requirement is that an engine part of some type should be replaced every x hours of engine operation time, should then be overhauled, and only be replaced in some engine after overhauling (and testing), for compliance with such rules information from on-board systems and systems for the maintenance shops must be brought together.

Of course, almost everywhere it is person as a 'something' about which information is kept all over. As you wrote, indeed,

the primary focus of most systems is not person but the roles that a person plays in one or more contexts within that system.

With persons, it is even evident that there are – more – regulations to comply with. The European Union now has General Data Protection Regulation. Anyway, using an information roundabout you have to comply with whatever regulation holds for the 'something' in question. But then, an information roundabout can also be the means you need to monitor et cetera compliance.

Where was I, conceptually, that is? I recommend to document even the obvious. For exactly at that point you are making your assumptions conceptually operational. As you may find yourself getting stuck, anyway, I often do, modeling is largely trial and error, it helps being able to retrace your modeling steps and try a different direction from the earliest point possible (symbolized by a horizon). Figure 1 shows that you are aware of some boundary. And that you have a choice of types of 'something' to be interested in at that scale of integrated order. Just to make sure about the scale appropriate for it, I have extended the term 'something' to 'io-something,' with io meaning integrated order. It sounds awkward at first, but you may get used to it. And when you can improve upon terminology, too, you right away should.

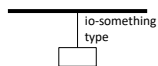


figure 1

Please note that I am developing a model, here. The sequences for using an information roundabout differ.

For a type of io-something, instances occur. And for such an instance, apart from supplying it with an io-scale identifier, you may want to make some information available to help identify it (and supporting the search option). Speaking of sequence, such values for filling a io-scale profile can – and should? – be drawn from databases brought into – a beginning – integrated order. However, conceptually I am so far only extending what is needed for a pivot; see figure 2.



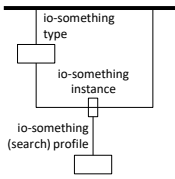


figure 2

It is time to consider the databases you want to interconnect. For every database instance you can now specify, as figure 3 shows, whether or not it holds information about somethings according to a type that is relevant to the io-something type. (And if it does not, why bother?)

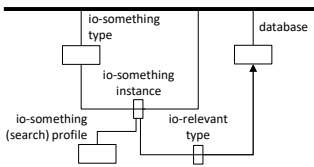


figure 3

The model is already nearly finished. On the basis of io-relevant type another profile is added, i.e., specifying selection & extraction of information about instances of that type from the targeted database. Agreed, for databases supporting the same format, such directions should be differentially modeled for avoiding redundancy. The point I want to emphasize here is that such, say, technical parameterization is included conceptually. Figure 4 sketches where positioning this particular profile logically starts from, anyway.

The final step in my demonstration of what an information roundabout entails conceptually is connecting instances from databases, not directly, but indirectly through what has been established as an io-something instance.

You will notice that this way an information roundabout has strictly been kept semantically neutral. It simply connects what you make it connect through relating identifiers, whatever is 'behind' each identifier to be presented in conjunction. An information roundabout, to serve its particular purpose, is left completely in the blind that what is identified in one database is about, for example, a bicycle, connected through an io-identifier with what might have been identified in another database as being about a health insurance policy. It couldn't care less what it connects, and that is why it is good for connecting, only, and why it is especially good at it.

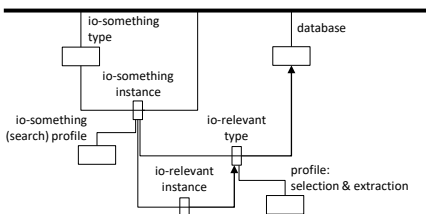


figure 4

So, in practice an information roundabout requires some additions. How do you want information to be presented for comparison, overview, et cetera? That is where it gets specific, meaningful, et cetera. However, always you first of all have to get at

information that is distributed all over, setting up a structure for continuing to do so in an integrated-orderly fashion. For that purpose, focus has to be and remain on interconnecting identifiers, only. The most general set-up is also the most flexible. Additions should remain ... additions, and not clutter an information roundabout proper.

### 71.15

I was hoping to hear from Martijn Houtman offering some explanation of the – technical – implementation of an information roundabout using KnitbITs. Anyway, my assumption is that you also want to learn about its practical feasibility, here and now. Am I right?

Apparently, I caught Martijn in a mood even more oriented at principles than I am. :-)  
What I have so far got back from him is that especially **from a most practical perspective** he agrees **in principle** with your assessment of inevitable regression when dealing with metadata with ... metadata. As you find your own views strongly confirmed, I am happy to right away forward his initial written reaction (my translation):

John is completely right. It is like consulting a psychoanalyst, only to hear her or him describe your problem for you, whereas s/he should be helping you solve it. Indeed, something similar occurs when you merely document information at yet another (meta)level. It doesn't solve the problems you may have. Instead, you have added a problem, and so on, thus being even further removed from solving what is the problem you started from. To solve it, the single focus has to be on controlling where information is kept and how to get at it. You cannot get anywhere on the basis of increasingly abstract metadata in continued isolation at whatever level(s) of description.

More to follow.

### 71.16

My admittedly superficial impression from his series of papers on ontologies, in the sense of actual conceptual models, that is, is that Halpin is mainly going through a series of modeling exercises using ORM. I find his association with and orientation at RDF/OWL as target languages for implementation conceptually irrelevant. In fact, it distracts attention from ... ontology.

### 71.17

With ORM, as I understand it, an object's behavior, i.e. both static and dynamic properties, is differentiated through role-attribution. And for its roles, an object is taken to be different from other objects. So to speak in between objects stand one or more of their role-aggregates, here as depicted with Mepatttern in figure 1.

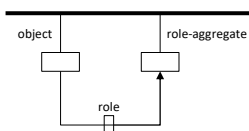


figure 1

When I am right about the recursion ORM allows, it can be expressed with Metapattern as a retyping of – an instance of – role-aggregate, or some subset of such an aggregate, as – an instance of – an object; see figure 2. This seems to me all of ORM's ontology.

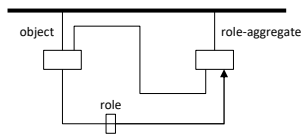


figure 2

Now, Metapattern's principle of describing an object's situational behaviors does not take an object to be different from the situation it behaves in, but **as part of** it (just like other objects may be part of it, too). For example, while people playing a tennis match are indeed called opponents of each other, it is of course the whole match-situation, including spectators et cetera, in which they – and the spectators et cetera, for that matter – behave. See figure 3.

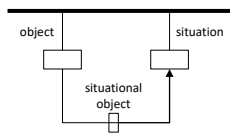


figure 3

Including retyping for recursion, all of Metapattern's ontology is preliminary sketched in figure 4.

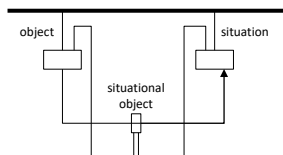


figure 4

With Metapattern, object and situation are relative concepts, though. Starting from situational object, this comes out better in figure 5. However, still not shown – how? – is a horizon as setting a model's boundary for such relativity by conflating – what are considered for the model – as the most encompassing both object and situation.

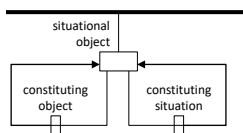


figure 5

## 71.18

As announced, distilled from some comments by Martijn Houtman, an outline of information roundabout's software:

The programming language used for the information roundabout is C#. Components are available in C# for setting up connection et cetera to several types of database/format, a.o. MySQL, MS SQL, Oracle, Active Directory, Excel and MS Access. In event of connection to (other) NoSQL databases, it must of course be catered for as their particular structures dictate.

In the brief conceptual introduction it is shown, see figure 4, that for an integrated-order relevant type in some database a profile is added supplying control parameters for selection and extraction of information about corresponding instances.

These profiles are implemented as xml and xsl files, including necessary and sufficient references to relevant connection component(s). Each time for a connection, from a particular profile – and information about the specific request – a xsl transformation generates a xml file which is subsequently deserialized to a C# object for actually dealing with the request through connection et cetera. The option of using the information roundabout for writing information to some database, rather than just reading from it, has been left out of that introductory text. The transformation leading up to a C# object for writing (also read: performing updates) adds diffgram.

### **71.19**

Several times I have reread your message, i.e., the one in which you suggest some “application[s] for Metapattern.” When it is an integrated order that you are after, and of course I believe you should at least conceptually at first, it is illogical to start from the assumption of different applications. For what I have suggested as idealized design, a general idea about approaching design borrowed from Russell Ackoff, with Metapattern you can just make a start more or less anywhere. Soon enough you’ll discover meeting up with what traditionally you might have kept conceptually apart. I am still just guessing here, but doesn’t modelling variety constituting “member status code” – also – lead you straight away into the thick of “dealing [with] types or kinds of things” including “tags and labels” for their description? I can imagine that modeling for integration initially comes across as counterintuitive. You often need to consider and order concepts upside-down (actually: downside-up) when compared to modeling for a separate, small-scale application. I would be most happy to help you get underway. We could continue with my no-doubt clumsy, highly insufficient elaboration of member status code. What do you find wrong with it? What needs to be included for disambiguation, not only across all currently relevant systems, but for what you expect for the future, too? Don’t try to be exhaustive, though. Let me know what here and you find especially missing. Then, from one concept comes another, and so on. You’ll quickly see an overview developing ...

