The Cologne Information Model: Representing Information Persistently [2009]

Thaller, Manfred

Veröffentlichungsversion / Published Version
Zeitschriftenartikel / journal article

Historical Social Research
Historische Sozialforschung

Manfred Thaller:

The Cologne Information Model: Representing Information Persistently [2009]

doi: 10.12759/hsr.suppl.29.2017.344-356

Published in:

Historical Social Research Supplement 29 (2017)

Cite as:

Manfred Thaller. 2017. The Cologne Information Model: Representing Information Persistently [2009]. *Historical Social Research* Supplement 29: 344-356. doi: 10.12759/hsr.suppl.29.2017.344-356.

Historical Social Research
Historische Sozialforschung

Other articles published in this Supplement:

The Cologne Information Model: Representing Information Persistently [2009]

Manfred Thaller[*]

Abstract: »*Das Kölner Informationsmodell: Informationen dauerhaft repräsentieren*«. If we want to decide, whether two digital files contain exactly the same amount of information, or of how precisely the amount of information they contain differs, we need an abstract model of the information, unto which the instances represented by the content of two different files can be projected. A meta model for this purpose is presented. It differentiates between the byte values representing the payload in a file and the significant properties of that payload represented by meta information in the file. That model is embedded into a broader discussion of the best way to understand the nature of information as it influences the processing of the representations derived from the data in question. Links to the software solution implemented in the Planets project for the processing of data following the model proposed are provided.

Keywords: Long term preservation, information theory, significant properties of digital objects.

1. Introduction

The long-term discussions, both within the Planets consortium as well as in general, quote as a truism that "preserving the byte stream is not enough". While this is undoubtedly true, what information systems ultimately store are byte streams and byte streams only, even if some of them are used to describe other byte streams. If we take the preservation of digital data serious, in such a form that they can be handled on a large scale by automatic processes, we need an abstraction which allows the handling of digital content in such a form that distinct units of digital data represent meaningful units of information which are functionally complete. On a pragmatic level of software engineering, the XCDL tries to accomplish this by providing for a language which allows to represent the complete content of arbitrary information objects – or at least shows a medium in which to express such representations.

[*] Reprint of: Manfred Thaller. 2009. The Cologne Information Model: Representing Information Persistently. In *The eXtensible Characterisation Languages – XCL*, ed. Manfred Thaller, 223-39, Hamburg.

This paper introduces, on a more abstract level, the conceptual model which underlies the XCDL. As this model reflects a specific concept of the nature of information and its representation on computers, it is introduced in a general way, independent of the specific targets set to the work packages within which it has been developed within Planets. As it is introduced within Planets for specific engineering purposes, however, the relationship between model and implementation is explained in section 7 below.

2. Preserving Information: An Engineering Dilemma

Already in the second paragraph of his path breaking paper on *A Mathematical Theory of Communication*, Shannon (1948) notes that "semantic aspects of communication are irrelevant to the engineering problem". This focus on a set of problems, for which practical solutions could immediately be derived from the argument presented, is probably one of the main reasons for the enormous influence Shannon's paper has had. On the other hand, one could say that what is discussed as information by him is misnamed: Within the widely accepted hierarchy of data, information, knowledge and wisdom, which is at the heart of most of today's explorations of the relationship between symbols and their intelligent processing, Shannon is clearly operating on the level of data, not on that of information.

That there is a difference between information and the symbols representing it is scarcely new: Indeed it is the starting point for most types of linguistic thinking. That this difference exists is sometimes tacitly, occasionally quite openly, acknowledged by most introductory texts into computer science, especially the ones with an engineering flavour. Indeed, among the various definitions of computer science, the assumption that it is the discipline dealing with ways to represent information in such a way that these representations can be processed is probably the most widely used one. This has the great virtue that we can process the data representing the information without solving the much more elusive problem of what information actually is. That this is unsatisfactory – as humans ultimately need information, not data – is obvious from the trend of the last decades to talk more and more of information technology rather than data processing. While this trend is obvious, it seems, however, not to be accompanied by consistent progress in the understanding of information itself.

