Psychol Res (1994) 56: 224–236

Psychological Research
Psychologische Forschung

The dynamics of Prägnanz

Peter A. van der Helm

Nijmegen Institute for Cognition and Information (NICI), University of Nijmegen, P.O. Box 9104, 6500 HE Nijmegen, The Netherlands; E-mail: helm@nici.kun.nl

Received April 13, 1993/Accepted December 16, 1993

Summary. There is quite wide-spread agreement about the relevance of pattern Prägnanz (Koffka, 1935) with respect to the human interpretation of visual patterns. There is less agreement about whether pattern Prägnanz is based solely on pattern information (static) or also on the history of the perceiver (dynamic). In Van Leeuwen and Van den Hof (1991), experimental data concerning serial patterns are presented within the framework of the dynamic-network approach initiated by Buffart (1986, 1987). These experimental data are claimed to give evidence against the static-coding approach initiated by Leeuwenberg (1969, 1971). In the present paper, however, I show first that Buffart's theoretical basis is incorrect, and that in fact Leeuwenberg's static-coding approach is the basis for the dynamic-network approach. Second, I show that those experimental data rather give evidence in favor of the static-coding approach, by using those same data for a test of the most recent static-coding model (Van der Helm & Leeuwenberg, 1991; Van der Helm, Van Lier, & Leeuwenberg, 1992). Finally, I propose a reconciliation between the two approaches, in the sense that the dynamic-network model could be shaped in such a way that it yields a simulation, and maybe even an enrichment, of the static-coding model.

Introduction

In this paper, I shall try to shed some light on the question as to whether the human interpretation of visual patterns is based either on just an account of pattern information (the *static* approach) or also on an account of the history of the perceiver (the *dynamic* approach). The primary impetus for writing this paper came from Van Leeuwen and Van den Hof (1991) which concerns the concept of Prägnanz (Koffka, 1935). Prägnanz is a notion that applies to the *goodness* of patterns or, rather, of pattern interpretations. A measure of Prägnanz is considered to be valuable in explaining or predicting the human interpretation of patterns. For instance, to explain why, in Figure 1 A, generally interpretation I1 is preferred and, in Figure 1 B, generally segmentation S1. For a long time, however, a continuing discussion has concerned the choice of an appropriate measure of Prägnanz (cf. Simon, 1972; Hatfield & Epstein, 1985). Van Leeuwen and Van den Hof (1991) is also concerned with this problem.

Van Leeuwen and Van den Hof (1991) is one of the latest in a series of publications concerning a specific dynamic-network approach to Prägnanz, initiated by Buffart (1986, 1987). That approach started as a branch of Structural Information Theory (SIT), a static-coding approach to Prägnanz, initiated by Leeuwenberg (1969, 1971). The present paper focusses on the competition between these two approaches.

In SIT's static-coding approach, a restricted set of coding rules is employed to encode a given pattern, yielding codes that are each assumed to represent a perceptually

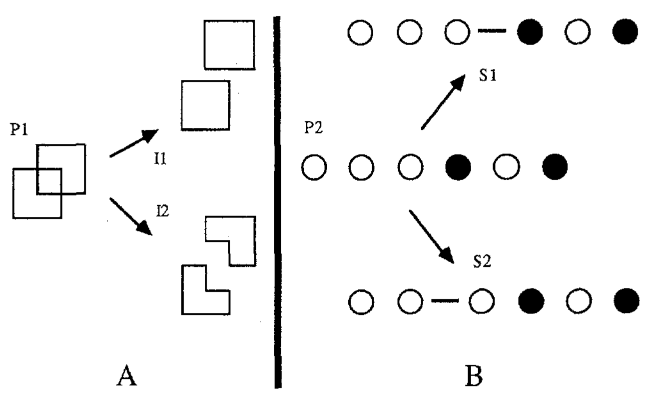

Fig. 1. In visual-shape perception, a major problem is how to predict the preferred interpretation of a pattern. In principle, a pattern can be interpreted in many ways. In Figure 1 A, two interpretations, I1 and I2, of linedrawing P1 are visualized: generally, interpretation I1 is preferred. In Figure 1 B, two segmentations, S1 and S2, of the patterned sequence P2 of black-and-white dots are given: generally, segmentation S1 is preferred

possible interpretation of that pattern. Then, in line with the minimum principle (Hochberg & McAlister, 1953), the simplest code (using some complexity metric) is assumed to represent the preferred interpretation of that pattern and therefore determines the Prägnanz of that pattern. Here simplicity is closely related to regularity. For instance, in Figure 1 A, interpretation I1 is simpler than interpretation I2, since a square is more regular than an L shape. Empirical research within this approach concerns, e. g., judged complexity (Leeuwenberg, 1969, 1971), pattern completion (Buffart, Leeuwenberg, & Restle, 1981; Van Lier, Van der Helm, & Leeuwenberg, 1993), foreground–background (Leeuwenberg & Buffart, 1984), and beauty (Boselie & Leeuwenberg, 1985). In that research, generally two-dimensional line patterns or surface patterns are considered. In the present paper, only patterned sequences of dots (as in Figure 1 B) will be considered, for two reasons. First, such patterns can be translated straightforwardly into the language used in both approaches presently under discussion and therefore enable a more direct investigation into fundamental differences between the approaches. Second, such patterns are the only ones used so far in empirical research within the dynamic-network approach.

The dynamic-network approach can be seen as a network implementation of the static-coding approach, with some differences. The main conceptual difference is that the preferred interpretation of a given pattern is not assumed to be represented by the simplest code, but by the most active code. The activity of the codes in the network is influenced not only by the given input pattern, but also by history, i. e., by earlier input patterns.

In the present paper, it is not this general concept of the dynamic-network approach that will be opposed, but its specific implementation, as given in, among others, Van Leeuwen and Van den Hof (1991). That implementation is claimed to be based on Buffart's (1987) concept of *hierarchical completeness,* but it will be shown below that this claim is incorrect. Furthermore, it will be argued that Van Leeuwen and Van den Hof's (1991) measurement of Prägnanz has to be rejected if one takes into account the most recent static-coding model, which is based on the concept of *accessibility* (Van der Helm & Leeuwenberg, 1991; Van der Helm et al., 1992). Finally, I shall sketch how this concept of accessibility may provide a good basis for a dynamic-network model as well. But first, both approaches will be introduced in more detail.