### **71.20**

You are actually doing me a huge favor. :-) For you are bringing up themes for testing Metapattern’s practical use, and so on to going over its assumptions. Of course I should be able to provide sound explanations, thereby also helping myself to additional understanding. When, indeed, Metapattern is a qualitatively different modeling method, there is a lot of ground to cover. Or, rather, depth to reassess. What I have learned from De Saussure’s work is more the general direction he points out. You’ll find some references throughout my book [Semiosis & Sign Exchange: design for a subjective situationism, including conceptual grounds of business information modeling](#). Also there, especially so in chapter 2, I find Peirce already more practical for the ... purpose of conceptual modeling, as he makes sign explicit as an intermediary element. And, of course, it is precisely this element we must now come to grips with at an unprecedented scale through networked digital technologies. Therefore, it is Peirce’s rather than De Saussure’s basic idea that I have extended for supplying Metapattern with proper axioms. Does this mean that I recommend reading Peirce’s work? No, at least not beyond his brief statements on the principle of semiosis. Apparently, Peirce had other interests. The directions he

did take his work before and after even, in my opinion, distract from approaching a modeling method for integrated order. Nevertheless, even when he himself did not follow it up, he made what I recognized as a crucial suggestion, and I gladly honor him for it.

What I will await patiently, is what turns out as what you mean by “tagging” et cetera. I would like, though, to comment more generally on, as you wrote, “consider[ing] the same things from multiple overlapping contexts for multiple overlapping purposes.” I apologize for the bluntness of my statements.

Then, what you call purpose I would consider motive as one of the ennead’s elements. A subject, that is, ‘something’ equipped with cognitive capacity, entertains motives. Through a focus, and in so-called Gestalt fashion, a motive is the necessary background for a concept, which is irreducibly foregrounded. Through a concept that is motivationally differentiated, a subject can behave in a situationally differentiated manner. As Peirce suggests, a sign mediates between a subject’s motivated concept and situated behavior. If so, a sign must be differentiated accordingly. And that is what Metapattern helps to express methodically at any scale. (Historically, I thought of such a modeling method first, to supply it only later with an appropriate theory. Isn’t that how innovations works? Idea first, rationalization after.)

A model is a sign. So, a model doesn’t include purposes. Or motives, for that matter. It can only include **descriptions** of purposes and/or motives. As such, they appear as contexts.

Now, the purpose :- ) with Metapattern is to eliminate “overlapping contexts.” For it is only without overlap, that a subject’s (also read: an object’s) situated behaviors – or motivated concepts – can be taken for unambiguous descriptions (vice versa).

Please accept that as yet I really have no idea what this has to do with “tagging,” let alone “tagging algorithms” and “an ingestion engine.” This “ingestion” part rather sounds like what an information roundabout is about (when covering both reading and writing). I am hoping and anxious to find out ... Here, I am only trying to elaborate from Metapattern’s assumptions.

At the risk of foolishly misunderstanding the example you have mentioned referring to what-is and what-is-not a chair, I have a visualized answer :- ) to offer. Please let me know when I have missed your point.

As I am familiar with the need for turning nodes downside-up, I start the model by including a classification scheme. For its structure. I have assumed a homogeneous hierarchy. By that I mean that to a keyword at one level, keywords at the next level may be connected, and so on. Then, a particular classification term entails the ordered set of keywords up to the – in this case – horizon. The structure of such a homogenous hierarchy is abstracted into a single symbol, as in figure 1.



figure 1

At the scale of integrated order, nothing is classified a priori. For idealized design, anyway, we therefore have to start with ... something. (Such radical abstraction will probably not be practiced later on, but at an early stage of design – which you might call analysis – I find it helps to avoid conceptual preoccupation.) In figure 2, the concept of something is added.

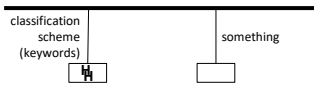


figure 2

Now, for describing situationally differentiated behavior, it is attributed to a situated object (also read: subject). So, instances of something must first of all be typed as either object or situation, as figure 3 suggests. It also shows the next step, that of constituting – instances of – situated object.

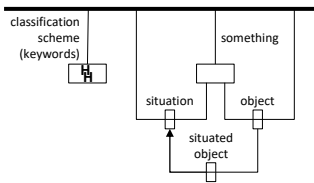


figure 3

It is only for – an instance of – situated object that at the scale of integrated order – one or more instances of – a type attribution can be made unambiguously, see figure 4.

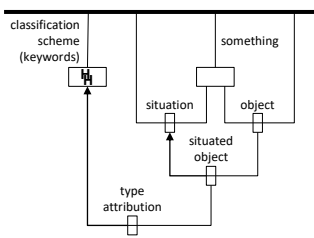


figure 4

From the type attribution, differentiation as relevant can continue.

I am sure that you see I am enjoying myself drawing up such models. I expect you to instruct me where I might already have gone wrong for your “system integration problems.”

## 71.21

Thank you for inviting me to the Metapattern folder you have set up with MindManager. I subscribed, but after some inspection I am afraid there I cannot be of any help by editing what you might post. As I see it, so-called mind mapping may help to associate concepts, but it doesn't provide control for disambiguation required for an integrated order. For it simply lacks the overriding contextual constraint. Somewhat similar to ORM, indeed when adding that constraint, it more or less turns into Metapattern, too. However, the paradigm shift away from atomism towards interdependence is paramount, and cannot be demonstrated, supported et cetera continuing to use what traditionally serves small-scale, stand-alone applications. I hope you appreciate my principled view.

What I can help you with, and am certainly most keen and motivated to do, is to offer explanation of Metapattern and to draw up conceptual models with ... Metapattern. And I am quite comfortable with the – extremely limited – drawing approach I now use. I find it serves its purpose.

What Metapattern from its principle of contextualized disjunction (!) does not assist a modeler with, is setting up a conceptual structure that might be used for – logical – inferences. Not subsequently, that is. I now seem to recognize that such a structure is what you are still after for “tagging [...] dynamically.” Am I right to suppose so? Anyway, I believe it cannot be done, period. It is not how language functions, let alone separate nouns. Ludwig Wittgenstein was already on to this by arguing that, as he called them, family resemblances are such that no property is shared by all family members. In my previous message, I actually forgot to be blunt, but now I should. :-) Please forget about essences, prototypes, core concepts, or whatever you want to pre-set as generally valid and allowing you to make inferences from. It is a dead end. Always has been, for that matter. For a limited scale you might still keep up the illusion that it works, but for an integrated order at any current, let alone future, scale it can only fail.

With Metapattern, you don't go for what is generally valid, but instead specify the always limited reaches of varied validity. That is why context – situation, really, but here I won't bother you with enneadic subtleties – constitutes the reach for what is taken as valid about an object. Then, 'within' a particular context, what is described for an object is all positive. There is no need for negative statements (also read: propositions). For what might be seen as negative about an object's description in one context, might be again positive about it in another. So, stick to the positive. However, it also follows – talk about logic :-) – that between contexts there is nothing to draw traditionally logical inferences from. In this sense, Metapattern is radical in its approach to so-called pre-coordination.

With all relevant contexts made explicit, and a human modeler is indispensable for proper interpretation, 'afterwards' it 'only' becomes a matter of identifying the relevant context (rather than expect some 'logic' to perform the interpretation). The only way to find out which contexts are relevant, is to draw up a conceptual model for what seems fit as integrated order. And the only reasonable approach is to model, say, operationally relevant subject matter. I don't think, when I may say so, that contextualizing something into a chair, or not, is particularly relevant for the “system integration problems” you are asked to help solve. I am not at all against theorizing, but it should always be practical. Your chair example lacks criteria for deciding one way, or another. I admit I fail to understand what problem might be solved by the categorization of (non-)chairs you've suggested. Why not right away aim at solving the both urgent and important problem of member status code? You are knowledgeable of the field(s), so you yourself can provide an immediate check on what is, and what is not, relevant.

In fact, I abhor exercises that are implicitly general. For nothing is general, everything practical is limited in relevance. Therefore, the equally practical task we face is to order such relevant differences without ambiguity. Forget formal logic based on atomist assumptions of general, i.e., context-free, validity. Beyond the most limited scale of application, it is nonsense (and also involves a counterproductive simplification of noun use). To put it mildly, I agree to “ignore the automated aspects of tagging.” :-) If that is what you are mainly aiming at, after all, also Metapattern cannot help you (as, in my opinion, nothing can).

Integrated order requires a qualitatively different logic; a key (meta)concept is nil-identity for connecting an object's contextual identities (each giving access to a positive description of that object's behavior 'within' the corresponding situation).

When attempts at solving problems continue to fail, at some point in time it seems reasonable to question basic assumptions – often implicitly – held so far and methods – often routinely – applied so far. When trying different assumptions et cetera, of course it is extremely difficult to let go of the previous set. I strongly recommend starting to model differently what is familiar (and it is what I can assist you with, too :-).

#### 71.22

I am standing by!

#### 71.23

Why does Metapattern, apart from a necessary boundary condition, that is, a horizon, provide just a single modeling construct type, i.e., contextual differentiation? The assumption is that there is no a priori limit to what requires differentiation. The construct type(s) must therefore be as free as possible from determination in order to optimally facilitate aiming at unambiguous actual constructs with modeling. And with what is not there, you also cannot go wrong.

#### 71.24

Upon my return, a few days ago, I found a copy had arrived of **Insurance Handbook for the Medical Office** (Elsevier, 14th edition, 2017) by Marilyn T. Fordney. Thank you again for supplying me with the reference. What I have so far learned from it as modeling goes, is what for now (!) may – and should – be productively ignored (which, luckily, is almost all of its contents :-). Please note, at this stage a more important lesson I could not have received; there is the danger of becoming paralyzed by an overwhelming amount of details.