For long-term preservation, this is extremely unfortunate: While for signals, which are transmitted between contemporary individuals, it can be assumed that whatsoever interpretative context is needed to make the signals meaningful is available for the sender as well as the recipient, the preservation of data which cannot be correctly translated into information would scarcely be helpful – Shannon does not deal explicitly with signal context, however.

Undoubtedly there exist models of information which are much further removed from the level of data; unfortunately, the way from these models to a piece of working software engineering is much less obvious than that between Shannon's model and signal processing gear. A very interesting example is Langefors' "infological equation" (1995):

(1) $I = i(D, S, t)$

Information (I) is understood here as the result of a process of interpretation (i) that is applied to data (D), applying previous knowledge (S) within the time available (t). The great attraction of this model is that – unlike Shannon's – it explicitly promises to model the meaning of messages, which are explicitly excluded from consideration by Shannon. To emphasize the difference between the models, we could say that Shannon assumes information to exist statically, therefore it can be broken into discrete units, independent of any process dealing with it, while Langefors understands information to be the result of a dynamic process, which, having a relationship to time, goes through different stages: So the amount of information existing at t_n is not – or not necessarily – equal to the amount of information at t_{n-1}, the ongoing process i having had the chance to produce more of it in the meantime.

While this equation is intellectually most attractive, it does not easily translate into a software design, however. In my opinion, that is due to the fact that the purpose of the model is to explain how information is derived from data; in doing so, it employs knowledge, however. Knowledge, unfortunately, as we have noted at the beginning, is two levels of abstraction further from data than information. So it is hard to see how it would not have to pre-exist, just as in the case of Shannon (where it does not create an engineering problem, as it is ignored, while it will create a problem with a system to implement Langefors' model, as it is an explicit functional component there).

Or, in a preservation context: To engineer a system for the preservation of information based on Shannon is impossible, as he ignores an important dimension; to engineer such a system for the preservation of information based on Langefors is impossible, as it requires the preservation of an abstraction of a higher order ("knowledge") than the abstraction ("information") for the preservation of which we strive.

3. Observations on the Conceptual Relationship between *Data* and *Information*

When we look for examples to describe the difference between *data* and *information*, we can choose from many.

We can, e.g., claim that a byte containing the string of binary digits "01100001" represents *data*. These data become *information* within a software system by the definition in the source code whether this byte represents an integer (mapping the *data* "01100001" into the *information* "number 97") or an ASCII character (mapping the *data* "01100001" into the *information* "character 'a' "). (Example 1)

We can, however, also use the example of the number 97 to describe the difference between the *data* item "97" and the possibility to derive the items of *information* "temperature of 97 degrees Celsius", "temperature of 97 degrees Fahrenheit" or "speed of 97 miles / hour" from it. (Example 2)

This example could be extended: We could easily claim that a "temperature of 97 degrees of Fahrenheit" is a *data* item, which turns into *information* only when it

is interpreted as either "today's noon temperature at Tromsø" or as "today's noon temperature at Cairo", leading to two completely different interpretations on the knowledge level when that item of information is contextualized within the two different climatic knowledge bases. (Example 3)

This example could also be contextualized in the opposite direction, however: As we know from discussions of optical storage hardware, as well as from Florida presidential ballots, the *data* item "impression of 55% of possible depth into the substrate of the medium", can be interpreted as the *information* item "0" just as well as "1", depending on the parameters of a drive's reading head or the political mindset of an election official, respectively. (Example 4)

This, in our opinion, indicates that there is in reality not a juxtaposition between two clearly separated concepts "data" and "information", but a continuum of representations between these two terms, each describing an *idealtypische* construction, which, as is the nature of *Idealtypen*, cannot be observed unadulterated in real life.

When we look at the four introductory examples again, we furthermore notice that each of them describes four interpretative processes, reminding us of Langefors. Using the numbers assigned to the examples above as coordinates on the continuum between data and information just proposed, we could claim that the relationship between, e.g., the example representing bits as data and typed programming language variables as *information* (example 1) and the one representing typed programming language variables as data and variables of a statistical system as *information* (example 2) is expressed by:

$$(2) \qquad I_2 = i\,(I_1, S_2, t)$$

For the point x on the continuum between "data only" and "perfect information", this can obviously be generalized to

$$(3) \qquad I_x = i\,(I_{x-1}, S_x, t)$$

which unfortunately still leaves us with the fact that we use S, being expressed in "knowledge", requiring that information has been derived from data and converted into that higher order abstraction already before the process described by this formalism started.