The static-coding approach

The static-coding approach of SIT (Leeuwenberg, 1969, 1971) is a coding theory according to which a pattern can be encoded by means of certain coding rules. These coding rules are applied to a symbol series representing the pattern. For instance, (the contour of) a square-like subpattern as in interpretation I1 in Figure 1 A can be represented by the symbol series *kakakaka* in which *a* represents an angle of 90° and *k* the length of each of the four line segments. Similarly, pattern P2 in Figure 1 B can be represented straightforwardly by the series *aaabab* in which *a* represents an empty circle and *b* a full circle. Now, the encoding of such a symbol series yields all possible codes of that series, each code reflecting an interpretation of the represented pattern. Then, in line with the minimum principle, the simplest code is taken to reflect the preferred interpretation of the pattern, and therefore determines the Prägnanz of the pattern. In this section, the coding rules will be discussed first, and then the measurement of code complexity.

Coding rules

In general, each coding rule describes a specific kind of regularity, on the basis of identity relationships *(identities)* between symbols in a symbol series. First, two coding rules will be discussed for which rather general agreement exists about the perceptual relevance.

First, the *Iteration rule*, or *I rule*, which can be applied to express that a (sub)series contains successive identical symbols.

Iteration rule: $kkk \ldots kk \rightarrow N*(k)$

The expression at the right-hand side is called an *I form*, in which *N* equals the number of symbols *k* in the series at the left-hand side. For instance, the series *aaaaa* can be encoded into the I form 5*(*a*).

Second, the *Symmetry rule*, or *S rule*, which can be applied to express that a (sub)series contains pairs of identical symbols, nested around a so-called *pivot*.

Symmetry rule: $k_1 k_2 \ldots k_n\ p\ k_n \ldots k_2 k_1 \rightarrow \mathrm{S}[(k_1)(k_2)\ldots(k_n),(p)]$

The expression at the right-hand side is called an *S form*, in which (*p*) is the pivot. For instance, the series *kapmpak* can be encoded into the S form S[*(k)(a)(p),(m)*].

The following examples illustrate that identities between subseries can also be expressed, inducing a so-called *chunking* in the series (1 a, 1 b), that the pivot in an S form may be empty (2), that I forms and S forms may be nested hierarchically (3 a, 3 b, 3 c), and that generally several codes are possible for a given symbol series (4 a, 4 b):

1 a.	*ababab*	→3*(*ab*)	→ chunking: (*ab*)(*ab*)(*ab*)
1 b.	*badpqvwpqbad*	→S[(*bad*)(*pq*),(*vw*)]	→ chunking: (*bad*)(*pq*)(*vw*)(*pq*)(*bad*)
2.	*abppab*	→S[(*ab*)(*p*)]	
3 a.	*bapabapa*	→2*(*bapa*)	→ 2*(*b* S[(*a*),(*p*)])
3 b.	*aabppaab*	→S[(*aab*)(*p*)]	→ S[(2*(*a*)*b*)(*p*)]
3 c.	*ababbaba*	→S[(*a*)(*b*)(*a*)(*b*)]	→ S[2*((*a*)(*b*))]
4 a.	*aaabab*	→3*(*a*) S[(*b*),(*a*)]	
4 b.	*aaabab*	→2*(*a*) 2*(*ab*)	

Such codes are assumed to reflect allowed (i. e., perceptually possible) interpretations. For instance, the code in example 4 a expresses the identity of the first three symbols in *aaabab*, plus the identity of the fourth and sixth symbols. This corresponds to precisely all identity of symbols in, e. g., the symbol series *xxxyzy*. Therefore, *xxxyzy* is called an *abstract code*, representing the interpretation reflected by the code 3*(*a*) S[(*b*),(*a*)] (Collard & Buffart, 1983). So a pattern interpretation is represented by a symbol series (an abstract code) that indicates the pattern information (identities) responsible for that interpretation. To illustrate the use of all this, remember that pattern P2 in Figure 1B could be represented by the symbol series given in example 4a. Now, segmentation S1 in Figure 1B is taken to be based on the interpretation reflected by the code in example 4a. This may be clear by the fact that the I form and the S form in example 4a correspond to respectively the first segment *aaa* and the second segment *bab* in that segmentation S1. A similar argument holds for example 4b and segmentation S2 in Figure 1B.

Beside the I and S rules, several other coding rules have been proposed in coding theories like SIT. For the present paper, three sets of coding rules are relevant. First, the so-called *ISA rules* which are the only ones presently used in SIT (see below). The ISA rules comprise, beside the I and S rules, only the so-called *Alternation rule*, or *A rule*. The A rule can be applied to express the fact that a series contains successive subseries that either all begin or all end identically. Both cases are defined as follows:

Alternation rule:
$kx_1\ kx_2\ \ldots\ kx_n \rightarrow <(k)>/<(x_1)(x_2)\ldots(x_n)>$
$x_1k\ x_2k\ \ldots\ x_nk \rightarrow <(x_1)(x_2)\ldots(x_n)>/<(k)>$

For instance, the series *arasat* can be encoded into the A form $<(a)>/<(r)(s)(t)>$, while its reversal *tasara* can be encoded into the A form $<(t)(s)(r)>/<(a)>$.

Second, the so-called *DIS rules*, which were used in SIT until several years ago, and which, in Van Leeuwen and Van den Hof (1991), are given as the coding rules used in SIT. The DIS rules comprise, beside the I and S rules, several other coding rules. The most prominent one is the so-called *Distribution rule*, or *D rule:*

Distribution rule:
$k_1x_1k_2x_2\ldots k_Nx_N \rightarrow \mathrm{D}[(k_1)(k_2)\ldots(k_n),\ (x_1)(x_2)\ldots(x_m)]$
for: N/n and N/m integers, $k_i = k_{i+n}$ $(i = 1,\ldots,N-n)$,
and $x_j = x_{j+m}$ $(j = 1,\ldots,N-m)$.

For instance, the series *arbsatbrasbt* consists of the two intertwining series *ababab* and *rstrst*, and can be encoded into the D form D[(*a*)(*b*),(*r*)(*s*)(*t*)].