Of structural relevance, and therefore deserving priority for modeling, seems to me to be a variety of even conditionally recursive relationships between, calling them by a general term, actors. For ‘surrounding’ – in the senses of facilitating, constraining, et cetera – some, say, elementary interaction between a receiver and a provider of care both they and others may be(come) involved in various capacities, with each differential relationship (also read: situation, to be modeled as context) serving to make contributions unambiguously explicit.

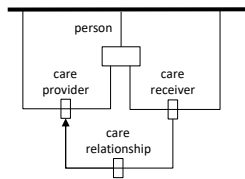
I do hope I don’t sound too cryptic.

Indeed, for medical care such varying sets of differential relationships seem often quite elaborate, but I would say its (meta)structure is not categorically different from how behaviors are coordinated throughout society. Considering such relationships I don’t see any basic difference with, for example, how legal process is organized. For me it follows that the idea is to first of all concentrate on designing an as generally applicable model as currently imaginable of – the intricacies of – social interaction. It may sound like a counterintuitive move, but when it succeeds many traditionally evolved problems with specifically targeted information systems simply dissolve. The ‘case’ of medical care certainly serves as a great source of inspiration (and taking it as an ‘example’ guarantees that the model is relevant for it).

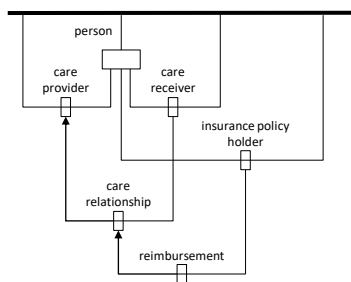
I have already been doodling a bit, too. There is really everything wrong with what I have come up with, but you may get a taste of the modeling approach. I have started from – the type for – **person**, deriving from it as differentiations **care provider** and **care receiver**. Subsequently, a **care relationship** can be established between care



provider and care receiver. Please note that this model thereby also covers self-medication.



Now, a person may be an **insurance policy holder**, providing **coverage** for her/him being a care receiver. And from such coverage follows **reimbursement** of the particular cost of her/his – side of the – care relationship.



Of course I am aware that I am only displaying my profound ignorance of administrative procedures for medical care. Again, I am just trying to demonstrate that the answer always (!) lies in necessary and sufficient differentiation (and that is what Metapattern methodically supports). A great many of false starts awaits us before we might say that we have adequately solved this large-scale modeling puzzle. However, without trials and errors there is no way to discover what might actually work for such intriguing variety.

### 71.25

For now :- ) I am in, say, half-(dis)agreement with the analysis of “ha[ving] defined member status 'states' differently” coming down to “a simple communication error between two application systems.” Are you sure there is no ground for it? It could be that different policies are managed differently, involving a.o. different sets of relevant states. If so, as long as such differences are maintained, they should be kept explicit (through pertinent contexts). However, when such differences turn out, indeed, to be irrelevant for whatever purposes, of course Caroline and you are right that Metapattern is superfluous (to solve that particular problem, that is, but of course not for pervasive “variations in meaning”).

As far as “simplify[ing ...] models” is concerned, my approach would be to start from conceptually widening the scope. It certainly looks like making problems more complex to solve rather than simple. But that is only apparent. What matters is the extent of relevant interactions. In order to understand how insurance ‘fits in,’ encompassing social practices should be modeled. That way, interactions can be supported seamlessly across domains previously kept separate through limited/-ing applications. Now, that is really simple and makes the initial effort of seemingly complex modeling pay off; in the process, you’ll have also sorted out “the underlying data modeling patterns” to facilitate, when possible, exchanging information between databases structured accordingly (also see my previous remarks on an information

roundabout).

With practically unlimited scope for a model, nothing may be kept implicit. My impression from popular literature on conceptual modeling is that, apart from a technology bias, an organizational bias prevails. That is, the idea is that the resulting model should serve requirements of some organization (rather than aim at whatever interactions, i.e., regardless of variety of – affiliations of –participants). With pervasive networking capabilities available, this organizational bias has become obsolete, and a risk to an organization holding on to it.

For a – much – broader model, an organization should not start from its ‘own’ business, but make conditions (also read: contexts) explicit for it, as I said above, to fit in. And from an overview, somewhat paradoxically, a particular organization may then consider how it might support more of the interactions (also read: value chains). Taking advantage of networking opportunities, of course an organization should be first to practice widening the scope of modeling.

Another point I find worth mentioning is that strictly conceptually there should be no difference in modeling for operational information systems on the one side and for a so-called data warehouse on the other; eventually, no separate data warehouse is required.

Anyway, I myself am from a social equity point of view interested in trying to model the varieties of health insurance as an integral part of a social whole. It explains why initially I don’t – seem to – address insurance issues at all; instead I am sort of experimenting with making, say, necessary and sufficient preconditions explicit for including it in an integrated order.

Thank you for enthusiastically mentioning the possibility of “a viable commercial product” based on Metapattern. However, conceptual modeling for an integrated order such as I envision requires human creativity. With contexts lacking, so-called artificial intelligence wouldn’t ‘know’ how to differentiate responsibly, while as soon as contexts are properly included no intelligence is artificially required. There is no replacing a good design(er) or, as you would probably say, a good analys(t).

I would like to add that I am not starting my business, but trying to tie it up. It has always been about self-funded research (because I didn’t succeed in convincing any business or government organization, or academic institution for that matter, of the need for a shift in scope and consequently in both paradigm and modeling method for information exchange). I self-invested in software development only for the purpose of demonstrating the new theory in practice (but even that didn’t help). So, I should already be retired but, apart from my pension being far from sufficient to live from, cannot help :- ) to continue my ‘research.’ As my voluntary contribution I am happy – and motivated – to try and figure out a socially – more – comprehensive model for structural relationships including those for administering health care. It keeps me busy, and the models I am designing may somewhere sometime at least serve some educational purpose (if only for myself :-).

## **71.26**

Meanwhile, I am afraid I am still at the stage of modeling-around-in-circles, that is, muddling, really. It shouldn’t come as a surprise given such a comprehensive scope. Anyway, no need to get frustrated, not yet. I’ll continue trying out different concepts to start from, and see how far they may take a conceptual model of/for a somewhat more integrated order. At least it helps to find out what doesn’t work.

## 71.27

When aiming at the widest imaginable scope, starting from concepts associated with some – traditional – domain must be avoided. For boundaries between domains can actually not be kept up. Medical health care, too, is interwoven with all sorts of other ... What? Whatever.

A most generic way of selectively behaving in reality is, I would say, to distinguish events. Can it be done objectively? No. In fact, recognizing bias explains the – need for, ultimately – administration of formal justice (which is of course biased in its own way, but at least it – usually – helps people to get on with their lives). See figure 1 for a model according to which different actors may come up with different reports of an event.

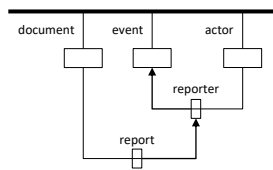


figure 1

With only a minimal addition, this model also applies to administration of justice. Parties to a conflict all – have to – submit their ‘versions of an event.’ As such, relevant documents largely ‘make up’ the event of the court case. This is shown in figure 2; what Metapattern considers a constitutional relationship is assumed between document and event.

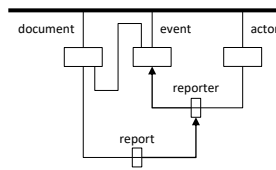


figure 2

For a court case as an event, the judge counts as an actor, too. Her/his report is the verdict. Then, executing the verdict is yet another event, in its turn more or less constituting it. What happens ‘in’ a court case, is no longer modeled separately on the basis of legal terminology. It is more generally positioned as an element in a network of events, itself now simply being one of such events.

I emphasize that nowhere in figure 2 it is made apparent that the model is about administration of justice. Set seamlessly in a wider scope, instead, administration of justice is more comprehensively facilitated (as is whatever is seen both leading up to it and derived from it).

I believe I am on the right track with a similar, say, non-explicit model for medical health care (from which may, after all, originate legal procedures). So, to be continued. Your comments of a more philosophical nature befitting this earliest stage of conceptual modeling are of course most welcome!

## 71.28

I wouldn't be surprised when you think that I have gone mad. Have I completely lost sight of your actual problems? What's the relevance for – digitally supporting – administrating health care insurance?

What I am trying to capture is some general characterization of social dynamics. My idea is that ‘something happens’ which is then provided with the status of consolidation by making a record of it, i.e., a document. In the ‘form’ of a document, that is, as something static, dynamics 1. come to a momentary halt and 2. are thereby given a hold for continuation (while limiting the need for backtracking when what ‘happens next’ goes wrong by securing an audit trail). Such, say, alteration facilitates order when the scope of interactions extends beyond ‘familiar’ contacts. It therefore exemplifies the transition from prehistory to history. So, nothing new, really. It certainly doesn’t change with the availability of digital technologies. However, such technologies do greatly accelerate event-document dynamics. Regrettably, we remain largely confused about their optimal use as long as we fail to miss the principle for historical order in our society. It is this principle that we have to properly grasp, first of all. Shown much simplified as a chain of single instances, an event (see figure 1.a) may be considered to include its recording (see figure 1.b), with that record/document occasioning the next event (see figure 1.c), and so on.

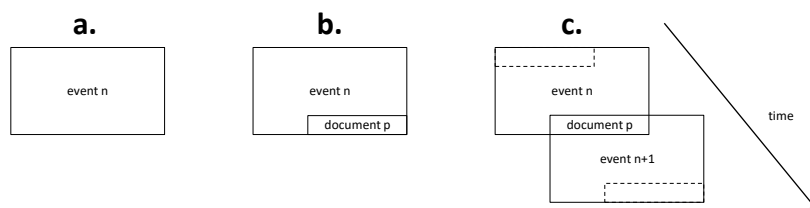


figure 1

In the occurrence of a particular event, whatever subjects and objects may be involved in whatever ways. I apologize for my contrived, if not outright ungrammatical, language use. For the intended scope, I am most likely putting a strain on English-language idiom, with English not being my mother tongue in the first place. Anyway, taking a cue from figure 1.b, in figure 2 I have indicated such varied involvements.