The difference between *information* and *knowledg*e is open to discussion just as the difference between *data* and *information* is. One definition, which will presumably be accepted by many as explaining the phenomenon partially, even if they do not consider it complete, defines *knowledge* as the set of conditions which allow an actor to act adequately if confronted with a piece of *information*. To generalize the relationship between all imaginable types of information, and all the possible sets of conditions necessary to act upon them adequately, would clearly be a daunting task if we, e.g., consider the sets of conditions necessary to decide upon the adequate actions when confronted with possible interpretations of the body language of a participant in a social interaction.

Fortunately, within digital preservation, the scope of knowledge is much more restricted: Identifying the irony contained in a digital document and acting correctly upon it is clearly beyond the goals of digital preservation. Displaying the stored data in such a way that that irony, as transmitted by the text or its visual attributes,

could be identified and acted upon by a human reader who has the necessary knowledge is clearly within these goals. This example has been chosen intentionally: Quite obviously, there is much irony contained in texts that have been transmitted to us in traditional paper-bound documents which totally evades us, as we do not have the knowledge necessary to identify it. And – at least within the scope of this paper – we assume that digital preservation requires preserving all aspects that could have been preserved within the pre-digital tradition of information. Making digital information as robust as non-digital information is a clearly circumscribed and solvable task for the near future. Trying to solve all sorts of conceptual problems for the preservation of digital information, which have never been solved for pre-digital information, is fascinating: It may, however, blur the solvable core of the task so much that it prevents perfectly feasible solutions which are needed now.

If we accept that digital information has ultimately to be preserved as "byte stream" – data –, we are either assuming that the knowledge needed to re-generate the information from the data is preserved in some other way, or we accept that, at least for the reduced scope defined above, we have to be able to reduce that knowledge to information, which in turn is stored as data alongside the data representing the information we want to preserve in the first place. Let us for these purposes assume that knowledge can be generated from information in a similar way as information can be generated from data, say:

$$(4) \qquad S_x = s\,(I_{x-1},\, t)$$

Let us also generalize our notion that information is derived from data: We maintain that an "item", which is closer to data than the resulting one, is transferred to another "item", which is closer to information than the original one. We also maintain that this process may be repeated, resulting in a third "item" which is still closer to pure information than the immediate one. We do not claim, however, that there are a clear discrete number of stages along the scale used: So two stages in this repeatable process of deriving information from data can only be described by an ordinal scale. We know that each "item" during a repetitive application of the infological equation will be further from the data level and closer to the information level than the preceding one. There is no possibility to measure the distance between two such steps. To indicate this we drop the numeric subscript from (2) and replace it by a generic one. Furthermore, we indicate that the item of information we are starting with and the previous knowledge we use need not be related to the new level of interpretation by the same distance. Then we get:

$$(5) \qquad I_x = i\,(I_{x-\alpha},\, S_{x-\beta},\, t)$$

If we now insert (4) into (5) we get:

$$(6) \qquad I_x = i\,(I_{x-\alpha},\, s(I_{x-\beta},\, t),\, t)$$

If we accept the reasoning presented, we have reached a stage where – on the purely conceptual level – we have reduced the problem of preserving information to a problem of conserving *data*, within the context of two classes of data-driven processes (i(...) and s(...)). Or, in the most general form, to the statement that infor-

mation develops out of data, by relating data to each other in such a way that the context data items mutually provide to each other generates information items.

This, furthermore, assumes that there are some not-yet-named items which at some stage hold data – like the string "Planets" – and at a later stage, when they are placed into a sufficiently rich context by other data, hold information – such as the *name of a project*.

As long as we restrict ourselves to the restricted scope of preservation as defined above, s(...), by the way, can be preserved relatively easily. If we take the example of the two numbers 300 and 400 within an image, these two numbers clearly represent *data*: Knowing that they stand for width and height turns them into *information*; the process needed to map the byte stream representing the pixels described by them into the rendering of an image is the *knowledge* needed to extract the image.