Third, the so-called *DIT rules*, which, according to Buffart (1987), are derived from the concept of *hierarchical completeness* (see below), and yield precisely all interpretations represented in his network model. The DIT rules comprise, beside the D and I rules, only the so-called *T rule*:

T rule:
$k_1x_1k_2x_2\ldots k_{N-1}x_{N-1}k_N \rightarrow \mathrm{T}[(k_1)(k_2)\ldots(k_n),(x_1)(x_2)\ldots(x_m)]$
for: N/n and $(N-1)/m$ integers, $k_i = k_{i+n}$ $(i = 1,\ldots,N-n)$,
and $x_j = x_{j+m}$ $(j = 1,\ldots,N-1-m)$

The T rule may look like the D rule, but has some overlap only with the S rule. In fact, in Buffart (1987), the S rule is claimed to result, but in a personal communication Buffart confirmed that it is actually the T rule that should have been claimed to result. For instance, neither the D nor the S rule can describe all identity of symbols in the series *abaca*, but the T rule can: with the T form T[(*a*),(*b*)(*c*)]. Furthermore, for, e.g., the series *abacaba*, the identity of symbols described by the S form S[(*a*)(*b*)(*a*),(*c*)] can also be described by the nested T forms T[(*a*),(T[(*b*),(T[(*a*),(*c*)])])]. However, the nested T forms do not allow any further encoding of arguments, whereas the argument (*a*)(*b*)(*a*) of the S form can be encoded further, namely into S[((*a*)),((*b*))].

The old complexity measure

Once the coding rules have been applied to get all possible codes of a symbol series, the so-called *I load* (structural-information load) is used to quantify the complexity of each code. Then, in line with the minimum principle, the simplest code (the one with minimal I load) is taken to reflect the preferred interpretation of the pattern represented by the symbol series. Thus, the Prägnanz of the pattern is determined by that simplest code, and quantified by its I load (the lower the I load, the higher the Prägnanz).

The I load, as presented in Van Leeuwen and Van den Hof (1991), was used in SIT until several years ago and will therefore be called the *I_{old} load.* The I_{old} load was meant to reflect the amount of memory space needed to store a code. Independently of the specific set of coding rules, the I_{old} load equals the number of pattern symbols in a code plus the number of I and S forms in that code (a pattern symbol is an element of the symbol series that represents the pattern). For instance, the code 3∗(*a*) S[(*b*),(*a*)] gets an I_{old}-load value of $I_{old} = 5$, since it contains three pattern symbols, one I form, and one S form. Although empirically the I_{old} load performs reasonably well, it gives rise to skepticism, since, if I forms and S forms are counted, it depends on syntactical artifacts of the model (Hatfield & Epstein, 1985).

Accessibility and the new complexity measure

The reason why, presently, only the ISA rules are used in SIT is that the ISA rules are the only so-called *transparent holographic* coding rules. The notion of *holography* applies to the intrinsic character of the kinds of regularity that are assumed to be perceptually relevant. There are only 80 coding rules describing a holographic kind of regularity. The notion of *transparency* applies to the hierarchical character of codes, and concerns the hierarchical compatibility of kinds of regularity. Among the 80 holographic coding rules, only the ISA rules allow for a hierarchically transparent description of regularity. Together, the notions of holography and transparency constitute the concept of *accessibility*, which has been elaborated extensively in Van der Helm and Leeuwenberg (1991). Here, the notion of transparency is also necessary to understand the new complexity measure.

Transparency. Transparency implies that the hierarchical character of codes is not just a syntactical artifact of the coding model, but a psychologically meaningful aspect of the description of regularity. This may be illustrated as follows. The symbol series *ababbaba* can be encoded into the S form S[(*a*)(*b*)(*a*)(*b*)]. The argument (*a*)(*b*)(*a*)(*b*) of this S form is said to represent a higher hierarchical level, and can be encoded into the I form 2∗((*a*)(*b*)) which, nested in the S form, yields S[2∗((*a*)(*b*))]. Now, observe that any

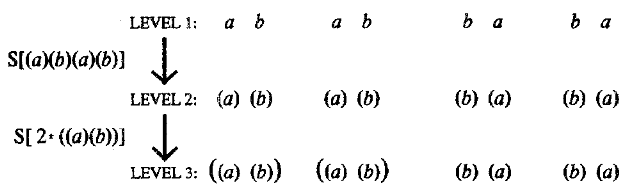

Fig. 2. The transparent Symmetry rule. The S form S[(*a*)(*b*)(*a*)(*b*)] induces a chunking in the level-1 symbol series, represented by the level-2 series. The I form 2∗((*a*)(*b*)) in the S argument corresponds unambiguously to the I form 2∗(*ab*) in the level-1 series, inducing the chunking (*ab*)(*ab*) which can be superimposed on the level-2 series, leading to the hierarchical chunking represented at level 3

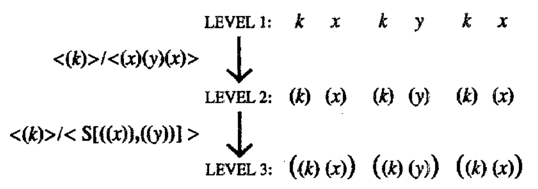

Fig. 3. The transparent Alternation rule. The A form <(*k*)>/<(*x*)(*y*)(*x*)> induces a chunking in the level-1 series, represented by the level-2 series. The S form S[((*x*)),((*y*))] in the second argument of the A form corresponds unambiguously to the S form S[(*kx*),(*ky*)] in the level-1 series, inducing the chunking (*kx*)(*ky*)(*kx*), which can be superimposed on the level-2 series, leading to the hierarchical chunking represented at level 3

kind of regularity in the argument (*a*)(*b*)(*a*)(*b*) of the S form corresponds unambiguously to the same kind of regularity in the first half *abab* of the symbol series *ababbaba*. For instance, the I form 2∗((*a*)(*b*)) in (*a*)(*b*)(*a*)(*b*) clearly corresponds to the I form 2∗(*ab*) in *abab*. So in an almost visual sense, the S form is transparent, i. e., through the S form, regularity in the symbol series can be seen. Since this holds for any S form, the S rule is called a *transparent coding rule*.

In order to understand transparency further, remember that an ISA form induces a chunking in the symbol series (see above). Such a chunking can be found by decoding an ISA form without removal of the parentheses in the ISA form. Now, according to the concept of transparency, the code S[2∗((*a*)(*b*))] in fact combines an S form S[(*a*)(*b*)(*a*)(*b*)] and an I form 2∗(*ab*). Both the S and the I forms induce a chunking in the symbol series. Only for transparent coding rules, two such chunkings are always compatible and the combination is called a *hierarchical chunking* (see Figure 2).