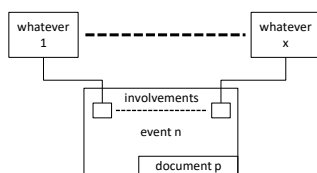


figure 2

Traditionally, that is, with paper-based documentation and the likes, the document was the only record of the event. It therefore had to contain a description of all – what count as – relevant involvements. This changes when one and the same ‘medium’ is used both to – partly – facilitate the actual event and to provide a document about it. Much of the latter is then ‘in fact’ already supplied by the former. Still from a traditional archival perspective, the former is labeled metainformation (more popularly known as: metadata). As the set of relevant information acquired during the operational phase of the event increases and (!) remains accessible to properly authorized actors (more about actors soon), what the ‘document’ – still – needs to cover diminishes correspondingly. Another way of putting this is that a

reference is sufficient under the denominator of document; its contents are distributed but may be grouped on demand through their relationships. Next, I'll elaborate on actors. To be continued.

### 71.29

I agree with what your “French acquaintance” says in encouragement. And I am of course most happy you are welcoming what develops as “a fresh look.” It would indeed be great – and hugely important, beneficial and so on – to get the need for conceptual modeling acknowledged.

A saying in French that I have learned to keep in mind especially for modeling at the scope of integrated order reads: *reculer pour mieux sauter*. That is, only when you allow for sufficient space, and even far more space than may seem warranted at first, you can make a proper run for it and jump much farther (and in fact land where you need to be, and quickly at that). In English, perhaps a saying like ‘look before you leap’ comes somewhat near.

The reason I am offering contributions in a piecemeal fashion is that I am collecting and reworking some modeling work previously documented in Dutch. So, I am not just translating, but at/for every step considering (also read: imagining) whether or not it is actually taking us where it can now – also – be useful for your requirements. I'd like to postpone a reaction to your remarks on “various players [with] overlap[ping] roles” being involved until after I have included the concept of actor to the – developing – model. It seems you are getting ahead of me, :-) so now I have to catch up.

### 71.30

An event is – understood as being – shaped by behaviors of assorted ‘whatevers.’ On account of attributed behaviors, they may be called actors. Through involvement with a particular event, an actor is a persona, i.e. the “player” of a matching role. Whereas in a previous model, some such persona was presented as the concept itself, reporter it was there, here in figure 1 ... whatever personae are eligible for an actor (and in the capacity of any persona can an actor record – his/her report of – an event: document).

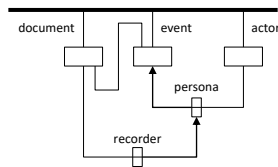


figure 1

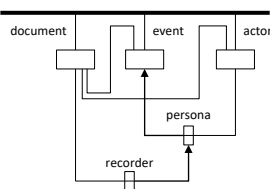


figure 2

I have already explained the nature of a so-called constitutional relationship. It makes for infinite re-use (also read: re-appearance) of instances following the same structure (which can thereby kept radically limited; clearly, huge benefits derive when

technical implementation can be correspondingly minimalist). In figure 2, a constitutional relationship is added between document and actor.

Actors are often constituted through sets of complementary instances, with all of them relevant for a more integrated order. Traditionally, an information system is paid for et cetera by some organization. Conceptually it then habitually disappears from view. The particular organization, or even part/function of it, is implicitly taken as the core of relevant meanings. For example, from the perspective of hrm a person is an employee. For an integrated order, however, it also needs to be made explicit that from the perspective of that person some organization is her/his employer.

With object-role modeling, such respective roles are combined to form a relationship. In actual fact, I would say, there are different personae at stake (for that is precisely why they were constituted as such; but only widening the scope can bring tis out). Each persona can subsequently participate 'more or less' on its own in all sorts of events, only when need be – through yet another event – brought into explicit involvement – again – with the persona, or personae, it was constituted along with.

My idea is that the varieties of medical health care insurance are a most relevant case in point. I am not finished with explaining the concept of actor, though.

Being eligible for one kind of actor, carrying the potential for actually-acting-as-a-persona, is often dependent on already being another kind of actor, and so on. If you want to practice as a medical doctor, you first of all need to be certified as one.

Earlier, when you want to enter medical school, you must have passed high school successfully.

And, say, actorial – is that English? – differentiation in our society mostly follows 'taking part in some capacity,' usually functional, in another actor. Take person x and organization y. Thus, there are the actors x and y. When they enter into a job contract, two more actors are 'derived:' x-as-employee-of-y and y-as-employer-of-x. I can only emphasize that managing an integrated order required such explicit differentiation. It may not be simply assumed that y-as-employer-of-x covers x-as-employee-of-y.

In figure 3 the conditional differentiation of instances of actor is expressed, too. You may remember that I am favoring a single symbol for an open-ended homogeneous hierarchy. As I believe that most classification schemes are structured as such a hierarchy, that is how I include them in a model. Please note, as the model is about types, that any number of actual coding systems, or whatever, fit (as long as for every scheme its unique name, or whatever, is included as its top-most element).

Then, for some actor, choose the relevant factor for differentiation; in the example of x, this is as-employee-of. Next, chose the actor acting as co-actor, in this case y, yielding x-as-employee-of-y to constitute yet another instance of actor.

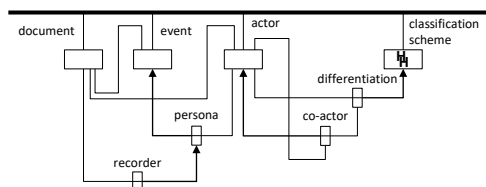


figure 3

Of course, it depends on the actual scope of facilitating interactions how many of such actorial constitutional cycles, with documentary 'evidence' for each set of

resulting actors, should be made explicit (and available to authorized users). My contribution is to make sure that structurally it doesn't matter. 'Whatever' is required, must always fit. Any change in scope, and thereby relevant instances and so on, should be accommodated with the same structure for variables.

In my naïveté I have thought of two examples as extremes. When I am abroad in need of medical care, I can visit a doctor whom I have to pay then and there; I ask for and am issued a receipt; upon returning home I send the receipt to my insurance company – hoping? – for reimbursement.

How different in terms of relevant actors from where and when I enter as a recognized insuree a medical institution which is under contract of 'my' insurance company.

I see a host of possibilities though relating to classification elements from other conceptual nodes in the model of figure 3, too.

A classification of actors should help. Suppose someone is bitten by a dog, let's call her/him z. Then, through z-is-pet-of-x ... For relevant examples I prefer to imagine the weird and wonderful; when they fit, what's thought of as normal should then pose no problem.

And what about respective classification of both events and personae? Combined they may provide necessary and sufficient parameters for determining the type of document to be 'filled out.'

Well, I am just guessing from ignorance. That is where I find that the book you've recommended comes in. It may be read as very detailed specification of requirements (although I am afraid it doesn't go beyond 'normal' :-). Does the model of figure 3, sketchy as it is, hold up? Where does it fall short? Could it be adapted? Then again, in several ways, it already offers even far more ... On the surface of it, you would never know it is about medical care insurance. However, from an integrated-order perspective that is precisely the critically relevant quality (but difficult to convince stakeholders of who might still expect only their own stakes to show).

To be continued.

### **71.31**

How I have suggested the concept of actor now helps me to remark, please note not more but less :-) specifically so, on your emphasis on recognizing "various players[,] the[ir possibly] overlap[ping] roles[,] and so on [leading to] different perspective[s]" as far as such variety goes. What I simply take from such complexity, is that even when there should be some closed-off domain we could be limiting ourselves to with a conceptual model, which in fact there isn't, the set of relevant roles is very large. How can we be sure of having drawn up an exhaustive list of relevant roles? And then, roles may change over time. When roles must practically be seen as elements of an open-ended set, there's of course all the more reason to treat as open-ended how "players" interact according to their roles for some event.

So, I take your examples as confirmation that some specific role should not be modeled as a separate type. For it would lead to an unmanageable multitude of types while being uncertain whether or not all relevant roles are covered by such types. And then there would be the nightmare of – trying to – establish pertinent relationships between them.

There's really no solution in approaching such variety half-heartedly, that is, assuming two classes of roles with its 'own' type (also read: concept) for each, say, primary role and all so-called secondary roles brought under a collective role-type.

Only radical generalization helps. A specific role is not modeled as a correspondingly specific type, but a general type is modeled allowing a specific role to be described by some value of it. The model comes out deceptively simple. However, it affords an open-ended (dis)play of interrelated values.

Please note the shift. From a compact model, so-called software quantitatively amounts to 'not much,' too. How digital processing actually proceeds, becomes to a much larger extent determined by values (as such, also read: parameters).

Management of parameters is critical (which should also be seen as 'programming'). With a sufficiently generalized concept of actor, at least in that respect boundaries between domains dissolve. I mentioned a dog biting a human who would subsequently need medical treatment. But fitted with appropriate 'values,' the model with its radically generalized types-for-concepts can also be made to work for animal health care. Indeed, "and so on." And I find it has lost nothing in its possibilities to help facilitate, for example, administration of justice, on the contrary.

What drives widening scope are what I have called constitutional relationships. It makes the – structure of the – model recursive both syn- and diachronically (concurrently and for temporal sequences). Recursion is the miracle cure for structural minimalism.