The function s(...) could therefore simply be preliminarily described as a set of rules needed to construct some basic types of information objects, such as images, out of the data that represent them. This task would be trivial if there existed only one class of such information objects in the world; it is unsolvable if that number of classes of information objects – not of information objects themselves! – grows towards infinity.

4. Consequences for an Engineering Context: An Intuitive Introduction – Texts

The problem with our concepts so far is that, while they may be convincing by themselves, many such attempts at formalization have simply no relationship to any working solution. The approach described above, however, lends itself very easily and immediately to an implementation for any specific class of information objects. If this is true and we find ourselves able to show that the number of necessary classes of higher order information objects to be described by s(...) is reasonably small, we would be much closer to a solution of the preservation challenge.

Let us relate the textual example given above – the string "Planets" and its relationship to the concept *name of a project* – to an introductory example from the world of markup languages. In the text fragment

<person><firstName>John</firstName><surname>Biggin</surname></person>

we can describe "John" and "Smith" as two tokens, which *carry* information, though they are data. Or rather more, we call them two series of discrete tokens {'J', 'o', 'h', 'n'} and {'B'', 'i', 'g', 'g', 'i', 'n'}. For all representations of information, there is one level at which these tokens become atomic: For images, e.g., at the level of pixel values, for trivial texts, at the level of individual characters. The "static" representation of the data is the string "Biggin"; that it constitutes a "surname" is derived from additional data providing a context.

To explore some of the consequences of this model, let there be four chunks of information:

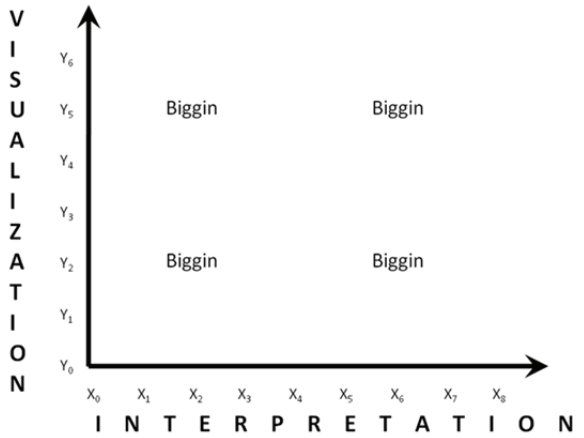
1) <person><surname><bold>Biggin</bold></surname></person>

2) <person><surname><italics>Biggin</italics></surname></person>
3) <airfield><name><bold>Biggin</bold></name></airfield>
4) <airfield><name><italics>Biggin</italics></name></airfield>

Which of the chunks are more similar to each other: (1) and (2) or (1) and (3)?

There is obviously no intersubjective answer to this question, as in one case we talk about a similarity on the interpretative level, on the other hand about one on the visualization level. Note that we have not claimed that one of these possible interpretations might hold more or less *information* than the other. Rather we propose to interpret this as a case where the two interpretations are promoting the data contained in the basic tokens towards information, do so in two independent directions, however, as in figure 1 below.

Figure 1: Basic Coordinate Concept



We have used here the concept of two orthogonal dimensions to interpret the relationship between two different ways of applying context to data items: Please note that the coordinates in these two conceptual dimensions, as we call them, are not measured in rational numbers: The statement that the chain (1) of content-carrying items above is located at (x_2, y_5) and that (3) is located at (x_6, y_5) is valid as soon as we accept that y_5 stands for "bold", x_2 stands for "surname of a person" and x_6 for "name of an airfield". Whether, e.g., a graph describing the semantic space of names exists, allowing a statement on the relative semantic proximity of x_2 and x_6, may influence the engineering usefulness of such a representation. It is not immediately necessary for its application, however.

This approach at modelling the context, which feeds Langefors' process i(...) creating information out of data, can be described as a conceptual space of *n* orthogonal conceptual dimensions in which the content-carrying atomic tokens are presented.