Two further examples may be illustrative. First, consider the encoding of *acbdad* into the D form <(*a*)(*b*)(*a*)>/<(*c*)(*d*)(*d*)>. The left-hand argument (*a*)(*b*)(*a*) can be encoded into S[((*a*)),((*b*))], and the right-hand argument (*c*)(*d*)(*d*) into (*c*)2∗((*d*)). However, the S form and the I form do not correspond to an S and an I form in *acbdad*. Therefore, the D rule is not transparent. A similar argument holds for the T rule.

Second, the A rule is transparent, as may be clear from the encoding of $kx_1kx_2...kx_n$ into the A form <(*k*)>/<$(x_1)...(x_n)$>. Any kind of regularity in the argument $(x_1)(x_2)...(x_n)$ of that A form clearly corresponds unambiguously to the same kind of regularity in the chunk series $(kx_1)(kx_2)...(kx_n)$ and therefore, also corresponds unambiguously to the same kind of regularity in the series $kx_1kx_2...kx_n$. See Figure 3 for a specific example in which also the resulting hierarchical chunking is shown.

The new complexity measure. The new complexity measure, the so-called *I_{new} load*, has been discussed extensively in Van der Helm et al. (1992), and may be introduced here as follows. First, consider the encoding of *abcabcab* into the S form S[(*ab*)(*c*),(*ab*)]. As follows from what we said above, this S form corresponds to an abstract code *xztpqtxz*, and induces a chunking into the chunk series (*ab*)(*c*)(*ab*)(*c*)(*ab*). The abstract code reflects the regularity expressed by the S form, and the chunking reflects the hierarchy induced by the S form. These two aspects can be combined by that chunking being imposed on the abstract code, yielding the *abstract chunking* (*xz*)(*t*)(*pq*)(*t*)(*xz*). Now, the I_{new} load equals the number of all different elements (symbols, and chunks containing at least two elements) over all hierarchical levels in such an abstract chunking. Thus, for the S form above, I_{new} = 7 since the abstract chunking contains seven different elements: *x*, *z*, (*xz*), *t*, *p*, *q*, and (*pq*).

The reason to count all different elements in the abstract chunking that results from a code is twofold. First, the number of different symbols in the abstract chunking equals the number of pattern symbols in the code and can therefore be said to reflect the amount of irregularity (residual nonidentity) in the code (cf. Collard & Buffart, 1983). Second, the chunks in the abstract chunking reflect the hierarchy in the code, and therefore the number of different chunks can be said to quantify hierarchy in terms of irregularity at higher hierarchical levels. So, the I_{new} load measures code complexity in terms of irregularity at all hierarchical levels.

Two further examples may be illustrative. In both Figures 2 and 3, the hierarchical chunking is also the abstract chunking. So, the code S[2∗((*a*)(*b*))] in Figure 2 gets I_{new} = 3, since the abstract chunking ((*a*)(*b*))((*a*)(*b*))(*b*)(*a*)(*b*)(*a*) contains three different elements: *a*, *b*, and ((*a*)(*b*)). Similarly, the code <(*k*)>/<S[((*x*)),((*y*))]> in Figure 3 gets I_{new} = 5, since the abstract chunking ((*k*)(*x*))((*k*)(*y*))((*k*)(*x*)) contains five different elements: *k*, *x*, ((*k*)(*x*)), *y*, and ((*k*)(*y*)).

The I_{new} load is more plausible than the I_{old} load since, unlike the I_{old} load, the I_{new} load does not depend on syntactical artifacts of the coding model. Moreover, the theoretical significance of the I_{new} load is enhanced by its strong relation with the ISA rules. For, the I_{new} load can be applied only to transparent coding rules, since it requires hierarchical chunking. So, it cannot be applied to, e. g., the D or the T rule, as these coding rules are not transparent (nor holographic, by the way). In that sense, the I_{new} load differs essentially from all other measures proposed so far, and provides a possible way out of an empirical deadlock. For it has been difficult to decide empirically which set of coding rules is most appropriate, since the complexity measure seems more decisive than the coding rules (cf. Simon, 1972). Now, however, empirical support for the I_{new} load automatically implies support for (the transparent character of) the ISA rules.

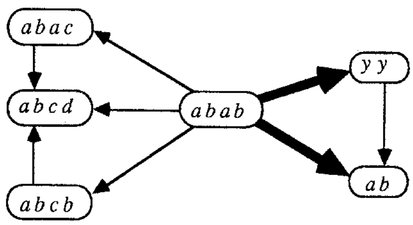

Fig. 4. Small part of network, around interpretation *abab*. The two bold links are substitution links, the other links are inclusion links. For instance, *yy* is inclusion-linked to *ab*, since *ab* contains the same number of elements, but less identity of elements. Furthermore, *abab* is substitution-linked to *yy* and *ab*, since substituting *ab* for each *y* in *yy* yields *abab*

The dynamic-network approach

The dynamic-network approach initiated by Buffart (1986, 1987) can, on the one hand, be seen as a network implementation of the static-coding approach discussed above. On the other hand, however, it is claimed to be based on Buffart's (1987) concept of *hierarchical completeness* (see below). In this section, the network model as such will be considered in a discussion first of the network structure and then of the measurement of Prägnanz.

The network structure

The network consists of nodes and two kinds of link. Each node contains an abstract code (see above) representing an allowed pattern interpretation. All allowed interpretations are thus represented in the network, and are assumed to be given precisely by the DIT rules. So, in other words, each node contains an abstract DIT code.

The links between the nodes represent perceptual inference relations between the interpretations. One kind of link represents so-called *substitution* relations, and may be illustrated by the three abstract DIT codes *abab, yy*, and *ab*. Note that *abab* can be obtained by substituting *ab* for each symbol *y* in *yy*. Because of that, the network node representing *abab* gets two efferent substitution links: one link to the node representing *yy*, and one link to the node representing *ab* (see Figure 4). The series *yy* is said to represent the superstructure of *abab*, and *ab* its substructure. But there are restrictions. For instance, the series *aabbc* could be said to have a superstructure *ayz* with substructures *y* = *ab* and *z* = *bc*. This, however, is not allowed, because it introduces an identity between a superstructure symbol and a substructure symbol (the symbols *a*), and also because it introduces an identity between the substructure symbols *b* which belong to different substructures. In other words, beside identities in the superstructure, new identities may be introduced only inside a substructure. For instance, for *aabbc*, an allowed superstructure is *yzc* with substructures *y* = *aa* and *z* = *bb*.