A critically important point is to not only include values for the variety of the 'other' in interaction. Instead of keeping "our firm" implicit, the 'self' must be equally differentiated for balanced support – at the scale – of an integrated order. Especially the latter removes the prime obstacle for optimal integration of services, as you no longer try to reconcile (also read: standardize) values that 'really' need to remain different (and you may come up with other beneficial roles for the firm, then more easily adopted, too).

Of course, a compact model doesn't make relevant details go away. It should make them far more manageable. What certainly appears "daunting in its complexity" may upon closer inspection to a large degree consist of a collection of all sorts of specifications. If so, for each 'item' the question is what it is supposed to specify. Following the maxim of 'a place for everything, and everything in its place,' the answer then lies in supplying the aimed-at 'what' in the overall model (which from such a necessarily generalized model will most likely be some value for some concept/type).

Am I still making some useful sense to you?

### **71.32**

Please, take your time "making sense of generalizing type." I really appreciate you are considering what might seem a counterintuitive approach: radically widening the scope puts heavy pressure on limiting types (or else you'll get swamped by them). It is really what mathematicians do when thinking up functions; there is only one algorithm for addition, abstracted from whether you are counting apples, pears, or whatever.

As you yourself instructed me right away at the earliest stage of our correspondence, changing to a de facto limitless scope certainly is "a hard case to make, especially in the US, where 'rugged individualism' is the cultural norm." :-). Especially try to run most unlikely thought experiments through the model, as I am doing for example when riding my bicycle to where I have to go, and back. And don't hesitate to submit to me cases you feel should fit. I hate to be wrong with the direction the model is – at



this stage, only – pointing at, but if so, the sooner I am cured from such a serious mistake, the better.

### **71.33**

I hope [your] colleague] hasn't been scared off by Metapattern. :-) [And] I would say there is no rush. On purpose I've tried to suggest a ridiculously generalized conceptual model. Thank you again for bearing with me. I am sure you want, and therefore should, move concepts and/or relationships around. Please do! At this stage, i.e., of conceptual modelling, I find the challenge is not at all technological (if ever). So, it doesn't look much like work, not what most people consider real work, that is, because you are not seen to be busy building et cetera. However, what (meta)conceptual compactness remains valid after prolonged scrutiny – and it does take time for imagination to take effect; the back of our mind, whatever that is, is where imagination is most productive – provides exponential benefits. So-called constitutional relationships where some type is retyped into another type for – yet – another cycle of instances, and so on, help do the trick of recursion (which limits the model while expanding the possibilities it offers).

### **71.34**

I am most happy to join you in further exploring modeling.

### **71.35**

There must be a bright side. And indeed, there seems to be. For it might be a huge advantage that in some official sense you may start from, say, conceptual scratch. From my perspective, admittedly biased and all that, it sounds quite odd that “[y]our company has only done physical data modeling.” Of course, assuming it to mean – something like – database description, the person doing the describing cannot fail having some idea of what s/he is preparing the database(s) for to contain information on in some accordingly structured fashion. So, in actual practice, to some degree or other there has always been done “conceptual data modeling.” However, mostly leaving it, and habitually without brief, too, to software developers hasn't always delivered operationally optimally structured data. I really do make an effort of sounding diplomatic. :-)

As a departure, and summing up my contributions to our correspondence so far, I would argue for conceptual modeling being practiced both explicitly and comprehensively at the scope of relevant integrated order (measuring that scope not in terms of technology et cetera, i.e. means, but in terms of social interaction, i.e. ends).

Connecting – the teams for – data governance and data architecture, great! I find you've made a mutually critical move: It is impossible to 'have' proper data architecture without conceptual data modeling; and then it is impossible to have proper data governance without data architecture, period.

As you have formulated it, the similarity is even obvious with what I am trying the Dutch Judiciary to adopt, that is, again, in your words, a data government policy including (!) a data architecture policy. An innovative employee is already convinced, but I am afraid he still has many colleagues to turn around. Anyway, for that purpose, I am promoting conceptual generalization as a modeling strategy such as I have also outlined in our earlier correspondence. As it is practically impossible to limit the relevant scope, ... the only practical approach left is to design/model for limitless scope.

Ready? I am looking forward to hearing soon from where you want to continue after “re-read[ing] our correspondence.”

### 71.36

I managed to find time for writing you a substantial message. At least, I hope you can appreciate it as such. I thought it might be helpful to you when I offer some direction for where to take conceptual modeling.

The bias of software engineering is, no surprise there, software. As any profession (im)matures, programmers et cetera habitually come to consider software as an end (and as such for them to determine unilaterally). However, what they are supposed to contribute to are tools, that is, means to what are ends for users et cetera.

I find neglect, and I am trying to put it mildly, of users and their information requirements easy to recognize. For it is immediately evident when a conceptual model is lacking. It is all the more difficult to get it addressed, because through substituting their ends for users' means, programmers and the like don't even have a clue what it is I am asking for. As I am not their boss, and as the person who is is often even more clueless than the programmers are, nothing changes. It should therefore also be no surprise that larger-scaled projects, i.e., where conceptual variety for users is the sorely unrecognized critical factor, fail to deliver even moderately adequate tools.

Thank you for enduring my outburst of frustration! :-) Again in recent weeks I have met even more extremely well-meaning individuals who in my opinion are simply extremely ill-informed about the relative nature of their contributions.

Of course, you cannot do conceptual modeling without deep understanding of what both is and is not impossible for making 'it' into a tool. But then, conceptual modeling is about acquiring – and documenting – a deep understanding of reality first. Such a model explains what the relevant information is and/or should be unambiguously about. Understanding is served by as much as possible abstracting from whatever technology might be applied for actual information management. In fact, the choice of technology/-gies can only be responsibly made on the basis of the conceptual model, that is, later.

As a general rule, don't pick a supplier when you still don't know what you might need. And first of all don't commission a particular supplier to help you discover what you need; you'll get proscribed what suits her/his needs. When you feel uncertain about what you want, seek independent counsel. That is why, in **Metapattern: context and time in information systems**, I stated that “[c]onceptual information modeling [...] is now an important profession in its own right.”[p. xix] So far wishful thinking on my part, I admit. But the obstacles to achieving such a status for conceptual modeling don't make it irrelevant, on the contrary.

It might help to see the conceptual modeler producing two different kinds of models. One – and the first to draw up, iteration for the time being ignored – is, as argued above, as purely conceptual as possible. It is about meanings of information of and for users.

When users agree that their (!) relevant meanings are being catered for properly, unambiguously, exhaustingly and so on, another model may be derived from the purely conceptual one. It is aimed, not at users for their understanding and approval, but at programmers et cetera for subsequently developing the actual tools.

This is not different at all from what we are familiar with for the so-called built environment. When you are not a (building) architect yourself, you engage one for

designing, say, a house. Upon your approval, 'your' architect draws up specifications for one or more building contractors (and in their turn they may elaborate such specifications for their particular contributions under the original architect's supervision on your behalf). In this analogy, for digital information systems assignments are still right away given to programmers-as-building-contractors. So far, no architect is involved 'in between.' Adding to the confusion is that programmers now often call themselves architects. Indeed, they do design – well, who doesn't – but they do so from a tool, rather than from a user perspective. And users, knowing no better, only pay attention to the job title, assuming that what will be produced is done from their perspective. Wrong!

I apologize, I am again getting carried away.

What I hope to have made clear, is that concepts should first of all be modeled without any consideration of how information-about-them should be facilitated by technology (with, for example, pen-and-paper of course deserving to be called a technology, too). When conceptual variety is at stake, there really is nothing more practical than suspending actual construction until after it has become clear enough what to construct. I am all in favor of trial and error, but without design-as-plan merely error results. You would mistrust a carpenter who would right away start hammering planks together, and even most confident and happy to do so, for constructing your new house.

Earlier in our correspondence I have suggested actor as a crucial concept to focus much variety of social interaction on. For the Dutch Judiciary a start is upcoming, I hope so, anyway, :-) to interconnect legacy databases – with everybody relieved not having to tamper with them – via an information roundabout with an added database supplying necessary and sufficient information keys with, where applicable, relationships between them. I believe the scope of what you aim to cover as an integrated order certainly also warrants its key importance, of such an actor concept, I mean. I'll therefore re-iterate (and in the process present a somewhat modified conceptual model, as I have done for the Dutch Judiciary, to accommodate later enhancements).

As there should be no technological constraints holding us back for conceptual modeling, why not take an all-encompassing perspective (which is of course a contradiction in terms :-)? So, there's a horizon and what appears as differentiated below it are at first just instances of something. Usually with Metapattern, only types are shown in conceptual models. You'll recognize the model as shown here in figure 1.

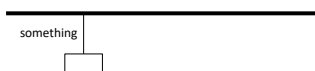


figure 1

Imagine that the instances of something are each uniquely identified, say with some number. Nothing else is assumed to be known about them **as somethings!** Of course, that is not much to go on. That is where classification comes in. I am assuming a straightforward structure for a classification scheme, that is, a terminological hierarchy (as a case of a homogeneous hierarchy). It will amply serve the purpose. Even one level might often suffice. Think of <person> and <organization> as respective instance terms. Then, for example something <87>

could be differentiated according to the term <person> to yield an instance of actor. For the model, as a departure from what I have suggested to you before, see figure 2; as I am setting up actor for recursion, I have reconsidered what to take as the beginning of series. Please understand that it is not a matter of being absolutely right or wrong. Conceptual modeling is to a large extent about varying assumptions, and seeing how far they will carry without contradictions surfacing.