We have already emphasized that a conceptual dimension can be connected to a metric which allows algebraic operations – e.g. the conceptual dimension "font

size"", but in our model it in no way has to be so. We have said above in (6) that, for applying it to preservation purposes, we require the infological equation from which we started to be converted into a fully recursive model, where Langefors'' "previous knowledge" can be preserved by the same mechanism which we use for the preservation of the data which are converted, with the help of that knowledge, into information. If we say that conceptual information can be described by dimensions, which, as long as they remain orthogonal, can be measured by *any* kind of information, we imply that a chain of content-carrying atomic tokens usually described as a text can be localized within a conceptual dimension by another text (the information contained in which can be traced back to another chain of content-carrying atomic tokens).

A simple example of this is provided by a text like:

> CAESAR
>
> 1.2.191 Let me have men about me that are fat;
>
> 1.2.192 Sleek-headed men and such as sleep o' nights:
>
> 1.2.193 Yond Cassius has a lean and hungry look;
>
> 1.2.195 He thinks too much: such men are dangerous.

In this case { Let me have men about me that are fat; Sleek-headed men and such as sleep o' nights: Yond Cassius has a lean and hungry look; He thinks too much: such men are dangerous. } can be described as having the conceptual dimension "author = Caesar", while the line { Let me have men about me that are fat; } or more precisely all characters constituting the tokens it is made of, have, independent of this, also the conceptual dimension "located on line 191 of scene 2 of act 1".

In an even more extreme case:

> To be or not to be, that is the question*
>
> *Hamlet, Act 3, Scene 1

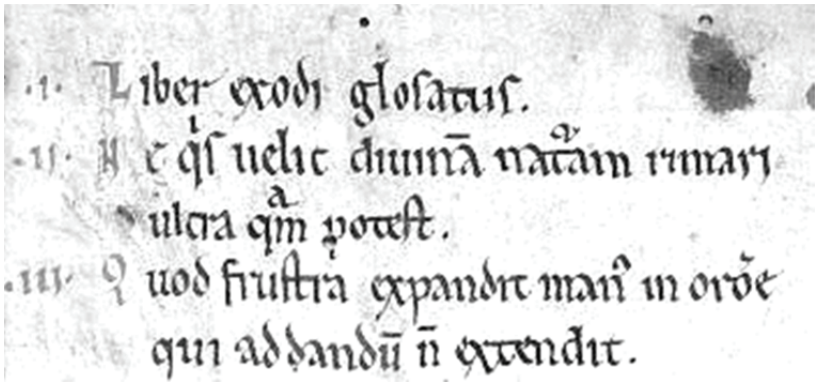
can be described as containing one content-carrying atomic token – '*' – which is measured on a dimension "background explanation" by the series of content-carrying atomic tokens "Hamlet, Act 3, Scene 1".

5. Consequences for an Engineering Context: An Intuitive Introduction – Non-texts

While we have chosen our examples from the world of texts so far, there is no reason why the previous assumptions should not be extended to cover other types of data. In the example given below, there is clearly a set of pixels which can be connected not only to the properties all of the pixels in the image have in common – being meaningfully interpreted as an image of x times y pixels, having a certain colour depth – but also to some which are specific for this set of pixels: e.g. representing the text "Liber exodi glosatus". That a conceptual dimension can have an image as value, we will not discuss in detail, as we assume it to be obvious.

There is a very fundamental difference between images and texts, however: content-carrying atomic tokens of texts occur in series which have exactly one dimension. A text is a series of characters following each other. That a text may be non-linear does not change this basic property. If at some stage in a text two branches split – be it the two parallel sections in typesetting, e.g., a text accompanied by emphasized keywords at the sides of the main page, be it the texts of different origins bound together to a composite document, as e.g. in critical editing – each of these two branches is nevertheless exactly one series of content-carrying atomic tokens (characters). So to connect a conceptual dimension to a section of the content-carrying atomic tokens, two numbers suffice: One defining with which content-carrying atomic token in the linear sequence the applicability of a conceptual dimension starts, and one which defines where it ends.

Figure 2: Image with Text



In contrast, within an image interpretative information expressed in arbitrarily many conceptual dimensions will always be connected to a two-dimensional area – either to the image as a whole or to a section of it, as in the example above where a transcribed string was connected to a subsection of an image.