The other kind of link represents so-called *inclusion* relations, and may be illustrated as follows. The abstract DIT codes *abab* and *abac* both express the identity of the first and third elements, but *abab* moreover expresses the identity of the second and fourth elements. So, if the interpretation represented by *abab* fits (i.e., is possible for) a given input pattern, then, automatically, also the interpretation represented by *abac* fits that input pattern. Therefore, the interpretation represented by *abac* is said to be included in the interpretation represented by *abab*. Because of that, the network node representing *abab* gets an efferent inclusion link to the node representing *abac* (see Figure 4).

The measurement of Prägnanz

In the dynamic-network approach, the static network structure (nodes and links) is seen as the carrier of a dynamic activation-spreading process. Although this process is not a topic in the present paper, its general idea may be explained as follows. The Prägnanz of a given input pattern is assumed to be influenced by history, i.e., by earlier input patterns. That is, the current strength, due to earlier input, of an interpretation is given by an activation value for the node that represents the interpretation. Then, in reaction to a following input, activation-spreading functions determine how the activation spreads through the links between the nodes. Finally, a node that ends up with the highest activation value is taken to represent the preferred interpretation and therefore determines the Prägnanz of the given input pattern. For more details about the implementation of the network model, see Mellink and Buffart (1987), Van der Vegt, Buffart, and Van Leeuwen (1989), and Van Leeuwen, Buffart, and Van der Vegt (1988).

Resonance as Prägnanz measure. In the present paper, I shall focus on the measure of Prägnanz as presented in Van Leeuwen and Van den Hof (1991). That paper is not directly concerned with the dynamic activation-spreading process, but with finding a "static" indication about how the dynamic-network model may be further developed. To that end, a static Prägnanz measure, named *Resonance*, is derived from the static network structure. The Resonance measure implies that the Prägnanz of a pattern is determined and quantified by simply the number of all *resonating* network nodes, i.e., the number of all allowed interpretations that fit the pattern. For instance. if Figure 4 represented the entire network and the series *aaaa* were given as input pattern, then the Resonance measure for that pattern would yield the value 4, as the four network nodes representing *abab, abac, abcb*, and *abcd* would resonate.

In Van Leeuwen and Van den Hof (1991), the Resonance measure was tested, together with the I_{old} load from the coding approach, in an experiment involving tasks such as pattern segmentation, goodness rating, and recall. The experimental data showed that the Resonance measure performs slightly better than the I_{old} load (see below). The authors concluded that a static Prägnanz measure based on pattern coding "does not exist" (p. 442), and that a static Prägnanz measure may make sense only if it is based on that notion of resonance. They argued that this "static" indication suggests that, in reaction to a given input pattern, probably only the resonating network nodes should take part in the dynamic activation-spreading process. This

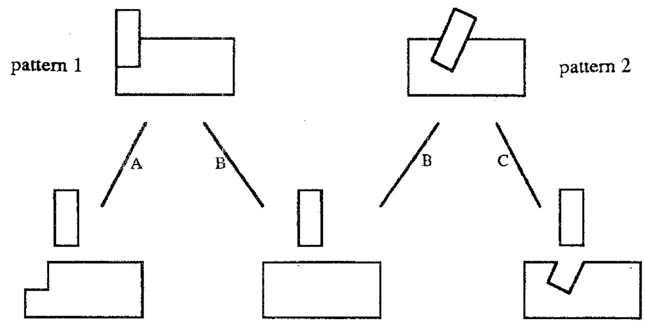

Fig. 5. Generally, for both patterns 1 and 2, the figure-ground interpretation (B) is preferred to the mosaic interpretation (A respectively C), on the basis of an intrapattern judgement for each pattern. For pattern 2, however, the figure-ground interpretation is generally considered to be stronger (more *prägnant*) than for pattern 1, which implies an interpattern judgement of the two patterns

would imply a revision of the activation-spreading functions proposed in earlier studies.

Discussion. The suggested revision of the activation-spreading functions might be a good idea, as it would imply that a given input pattern wakes up only those network nodes that may be relevant for that pattern. The question, however, is whether the Resonance measure (which is seen as a typical network measure) is responsible for the suggested revision. For, by definition, in the static-coding approach of SIT also, only all allowed interpretations of a given pattern are involved in the search for the preferred one. This implies that if the dynamic-network approach had been conceived as a network implementation of SIT, then the suggested revision would be nearly self-evident. Moreover, it implies that the Resonance measure could be adopted in SIT just as well. So, on the one hand, the fact that the Resonance measure performs (slightly) better than the I_{old} load does not mean at all that a static Prägnanz measure based on pattern coding does not exist. On the other hand, it does mean that the Resonance measure has to be taken seriously in SIT as well. To that end, the following two points are relevant.

First, the Resonance measure does not yield preferred pattern interpretations, and simply implies that pattern Prägnanz is determined by the number of all possible interpretations of a pattern. This contrasts with the I loads (I_{old} and I_{new}), which imply that pattern Prägnanz is determined (primarily) by one interpretation of a pattern, namely by the simplest, which is also assumed to be the preferred one. This means that the Resonance measure can be applied only to interpattern judgements, whereas the I loads can be applied to both interpattern and intrapattern judgements (see Figure 5). This is relevant to the experimental data.

The second point is that the Resonance measure may perform (slightly) better than the I_{old} load, but performs worse than the I_{new} load. The I_{new} load, however, is based on the concept of accessibility and cannot be applied to, e.g., the DIT rules, which are claimed to be derived from the concept of *hierarchical completeness* (Buffart, 1987). So, if accessibility and hierarchical completeness were equally valid concepts, then the theoretical plausibility of the I_{new} load would still be challenged. However, in the next section, Buffart's claim will be shown to be incorrect.

Hierarchical completeness versus accessibility?