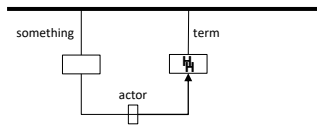


figure 2

Indeed, something <87> might also be differentiated according to the term <organization>, thus yielding another (!) instance of actor. Strange? When modeling concepts I prefer to postpone putting on constraints (and that is why Metapattern models at least look uncluttered). It helps to think: Why not?

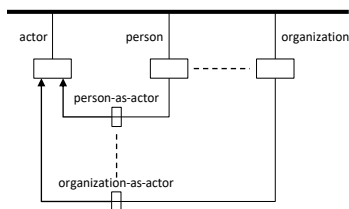


figure 3

I am sure that you'll appreciate that the model of figure 2 is an order of degree more flexible than when person, organization et cetera are typed separately as what amount to as actor types. For then, to arrive at a general type for actor, each a relationship would have to be included (see figure 3). And for changes in such actor types, the model would have to be adjusted, and no doubt more ... The more abstract, as types go, model of figure 2 avoids all that (but you have to use your imagination more to understand it; and at least for now especially refrain from objections of a technological, that is, software-engineering, nature).

You may also notice that the model of figure 2 allows for broader application. Why stop at actor in the sense of, for clarity expressed here as a pleonasm, active subject? For example, <table> could be available as a term, allowing for differentiation of whatever something instance into yet another instance of, well, let me call it factor rather than actor. Why not? Meanwhile, I stick to actor for labeling the particular model node.

The reason I am bordering on the pathological with my focus on actor – or whatever you want to call it, actor, that is :- ) – is that conceptual variety, touching upon and penetrating an ever larger set of social interactions, is otherwise simply impossible to control, including to adapt to, et cetera. The answer/solution is to assume wider-ranging variables, to be fitted with values as appropriate.

I did already draw your attention to the recursive nature of the actor concept. It is how rules for social interaction can be understood to differentially – allow us to – function. To qualify as a participant of some type in an event of some type, often another qualification must be fulfilled. For example, someone is only admitted to a

hospital for treatment on the condition of being the holder of a valid health insurance policy; someone is only allowed to treat patients when being certified as a health care professional. There is no telling beforehand where such conditional recursion stops.

Figure 4 shows a model for limitless actor recursion. The idea is that some actor instance is first functionally differentiated for which an instance of term is taken, too. For example, the actor instance arrived at from differentiating something <87> according to the term <person> is subsequently differentiated according to the term <policy holder; health insurance>. Next, the insurance company is brought into the differentiation. However, the relevant instance of actor should 'already' be available for that purpose. Instances for both functional differentiation and actor association supply necessary and sufficient values for yet another cycle of recursion. I repeat that I am primarily making conceptual assumptions. Do you see it differently?

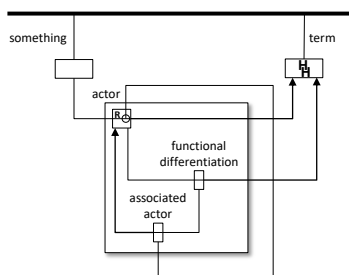


figure 4

The shaded rectangle is supposed to emphasize that recursion involves instances for all nodes inside it in order to establish a yet more differentiated instance of the, say, principal concept involved, i.e., actor.

When you are used to considering so-called applications separately, a provision for conditional recursion without limit may seem odd. But it really (!) is how behaviors of members of society are regulated. They are supposed to act in a variety of capacities according to circumstances. So, the more the scope of an integrated information order approaches all of society, the less escaping there is from explicitly facilitating such, say, actorial variety. In fact, the threshold for such radical abstraction being profitable, if not practically unavoidable, is soon crossed. Since the persons about whom information is registered can also be users in the current era of the Internet, authorization must cover the very scope of the integration information order. And users, in fact, are ... actors.

The depth of instantiation, and what values are relevant, depends on scope and variety of the actual integrated order that needs to be facilitated. Anyway, for the Dutch Judiciary the only viable approach is to avoid preset limits.

I am looking forward to continue our exchange. Do you recognize benefits from making this concept of actor the primary focus for ordering variety for what you want to achieve?

### 71.37

Overseeing moving data et cetera "to the Cloud" must require your fullest attention. All the more I appreciate getting a reply, even immediate, during such especially busy times for you, thank you.

Hurrah for actor! I am most happy to read what I take as your confirmation of my

critical suggestion, i.e., in your words, “making actor the primary focus for ordering variety [a]s an elegant way to model the complexity of the relationships among healthcare participants (persons and organizations interacting in a variety of roles, forming a complex network of relationships)[, ... with] limitless actor recursion [...] be[ing] the key to deep integration of our healthcare systems.”

Despite me strongly prohibiting you to think-things-technological while engaged in conceptual modeling, :- ) you are of course quite right to already wonder “how this would work.” I can reassure you that Martijn Houtman (Information Dynamics’ one and only developer) has made it, say, prototypically work some years ago. He is now upgrading it to an already less prototypical version for a small team of the Dutch Judiciary to use – free of charge regarding its experimental nature – connected to a so-called information roundabout (which we installed there previously, and has been in use successfully for several years now). Actor recursion structurally functions, so no problems there anymore, but how about the data administrator(s) actually optimally using it? Apart from streamlining the user interface, for variety control guidelines, if not rules, must be developed for classification terms, the order for optimal differentiation, and so on. There is still much to learn all around from that experimental experience. I am greatly looking forward to such operationally relevant tests.

The series of conceptual models I sent you in my previous message in fact lead up to an already more comprehensive conceptual model of facilitating integrated order where – physical – information resources (primarily read here: databases) constitute a federation. I believe such an organization, i.e., as a federation, to be unavoidable, and therefore necessary; there is no way that all pertinent information can be available in a single – physical – database. On the contrary, I would argue that the pervasive means for instantaneous digital communication promote distribution of information resources. Hmm, should I be writing this, now you’re involved in moving an “enterprise data warehouse from an on-premise database to the Cloud”? :- )

Where was I? I’ve drawn up that more comprehensive model for an additional database augmenting the integration potential of an information roundabout in Dutch. Translating it into English should be simple enough, with actor a Dutch term, too. So, soon I’ll send you the English-language version. As you have become familiar with Metapattern’s principle of conceptual differentiation, my idea is to refrain from comments, instead waiting for you to raise questions about what in that conceptual model you find obscure, in need of clarification, improvement, et cetera. Especially with conceptual modeling for integrated order there is – for we are living in a dynamic world, and only partially knowledgeable – no last word, so please express your constructive criticism and alternative conceptual constructs.

I hope, and expect, your Cloud project continues successfully.

### **71.38**

As announced, see attachment, took a few minutes at most. :- )

[See [Information roundabout: conceptual model for integrated-order management of information resources.](#)]

### **71.39**

The paper you’ve sent me a copy of, has nothing to do with Metapattern. The ontological assumptions of its authors remain largely implicit and do not go beyond a naïve atomism. Although they claim that

the ontology model can be directly related to the real world,

what actually continues to result is stubborn and blind commitment to technology. From such a predominant technological perspective, what people wrongly consider as a conceptual model is called an ontology. From an integrated-order perspective, though, ontologies in the plural are simply nonsense.

Ontology in the singular should be taken as the set of basic assumptions, only. You might call it a metamodel. Methodically applied, it should now facilitate modeling real-world variety; traditional atomism taken as such a metamodel/ontology is just not up to that task. The assumptions explicitly underlying Metapattern therefore recognize situational differentiation of object behavior.

A data warehouse, of course with due respect for the efforts it takes to set it up et cetera, :- ) is just another database. I have sent you a conceptual model for organizing keys (also read: links, references) to information registered across whatever set of databases. It is also published on the ww web, see [Information roundabout: conceptual model for integrated-order management of information resources](#). Repeating my reference, I'd like to remind you of what I recently put to you as a sort of request:

As you have become familiar with Metapattern's principle of conceptual differentiation, my idea is to refrain from comments, instead waiting for you to raise questions about what in that conceptual model you find obscure, in need of clarification, improvement, et cetera.

I do realize that you are extremely busy getting your company's data warehouse going in a new technical environment, so I don't expect you to react to my request soon. But since you yourself have almost immediately brought up the request for

help[ing] users discover, reuse and share [...] data [from] different business areas,

I can only urge you to consider my radically conceptual model first.

Yes, some time ago I have studied some of the work of Barry Smith, too. As I recall, it still suffers from premature orientation from – digital – technologies, too. While he does bring up relevant considerations, as a method it is far too complicated without unambiguously achieving adequate conceptual coverage. Please acknowledge that in this respect methodically less is integrated-orderly more.

Well, you did ask me to let you know my thoughts. Allow me to be especially blunt in arguing that there is no way to come up with a more encompassing design from extracting elements or whatever from a set of designs at smaller scale(s). At an increased scale, a design is sui generis. It therefore needs a creative human to produce such a more comprehensive design, or model.

Please take the model I am referring you to as an example. With abstraction and all, how could so-called logical inferences have produced it from a host of separate smaller-scale models? Impossible! Whereas you find the prospect "appealing," I can only warn you against ever undertaking such an effort. The "task" is not "daunting" at all; just forget about such "fun" completely. And should I fail to convince you, at least take a look at the integrated-order model I am referring you to, first, and let me know what you find right and/or wrong about it. And at this stage never mind developing tools (which I don't know anything about, anyway). It is first of all about understanding how concepts are interdependent.

## 71.40

You are very right, that is, I strongly agree with you, to prioritize “modell[ing] the complex [...] events underlying the [...] message exchanges.” Perhaps, however, I haven’t yet succeeded in making it sufficiently clear that a paradigm shift is in order. And I’d like to emphasize what I believe you can right away do about it.