There is another difference between images and texts, if understood as represented here. The content-carrying atomic tokens in an image – usually called pixels – are in recent images values which, by a direct process connected to some measuring device, usually express a degree of intensity. As measurements, they can be used within arithmetical expressions and comparisons. While in reality it may be a bit more complicated, at least in many images we can say that a pixel value of 100 is exactly twice as intensive as a pixel value of 50, and the pixel value 110 is 10% more intense than the pixel value 100. In terms of the metrics as used within statistical computations, we can therefore say that the content-carrying atomic tokens can be measured on rational scales.

With texts, on the other hand, such a relationship does not exist. A character with a pixel value 110 (‘n’) is not 10% more of anything than 100 (‘d’). Nevertheless, a meaningful relationship could be expressed between these two values, as 110

is later in the collating sequence. There is a relationship between the two values, therefore – usually measured on ordinal scales.

This type of applicable metrics is independent of the basic types from which we have chosen our examples: There are – or have been historically – images where the pixel values are not expressed on a rational scale, but where they represent look-up indices of a colour table, not even being rational but nominal, if we follow the distinction of applicable scales for measurement.

On a purely intuitive level we can therefore conclude that images and texts can both be described by the concept of content-carrying atomic tokens, where (a) the assembly of tokens follows a certain basic logic – one-dimensional sequence, two-dimensional matrix – and (b) there exists a specific system of measurement or system of metrics to express the relationships between content-carrying atomic tokens.

For both types of higher order information objects we can also say that they can be described by each of the content-carrying atomic tokens – or more intuitively: any meaningful complex of such tokens – can be measured by their position on a set of conceptual dimensions.

And, as the positions on these conceptual dimensions can be denoted by a very wide variety of items of information, the relative position of two content-carrying atomic tokens, or complexes of such tokens, on any conceptual dimension can be measured by completely different metrics. As the dimensions are orthogonal, it is furthermore not immediately necessary to try to map these various metrics upon a common one, as long as we measure distances only within individual dimensions.

6. Consequences for an Engineering Context: An Attempt at Formalization

To make the generalization easier, we will redraw our representation first by indicating the positions of individual token by the vectors describing their position on the interpretative dimension. To explain about the letter ‘g’ in the set of content-carrying atomic tokens being at the positions “bold” and “surname” on the dimensions “visualization” and “interpretation”, we therefore draw figure 3 below.

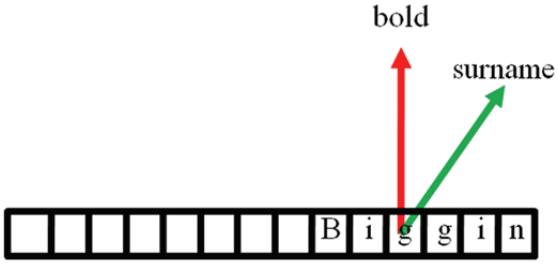
In the least general way, we can describe this representation of a text as the string of characters “Biggin” interpreted within the conceptual space defined by the dimensions visualization { bold, italic } and interpretation {surname, topographical name}.

Slightly more general, we can rephrase this by saying that any text like this can be described as a set of content-carrying atomic tokens within the conceptual space defined by two conceptual dimensions, or, more formally:

$$(7) \qquad \langle text \rangle ::= \{ T, C_1, C_2 \}$$

or, more generally:

Figure 3: Basic Example in Vector Notation



(8) <text> ::= { T, C_1, C_2, ..., C_n }

which we will write henceforth as

(9) <text> ::= { T, C_n }

to express texts as "sequences of content-carrying atomic tokens, each of which has a well-defined position within an n-dimensional conceptual universe".

Using the same formalism to describe our considerations of an image discussed above, we can write:

(10) <image> ::= { T_x x T_y, C_n }

to express images as "planes of content-carrying atomic tokens, each of which has a well-defined position within an *n*-dimensional conceptual universe".

From this we easily arrive at the generalization

(11) <I> ::= { T_1 x T_2 x ... x T_m, C_n }

or

(12) <I> ::= { T_m, C_n }

to express arbitrary information objects as *m*-dimensional arrangements of content-carrying atomic tokens, each of which has a well-defined position within an *n*-dimensional conceptual universe.