In Buffart (1987), the more conceptual approach in Buffart (1986) was formalized for symbol series, with the following central assumption:

Hierarchy assumption: *Allowed* interpretations are given by representations that can be *decomposed* into independent substructures plus a superstructure that relates those substructures, such that the superstructure is *hierarchically complete* and, recursively, each of the substructures is *allowed.*

Buffart claims that such hierarchically complete decompositions correspond one to one to the chunkings as induced by the DIT rules. Here, I shall rephrase only the main points of Buffart's formalization, which are clear enough to show that it is inconsistent.

Representations

Buffart's first step is to consider permutations, i.e., operations that interchange two or more symbols in a series. For instance, for the 5-symbol series S = *aabab*, permuting the second and third symbols, yields the series *abaab*. This permutation will be denoted by P = [1-3-2-4-5] in which the numbers are indices indicating the resulting order of the symbols. The application of P to S will be denoted by $P*S$.

Definition 1: An *invariance permutation* for a symbol series S is an operation that *describes* identity of symbols in S by interchanging symbols in S such that a series equal to S results;
a *representation* of a symbol series S is a *group* (see below) of invariance permutations for S;
a *characterization* of a symbol series S is a representation of S that describes all identity of symbols in S.

The notion of a *group* is standard mathematics, and refers to a set with nice mathematical properties. The main property is *closedness*. That is, if e.g., the two permutations P = [1-4-3-2-5] and Q = [2-1-5-4-3] belong to a representation R, then R also contains the permutation $P*Q$ = [2-4-5-1-3] ($P*Q$ means: permute the indices in Q according to P). Furthermore, a representation of an n-symbol series contains the (not really permuting) unity permutation E_n = [1-2-3-...–n] which is required to form a group.

For instance, the set of permutations R = {E_5,[2-4-3-1-5],[4-1-3-2-5]} is a group. R is a representation of series S = *aabab*, as R consists of invariance permutations for S (describing the identity of the symbols a). R is not a characterization of S, as R does not describe the identity of the symbols b. For series T = *aabac*, however, R is a characterization since R describes all identity of symbols in T. This holds even though there are invariance permutations for T, e.g. [2-1-3-4-5] that are not contained in R.

Clustering

Buffart's next step is to consider *clusters* in a series. A cluster is not simply a subseries, but a subset of the symbols in a series. For instance, the subseries *abc* and *acb* in the series *abcacb* are not identical, but the clusters {*abc*} and {*acb*} are identical.

Definition 2: A *cluster* is a subset of the symbols in a symbol series. A *clustering* in an *n*-symbol series is an operation that yields a series of *m* disjunct clusters ($1 \leq m < n$) which, together, contain all symbols in the series.

Definition 3: A representation *R allows* some clustering, if *R* possibly permutes symbols inside the resulting clusters, but only permutes symbols from one cluster to another by permuting clusters "as a whole", i.e., with one permutation.

For instance, consider representation $R = \{E_5,[1\text{-}4\text{-}5\text{-}2\text{-}3]\}$, which characterizes series $s_1s_2s_3s_4s_5 = cabab$. According to Definition *3, R* allows clustering into the cluster series $\{s_1\}\{s_2s_3\}\{s_4s_5\} = \{c\}\{ab\}\{ab\}$, since *R* permutes the last two clusters "as a whole".

A cluster is a subset, so, in principle, the symbols for a cluster may be gathered in any order from anywhere in the series. This implies that, if a representation *R* allows a clustering into some cluster series, then *R* allows any clustering that yields the same clusters (in some order). And also, in order to allow a clustering, a representation *R* does not necessarily have to contain permutations of clusters as a whole. Thus, for instance, any representation of *aabab* allows clustering into {*bb*}{*aaa*}.

Two special cases

On the basis of the formalization so far, Buffart distinguishes two special cases: *indivisible* representations and *diagonal* representations. An indivisible representation is a representation that does not allow any clustering except for the "meaningless" clustering into only one cluster. In Buffart (1987), indivisible representations are taken to be hierarchically complete "by definition". Now, indivisible representations characterize series in which all symbols are identical, as in *aaaaa.* Thus, together with the Hierarchy Assumption, indivisible representations correspond one to one to codes obtained by means of the I rule. No other relation between hierarchical completeness and the I rule is given, so in fact, the I rule results "by definition" and not as an "implication" of hierarchical completeness.

Definition 4: A *diagonal* representation *D* characterizes a series consisting of at least two "independent" subseries, i.e., with only identities inside such a subseries; a sub-representation of *D* that describes precisely all identity of symbols in only one of those independent subseries is a *subdiagonal.*

For instance, the series *aababccc* can be characterized by a diagonal representation, since it has *aabab* and *ccc* as independent subseries. In a personal communication, Buffart confirmed that in Buffart (1987) it was wrongly suggested that all symbols in an independent subseries have to be identical (then, his claim could be rejected much more easily). A diagonal representation allows clustering into clusters that each contain all symbols of an independent subseries. Such a cluster corresponds to a chunk (i.e., a subseries, not a randomly ordered set of gathered symbols). Probably because of that, in Buffart (1987), diagonal representations are hierarchically complete "by definition", and serve as the basis for the general definition of hierarchical completeness.

Hierarchical completeness

Before giving the general definition of hierarchical completeness (Definitions 6 and 7), we first need some standard mathematics, as given in Definition 5. Definition 5 may seem rather complex, but is followed by an explanation in simple words.

Definition 5: Let representation *K* consist of permutations K_i ($i = 1,2,\ldots,m$), and let representation *L* consist of permutations L_i ($i = 1,2,\ldots,m$) such that, for some permutation P: $P*K_i = L_i*P$ ($i = 1,2,\ldots,m$); then *K* and *L* are *equivalent* representations, and P is an *equivalence transformation* from *K* to *L*.

In simple words, Definition 5 states the following. Let *K* characterize the series S_1 and let *L* characterize the series $S_2 = P*S_1$. So, series S_1 and S_2 contain the same symbols, but in a different order. Then, *K* and *L*, are equivalent if the permutations in *K* correspond one to one to those in *L*. That is, if some permutation in *K* permutes certain symbols in S_1, then *L* contains a permutation that permutes precisely the same symbols in S_2, even though these symbols are moved by *P.*

In contrast to what Buffart seems to conceive, the equivalence of *K* and *L* implies that any cluster allowed by *K* is also allowed by *L.* For instance, representation $K = \{E_5,[4\text{-}5\text{-}3\text{-}1\text{-}2]\}$ of $S_1 = s_1s_2s_3s_4s_5 = abcab$ and representation $L = \{E_5,[3\text{-}5\text{-}1\text{-}4\text{-}2]\}$ of $S_2 = s_1s_2s_4s_3s_5 = abacb$ are equivalent under transformation $P = [1\text{-}2\text{-}4\text{-}3\text{-}5]$. For S_1, *K* allows clustering into $C_1 = \{s_1s_2\}\{s_3\}\{s_4s_5\} = \{ab\}\{c\}\{ab\}$. For S_2, *L* allows clustering (see Definition 3!) into $C_2 = \{s_1s_2\}\{s_4s_5\}\{s_3\} = \{ab\}\{ab\}\{s\}$, which contains the same clusters as C_1.