Traditionally, at least that is how I see it, putting digital technologies to our practical use is about programming so-called applications, with information and especially what it is supposed to mean largely taken for granted. That is an outdated approach, as digital technologies nowadays facilitate instantaneous interconnection of information resources. The crisis in digitization is therefore due to people remaining blind to the both limited and limiting nature of the application programming paradigm. More of the same never helps when problems have become qualitatively different from what they were.

So, a major reason why I abstain from using the term data is that I associate it with the application programming paradigm. It leads many people to mistakenly believe that meanings are static and should be unambiguously related to unitary data. By implication they consider so-called formal logic as the paradigm case for programmed processing of ‘data,’ i.e., as if meanings don’t matter (which is clearly nonsense, but still not recognized as such).

Changing paradigm (also read: worldview, metaphysics, et cetera) is notoriously difficult, if at all possible for the person holding – hanging on to? – it subconsciously. It really doesn’t ... change what should by now be obvious. Overall interconnection requires the shift to an integrated order of information, with priority of attention to the – even dynamic – variety of meanings. Only such an integrated order provides the setting for programming (and not the other way around).

As I said, most people don’t recognize the need for the paradigm shift. In fact, often they are not at all aware they are thinking and acting from some paradigm. So, since we are discussing trying to facilitate information exchange regardless of scale limits, what matters to us here is that mostly implicitly they still follow the application programming paradigm expecting it to help overcome the crisis. Forget it!

When I am roughly right about those two different paradigms, the outdated one runs even counter what the later one could help accomplish. Otherwise they wouldn’t be different as paradigms go, now would they?! As a corollary, it is highly unlikely that the persons, say, mastering the old paradigm can be of much use getting the new one accepted and productively implemented. Don’t even bother to involve them, an advice you should take from Everett M. Rogers in **Diffusion of Innovations**.

Luckily, shifting to another paradigm doesn’t have to be an all-out occurrence pinpointed in time. Preferably not, even. But without making a start somewhere, it will of course never happen. In my considered judgment, setting up an information roundabout is a very powerful way to get the shift going. I haven’t tried to explain it to you at length for nothing. More recently I have followed my efforts up by providing you with [Information roundabout: conceptual model for integrated-order management of information resources](#).

I am happy to assist with understanding 1. the need for a paradigm shift, 2. Metapattern as the modeling method of choice at the scale of an integration order and 3. an information roundabout for getting the shift started both quickly and cheaply. Some of the project ideas you have suggested, however, to me don’t seem to fit this strategic approach For how I understand what you are after with those



'solutions,' the application programming paradigm is not really challenged; of course it is possible to use Metapattern, but putting it to work for the outdated paradigm would not demonstrate, say, added value and therefore not help build the integrated-order case. It is certainly not what I want to spend my time on, even unpaid at that.

You didn't write me any comments yet on the complete model I've recently sent you. As I find that setting up an information roundabout is critically important in a planned change sense, I'll try to give you some additional pointers to help you get going. I have already mentioned that the model aims at concepts for social interaction in general. As such, it should not only cover, for example, administration of justice and medical health care as special cases. Fundamentally, there are no such special cases. Motto: everything is integrated, because integration is really everything. Should you come to agree that the recursive actor concept as I've modeled it, fits your requirements, and more, actually, you might also conclude that the most difficult part of pertinent conceptual modeling has already been done for you.

From the model you can conclude that an event may have participants. You'll notice that participant is always a straightforward differentiation of actor (with event as the relevant situation). In, say, preparation of actual participation, much of so-called complexity is pre-coordinated by the – possibility of – articulated instance recursion of the actor concept. Then, an actor instance may be counted upon throughout as a potential, or candidate, participant instance, i.e., 'set up and ready' to become actually involved in events as (!) participant.

The conceptual model displays concept types; instances are left implicit. Let me take your instance suggestions, and run them through the model.

There is an instance of something:

m.

Differentiating m according to the term

natural person

yields

m | natural person

as an actor instance.

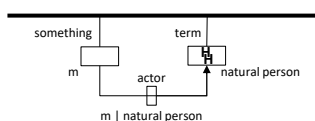


figure 1

In its turn, the instance

m | natural person

of actor can, here according to some instance of register (database), be differentiated into an instance of profile referral. As a property of such an instance of profile referral, the identifier

k

is available of the object corresponding to

m | natural person

in the actual register (database). As that actual register, while included in a federation of registers, is not included in the register for integrated-order management of information resources, it is placed outside the model's horizon for such coordination. Apparently there, through

k

as identifier, information about A as a (natural) person is available. Of course, from actor instance

m | natural person.

more instances of profile referral might be registered. Adding a classification sorts them out (to be modeled conceptually by simply relating the node for profile referral to the node for term, too).

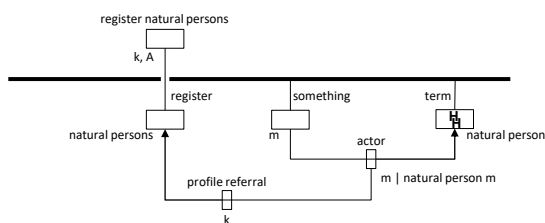


figure 2

Anyway, the instance

m | natural person

of actor should be taken as person A. Well, I am a doctor, too, but not of the kind you should accept medical treatment from. S/he should be properly certified. For the sake of your example I assume there to exist organizations overseeing such certification. In the manner of figure 2, another instance of actor could be

n | organization.

Then, actor instance

m | natural person

aka person A could first be supplied with the functional differentiation of doctor, and subsequently with

n | organization

indicated as the associated actor. Through recursion, a composite instance of actor results

m | natural person

doctor | (n | organization).

Starting from an instance of something, through a natural person, we've arrived at this person being a certified medical doctor. Please note that I am just making suggestions. There must be indefinite ways of going through cycles of instance differentiation for recursion. What I hope to demonstrate, is precisely such flexibility. I

am not in the position to decide upon which configuration of parameters might be optimal. I am only offering enough suggestions, so you may carry on.

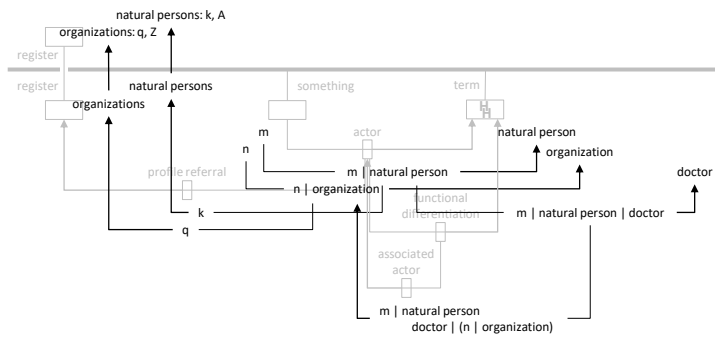


figure 3

I am sure you are starting to appreciate why conceptual models at the scale of an integrated order are type-oriented. :-)

As far as I understand what is meant by “rendering provider,” I wouldn’t bring it to bear on the actor concept. Following the conceptual model, an instance of actor

m | natural person  
 doctor | (n | organization)

may be declared a participant in an event. When participant(s) should be singled out as rendering provider(s), the relevant instance of participant must also be classified (as suggested above).

Am I right to suppose that some hospital may provide medical care from the perspective of a variety of health insurances of patients? If so, actorial differentiation of person A should in my opinion not be burdened with specifying under which particular health care program medical treatment is provided. Instead, I would add the hospital as a participant in the event in its insurance, say, capacity, corresponding with that of the participating patient. In a similar fashion, the patient’s insurance company and/or policy could be stipulated and added as a participant. When you get the impression that I am trying to force the real the variety in the types/categories provided for by the current conceptual model for integrated-order management of information resources, yes, you are quite right. Of course, now that I have drawn it up, I am first of all testing whether or not it will hold up. And when it does, why not adopt it? But of course you should execute your own thought experiments on the basis of it.

I repeat that the conceptual model is primarily oriented at facilitating an information roundabout. It could be expanded to cover the actual events, rather than being restricted to references. As with classifying participants, classifying events no doubt offers many additional possibilities, as does recursively structuring the concept of event (case), too.

Any actor instance may be supplied with one or more profile referral instances. For example, from actor instance

m | natural person  
 doctor | (n | organization).

a reference could be made available to the certifying organization's register with more information about person A's medical doctorship.

In order to give you more ideas, and with shortcut in notation, suppose two instances of actor have been composed as

person A | doctor | certifying organization

and

person A | employee | hospital employer,

respectively. Among these, then, it is possible to differentiate

person A | employee | hospital employer

as follows:

person A | employee | hospital employer

capacity | (person A | doctor | certifying organization).

And in that capacity only, A may be assigned to a hospital department, or whatever:

person A | employee | hospital employer

capacity | (person A | doctor | certifying organization)

assignment | hospital department.

From within some context, there is usually no need to display the full recursive development of an actor instance. Taking the last example, getting to

person A

from

m | natural person,

it is often sufficient when

person A [...] assignment | hospital department

is shown, with the possibility of viewing the complete instance.

Here, I am not going into the aspect of variety in time. KnitBITs as the software platform for Metapattern facilitates implementation of managing temporal changes of both instance values and type-nodal structure. Especially without the latter, you cannot maintain structural control at the scale of integrated order. And both are required for establishing audit trails of use.

I can well imagine that, even without considering temporal change, you find my examples difficult to take in immediately. That is why you have to practice yourself. The structural idea is to balance conceptual brevity with extension through parameters.

I am aware I won't be taken seriously when arguing to devalue standards for data exchange. As a longer term vision, though, I don't find it wise to continue to fill each other's data silos.