Examples for 3- or 4-dimensional information objects are easy to find: A video can generally be represented as a series of two dimensional objects; a 3-D simulation as a series of 3-dimensional snapshots.

In both cases, the arrangements of tokens and the conceptual universes, we understand a "dimension", be it one of token space or of conceptual space, as a definition of the metrics needed to express meaningful measurements within this dimension.

7. Relationship of the Abstract Model to the XCL

The abstract model described above may seem to be far from the eXtensible Characterisation Languages. What it describes, however, is the clean separation between the normData – content-carrying atomic tokens, which are mapped from their origi-

nal representation unto dimensions with a well-defined metric – and the properties describing arbitrary segments of them. normData mapping happens, e.g., when image information is converted from the object colour space used within an image file to the standard RGB colour space underlying the image definition of the XCDL. Only by mapping normData into such standard token spaces do they become comparable.

Connecting this construction back to the theoretical considerations we started from, we propose to understand an XCEL – which is a formal specification, i.e., a document describing the way in which information contained in a digital object can be stored as data in a file format – as an implementation of Langefors' i(...), the process generating information from data. The knowledge about the behaviour of one of the information objects defined within the preceding section – text, image – is an implementation of the s(...) which we proposed analogously to i(...) in formalism (4) of section 3 above.

8. Relationship to Layout and Semantics

It has been shown (in an unpublished internal deliverable of the Planets project) that the inclusion of the requirement to preserve layout characteristics of a document, beyond preserving the information contained within the actual documents, leads to conflicting recommendations. While the capability to distinguish between a text contained in the main section of a document vs. the same text being part of a footnote is preserved better for a *human* reader of the document by one decision, the same capability may be preserved better for *handling by software applications* by a different, indeed contradictory, decision.

On the purely conceptual level this can be described as a case where knowledge about a layout principle pre-exists within the human reader, say "If at the bottom of the page a text in a smaller type font exists, where individual paragraphs of such texts are preceded by raised numerals, such a block of text represents a footnote to that portion of the main text, where the same numeral appears in a smaller, raised font within the main text". For the human reader the rendering of that relationship by a word processing system – as e.g. an Office application – and a page descriptive one, – like e.g. PDF – cannot be distinguished, as (s)he derives the relationship in both cases from the kind of knowledge about layout described before.

For a technical system interpreting the data, the relationship is expressed explicitly in virtually all word processing formats; it can, therefore, easily be extracted and expressed within an XCDL representation of an office document. In the case of a page description language like PDF, however, the file contains only layout information and does not make the relationship between two blocks of text explicit.

Obviously it would be possible to implement at least some types of pre-existing knowledge about layout in such a way that the implicit relationship is extracted from a generalized representation of the textual object. A number of ways to achieve this would be possible. One possible procedure could be as follows:

1) Extract from the file to be processed a new type of textual object, where the content-carrying atomic tokens form a plane. Each token is described by arbi-

trarily many conceptual dimensions, two of which represent the position of the token within the plane.

2) Apply an additional algorithm unto this preliminary textual object, mapping properties so far expressed by the positions in the textual plane into conceptual dimensions.
3) Transform this textual object, where the content-carrying tokens are arranged in a plane, into the standard textual object where the content-carrying tokens form a linear series.

This procedure would be completely compatible with the conceptual model chosen: As $S_x = s\ (I_{x-1}, t)$ depends on time and describes the transformation between information and knowledge as gradual, we have here an instance of a procedure where the data is simply further advanced on the gradual conversion from data into knowledge after the application of the second transformation.

We would like to emphasize again that the goal of the XCL is not to solve problems of preservation considered unsolvable in the pre-digital handling of information, but to bring digital preservation unto the same level as pre-digital preservation. Nevertheless, the example given above for the integration of knowledge derived from an interpretation of layout into the basic XCL extraction logic encourages us to assume that the strategy described here should also be explored to implement preservation strategies for other types of content, notably semantic ones, which are also derived from an interpretation of lower levels of extracting information from the stored data.