Definition 6: Two representations, say *K* and *L*, are *hierarchically equivalent* if they are related by a *hierarchical equivalence condition*; such a condition is given by an equivalence transformation (from *K* to *L*) plus two clusterings (one allowed by *K*, the other allowed by *L*) that yield the same set of clusters.

Definition 7: A non-indivisible and non-diagonal representation *F* is *hierarchically complete* if for each diagonal representation *D*, with *D* equivalent to *F*, it holds that each hierarchical equivalence condition, which relates a sub-representation Sub(*F*) of *F* and a subdiagonal Sub(*D*) of *D*, also relates *F* and *D*.

On the basis of these definitions, Buffart tries to prove several theorems, which can be summarized as follows.

Theorem: A nonindivisible and nondiagonal hierarchically complete representation allows a clustering into two clusters which correspond one-to-one to the two arguments in either a D or a T form.

This theorem together with the Hierarchy Assumption, should complete Buffart's claim that hierarchical completeness is the basis for the DIT rules (remember that the I rule results "by definition"). However, consider the following counter-example.

Counter example

With the DIT rules, all identity in the series *abacab* can be described by only the code $D[(a),T[((b)),((c))]]$. So, if Buffart's claim is correct, there has to be a hierarchically complete representation, say *F*, that characterizes *abacab*. This *F* has to be like *F* in Definition 7, since *abacab* cannot be characterized by a diagonal or indivisible representation. So, there also has to be a diagonal representation *D* as in Definition 7. This *D* has to characterize a series with independent subseries *aaa* and *bcb* corresponding to the arguments in the D form $D[(a),(b)(c)(b)]$.

There are several possibilities of choosing F and D, but the results would be the same. For instance, let *D* characterize *aaabcb*, and let *D* be the group of permutations generated (remember the closedness of a group) by the two permutations D_1 = [2-3-1-4-5-6] and D_2 = [1-2-3-6-5-4]. Let *F* be the group of permutations generated by the two permutations F_1 = [3-2-5-4-1-6] and F_2 = [1-6-3-4-5-2]. Then, *F* and *D* are equivalent under transformation *P* = [1-3-5-2-4-6] which transforms *abacab* into *aaabcb*.

Now, take Sub(F) = $\{E_6,F_2\}$ and Sub(D) = $\{E_6,D_2\}$. Sub(D) is the subdiagonal that only permutes inside the independent subseries *bcb* of *aaabcb*. Sub(F) and Sub(D) are, just like *F* and *D*, equivalent under transformation *P*. Furthermore, Sub(F) and Sub(D) are related by many hierarchical equivalence conditions: among others, by means of clusterings by which the three symbols *a* are not clustered into one cluster, e.g., {*aa*}{*abcb*}.

Now, such clusterings are not allowed by *F*, nor by *D*. So, according to Definition 7, *F* is not hierarchically complete. But this contradicts Buffart's claim that *abacab* can be characterized by a hierarchically complete representation.

Discussion

In the counterexample, it may look strange that the subdiagonal Sub(D) of the independent subseries *bcb* is used for clustering in the other independent subseries *aaa*. However, this trick is not only possible with Buffart's formalization, it is also precisely the trick used by Buffart to prove that series like *abacab* can be characterized by a hierarchically complete representation. This shows that Buffart's formalization is inconsistent as, apparently, anything can be proven, which renders it unacceptable.

One problem with Buffart's formalization is the use of clusters and the definition of allowed clustering: permutations inside one cluster may affect the situation between clusters and inside other clusters. Nevertheless, Buffart's mathematical tools, i. e., groups of invariance permutations, are interesting as means to investigate hierarchy, at least, if chunks are used instead of clusters. Then, for instance, it is possible to formalize a restriction on the hierarchical structure of representations (I shall not elaborate on this), leading to the substitution links in the network model (see above). It is also possible to provide an alternative formalization of the notion of transparency as given within the concept of accessibility (Van der Helm, 1988).

Another problem with Buffart's formalization is that those tools only deal with hierarchy and seem insufficient to obtain a plausible set of coding rules. It is as if one tries to select the best basketball players by considering only their lengths: length is relevant, but is not what the game is basically about. Analogously, although hierarchy is relevant, coding rules basically describe regularity. Within the concept of accessibility, for instance, the notion of holography is used as basis for regularity. Buffart's tools may be suited for an a-posteriori description of regularity, but seem unsuited for an a-priori basis of regularity.

Summarizing, one has to reject Buffart's formalization and his claim that the concept of hierarchical completeness is the basis for the DIT rules. This implies, on the one hand, that the network approach has to search for its real roots in the coding approach of SIT. On the other hand, it implies that the concept of hierarchical completeness does not undermine the ISA rules and the I_{new} load, which are based on the concept of accessibility.

Table 1. Schematic overview of approaches of Prägnanz

Approach	Implementation	Measure for a given pattern
Network approach	Network with all DIT codes as nodes, and substitution relations and inclusion relations as links; incorrectly claimed from hierarchical completeness	Resonance measure: number of all nodes that fit the pattern
Coding approach	Encoding scheme with mainly the DIS rules; intuitively chosen	I_{old} load: minimal number of symbols and I and S forms needed for a code of the pattern.
	Encoding scheme with the ISA rules; based on accessibility	I_{new} load: minimal number of symbols and chunks needed for a code of the pattern; requires transparent coding rules

Experimental tests of the Prägnanz measures

In the previous sections, I have discussed two approaches to Prägnanz (a network approach, and a coding approach), three sets of allowed interpretations (obtained by the DIT rules for the network approach, and by the DIS rules or the ISA rules for the coding approach), two theoretical frameworks (hierarchical completeness for the DIT rules, and accessibility for the ISA rules), and three static measures of Prägnanz (Resonance for the network approach, and I_{old} load and I_{new} load for the coding approach). Table 1

Table 2. Schematic overview of compared models

Name	Implementation	Measure of Prägnanz
Resonance model	Network of DIT-codes	Resonance-measure
I_{vL} model	DIS rules	I_{old} load
I_{old} model	ISA rules	I_{old} load
I_{new} model	ISA rules	I_{new} load

provides a schematic overview. I have argued that, in Buffart (1987), hierarchical completeness is incorrectly claimed to be the basis for the DIT rules. So, in fact, the DIT rules have to be seen as merely an intuitively chosen variation on Leeuwenberg's (1969, 1971) DIS rules which were chosen also intuitively. Furthermore, I have argued that accessibility implies a strong relation between the ISA rules and the I_{new} load, since the I_{new} load requires transparent coding rules.