In my view, the answer lies with differential authorization. Initial messages, then, need only contain necessary and sufficient references, with further information selectively made available to such a duly authorized ... actor on a transaction basis.

My conceptual model already aims at such a future.

In the meantime, :-) in support of message standards such as HL7 or, in fact, whatever way you want to structure messages, the model should help you recognize, from a federation of databases, both how relevant information may be selected for message composition and how messages may be decomposed to distribute information over registers.

Rather than positioning message structure at the center for overall importance, its structure becomes supportive when considered from an integrated-order perspective. For message structure, too, should be derived from integrated-order structure, and not the other way around. Maintaining an orientation at separate applications is what has subsequently made message structure critical for practical coordination, and “incredibly complex” in the process. However, it doesn’t help to address variety of meanings. With the shift to a comprehensive model for integrated-order, separate applications as they have been traditionally set up, will disappear. Much complexity dissolves. Of course it does not ... mean the end of programming, but the start of programming from an integrated-order perspective. We remain lost while deepening the crisis without first of all drawing up a conceptual model at the real scale of exchange.

Regarding “a modeling tool” in the sense of a drawing tool, I first of all refer you to my remarks in my email message dated May 20th, 2019[; for its contents, see **note 71.4**, above]. The really only relevant ‘tool’ is the modeler her- or himself. I repeat, for drawing actual models, I find – programs such as – PowerPoint and Visio fully adequate. What could a more extensive tool possibly add, that is, to strictly conceptual modeling? Please recognize that – what I consider to be – conceptual models are not suited to generate programs from, et cetera.

Without a tool supplying the false illusion of control and preferably even seamless relevance for programming, a conceptual modeler is forced to focus on ... conceptual structure. Whatever tool should not distract or, even far worse, steer her or him. When “data architects and data analysts” demand a more extensive tool than what Powerpoint and the likes provide, I wouldn’t trust them with conceptual modeling. Drawing points and connecting lines is not the problem. It is coming up with relevant ideas what they conceptually stand for, that we should be concerned with. There is no substitute for first of all properly understanding what information is about, really.

You cannot do without such modelers, but then the fewer the better. Look at the mileage from the actor concept. Try it out!

You cannot arrive at a conceptual model for integrated order through addition of separately drawn-up models. Integrity of design requires focused design. Although with another object of design in mind, that is, an operating system, Frederick P. Brooks emphasizes the need for such integrity in his book **The Mythical Man-Month**. Especially for design, more people means less coherence. Quality suffers. Who cares for a drawing tool when conceptual abstraction is concerned? A conceptual modeler should really have something else on her or his mind. What you are actually saying is that conceptual modeling is “out of the running” because nobody understands what it involves. Now, regretfully, that I cannot change. I have tried, but no. Please also reread my earlier message.

Thank you for suggesting using Metapattern for “facilitating the transformation of unstructured data to structured data.” Actually, the idea of so-called contextual differentiation is to make unambiguous structuring of data/information practically possible regardless of scale. So, by using properly Metapattern-modeled information resources, there are no unstructured data left to transform. :-)

Of course, there will always be unstructured data. What you could do with Metapattern is to model a target structure. This brings us back to the need for modeling at the scale of an integrated order. For it can only be that very model providing the structure to aim for when trying to transform unstructured data. As you must have gathered, I am inclined to think about so-called artificial intelligence, machine learning et cetera more in terms of risks than benefits. Mostly it is hyperbole, anyway. What can possibly be achieved responsibly without making explicit what results may mean to stakeholders, that is, through real-life concepts touching them?

When it is “quick results” that people want, I repeat as my conviction that they are well advised not to continue with what has proven to fail at what is already for some time now really a qualitatively different scale of instantaneous interconnection between individuals. When decision makers don’t adjust, I find they limit themselves ... and the rest of us. And it seems we can also not rely on so-called it-professionals to initiate the long overdue paradigm shift. Indeed, the surest way of never achieving something useful, is to stick to an incommensurable paradigm. Yes, I am all in favor of trial and error when learning is in order. However, I find it unethical to participate in what to me is already abundantly certain to fail. You can count on me to try to make that clear, and to suggest an approach that actually will work.

I am also aware that many people don’t appreciate such a principled attitude, insisting on obedience instead. It is not at all that they know any better, on the contrary, but they insist on organizational hierarchy for authority. Well, that is why I don’t work for a “director.” I am afraid many, if nowadays not most, so-called decision makers don’t easily take and follow advice calling for a change of course (let alone paradigm). Usually a personally felt disaster has to strike for them to support change (and until they withhold from panic, all we can do is wait, being around and hoping that someone equally lacking in insight as they do doesn’t gain control).

You might be able to start change ‘under the radar,’ i.e., without formal permission. When a civil servant working for the Ministry of Foreign Affairs, now a long time ago, that’s how I helped to get all major changes off, that is, bottom up. Again, at the moment I find an information roundabout ideally suited. Get results at hardly any costs and give the “director” all the credits. S/he then might get a taste for more. Anyway, there is definitely no quicker way to get proper results than by coming to grips with real information requirements, and everything falling short of the scale of integrated order can only fail, slowly and expensively so. It is not what motivates me.

I can only point out the need for change as I see it. Well, I can do a little bit more. And I believe I am certainly doing so by arguing how the conceptual model I’ve made available to you, and here again tried to explain at considerable length, helps setting the course for coordinating information resources in practice (!) at the scale of integrated order. I hope you recognize, and from a common sense point of view I would say that your “management” should be extremely eager to support you, that you may be combining crisis management, successful at last, with laying the solid

foundation for a properly scaled integrated informational order. When on the contrary you don't find my case for such synthesis credible, and I would very much regret such an outcome, I agree with you that "we had best not pursue it" (which shouldn't stop us from exchanging messages at the personal level).

**71.41**

What would we do without time zones? Next step? Since you ask, :- ) and thank you for putting trust in my judgment, in my last message I hinted at "many additional possibilities [from] recursively structuring the concept of event (case), too."

The idea with actor as an articulated concept is to arrive at a single point of departure for recognizing participation in whatever (!) may be considered as an event. For we can now just assume an appropriate instance of actor being, say, prepared and take 'the rest' from there.

The other way around, that is, starting from the concept of event, such open-ended coverage has not yet been given due attention, however. There must be all sorts of reasons to distinguish between events, to differentiate what thus becomes an encompassing event into partial events et cetera, to relate events at any level of differentiation, and so on. I can image that, also for medical health care, reasons may vary from optimizing the quality of actual health care – which I hope receives priority – to supporting financial arrangements such as insurance to facilitating audit trails throughout. As with actor, a preset hierarchy with its fixed levels is certainly not viable for event.

From the perspective of participant as the pivotal concept, I propose to shift attention from modeling complexity underlying actor to complexity underlying event. At least for now, the actor concept seems adequately articulated. What about event? Similar advantages should be attainable from conceiving of a singular point of departure 'on the other side' of participant, too.

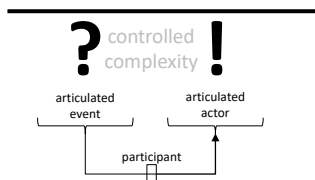


figure 1

I pause to applaud your interested patience. So-called data specialists et cetera must be under the impression that I am taking a terrible detour at best. More likely, they dismiss such efforts as idiocy. They have no time for it, for they have real work to do. Do they? I can only emphasize that an ounce of conceptual design is worth countless pounds of digital development.

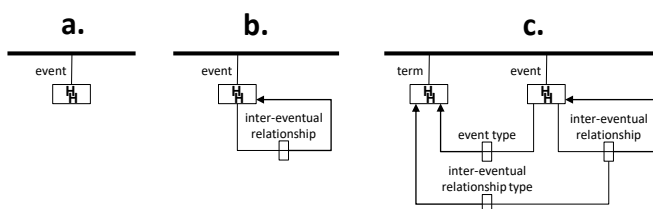


figure 2

In order to arrive at – a model of – a properly articulated concept of event, I find it helps to think beyond the scope of what seems immediately relevant. You might even consider a software development project. :-)

A simple structure is of course what I call a homogeneous hierarchy. From a set of, say, structural elements, a set of hierarchies can be composed. Including both such ingredients and results I am using a short-hand symbol, see figure 2.a.

Additional structural possibilities exist when any element-in-hierarchy may be considered related to any other element-in-hierarchy, see figure 2.b. And more, say, complexity control is possible when any element-in-hierarchy and/or any – what I've called, and thank you again for bearing with me – inter-eventual relationship may be classified, see figure 2.c.

Please note that at this stage I am merely demonstrating some rather basic possibilities for articulating event. As always, the key question concerns the variety in/of real life. So in your opinion, what is all relevant to encompass under the concept of event (case)?

And to suggest more ideas to you, and not so basic, because we seem rather satisfied with how the articulation of actor is working out, why not transpose its structure to event? See figure 3.

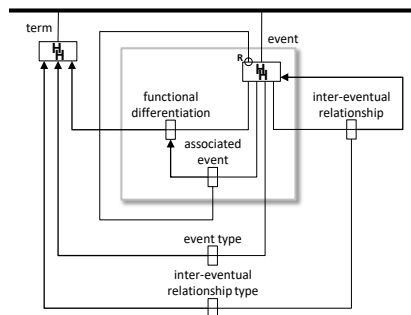


figure 3

I know you are very busy from responsibilities as your “management” sees them. :-) Therefore I don't expect you to turn to seriously consider event's eventualities soon. When you agree that I've proposed a productive “next step,” please take your time! For when you feel confident enough about both actor and event, you'll come to appreciate that most of integrated order's relevant complexity has at least conceptually been brought under control.