In this section, I shall repeat the analysis of the experimental data that were obtained in Van Leeuwen and Van den Hof (1991), for four models (see Table 2). First, from Van Leeuwen and Van den Hof (1991), I shall copy the results for the Resonance-model (network of DIT codes plus Resonance measure) and for the I_{vL} model (DIS rules plus I_{old} load). Second, I shall add the I_{old} model (ISA rules plus I_{old} load) and the I_{new} model (ISA rules plus I_{new} load). The Appendix in Van Leeuwen and Van den Hof (1991) and the Appendix of the present paper together provide the necessary data.

One reason for adding the I_{old} model is to get a more detailed insight into the decisiveness of the chosen complexity measure as compared with that of the chosen coding rules. That is, the I_{old} model is constituted by the complexity measure of the I_{vL} model and the coding rules of the I_{new} model. Another reason is to get a more acceptable test of the I_{old} load than the one provided by the I_{vL} model. That is, beside the actual three DIS rules, the I_{vL} model employs several other coding rules, among which one, the so-called *Continuation-rule*, which has been applied wrongly, i. e., not as it was meant to be applied. Several other minor flaws in the experimental set-up in Van Leeuwen and Van den Hof (1991) will be indicated below. However, the main reason for the present analysis is to show unquestionably, i. e., with the same experimental data, that the conclusions in Van Leeuwen and Van den Hof (1991), which favor the Resonance model, are untenable if the I_{new} model is taken into account.

Updated experimental analysis

For a detailed description of the experiment, see Van Leeuwen and Van den Hof (1991). In short, in each of 128 stimuli, the target was a seven-element series of empty and full circles, similar to the pattern in Figure 1 B. There were five tasks: grouping (free pattern segmentation), immediate recall (recall right after 200-ms target presentation), unpracticed goodness rating (without any other task), practiced goodness rating (after grouping task and immediate-recall task), and intermediate recall (recall of 1,000-ms presented

Table 3. Correlations between theoretical models and experimental tasks (N = 128 series)

Model	Grouping entropy	Unpracticed rating	Practiced rating	Immediate recall	Intermediate recall
I_{new}	.732*	.760*	.804*	.569*	.568*
Resonance	–.681*	–.647*	–.720*	–.557*	–.568*
I_{vL}	.579*	.613*	.629*	.305†	.412*
I_{old}	.571*	.534*	.565*	.384*	.506*

† $p<.0005$
* $p<.0001$

Table 4. Intercorrelations for the theoretical measures (N = 128 series)

Models compared	Correlations
I_{new}, Resonance	–.735*
I_{new}, I_{vL}	.630*
I_{new}, I_{old}	.664*
Resonance, I_{vL}	–.686*
Resonance, I_{old}	–.753*
I_{vL}, I_{old}	.679*

* $p <.0001$

Table 5. Within tasks, t test on differences between theoretical models (df = 125)

Models compared	Grouping entropy	Unpracticed rating	Practiced rating	Immediate recall	Intermediate recall
I_{new} – Resonance	1.209	2.726**	2.290*	0.231	0.000
I_{new} – I_{vL}	2.994**	3.049**	3.967**	4.187**	2.472*
I_{new} – I_{old}	3.269**	4.751**	5.495**	3.068**	1.051
Resonance – I_{vL}	2.010*	0.661	1.920	4.316**	2.676**
Resonance – I_{old}	2.407*	2.368*	3.557**	3.321**	1.211
I_{vL} – I_{old}	0.143	1.425	1.184	–1.196	–1.530

* $p <.05$; ** $p <.01$

target after the previous target has been recalled and the next target has been presented). In the grouping and rating tasks, the target was presented until the response was given.

For the rating tasks, mean ratings per stimulus were calculated; ratings were from 1 (very orderly) to 7 (very disorderly). For the recall tasks, mean number of errors per stimulus were calculated: an error is a wrongly recalled element. For the grouping task, agreement among the subjects was quantified per stimulus using the entropy formula $\Sigma\{(x_i/n)ln(x_i/n)\}$ with x_i the number of subjects who gave grouping i, and n the total number of subjects; a higher entropy means less agreement. The results of these calculations are given in the Appendix of Van Leeuwen and Van den Hof (1991).

Table 3 shows the correlations between the theoretical and experimental data. As in Van Leeuwen and Van den Hof (1991), the Resonance measure was log-transformed by means of the formula ln(1+Resonance). Table 3 shows that the I_{vL} model and the I_{old} model score considerably lower than the Resonance model and the I_{new} model, and that the I_{new} model scores equal to the Resonance model for the intermediate-recall task and better for the other tasks. Table 5 shows the result of a t test on the differences between the (absolute) correlations, within tasks, using the (absolute) intercorrelations between the models as given in Table 4. In short, Table 5 shows that the I_{vL} model and the I_{old} model do not score significantly different, and that the I_{new} model scores significantly better than the Resonance model for the rating tasks.

Discussion

The three static-coding models. The nonsignificance of the difference in performance between the I_{vL} model and the I_{old} model is relevant. It confirms that the chosen complexity measure (I_{old} load) is probably more decisive than the chosen coding rules (DIS rules versus ISA rules) and illustrates the empirical deadlock in deciding between different sets of coding rules (cf. Simon, 1972).

The results for the I_{new} model (ISA rules plus I_{new} load), however, show a way out of that deadlock. The I_{new} model scores highly significantly better than both the I_{vL} model and the I_{old} model for four tasks, and significantly better respectively better for the fifth task (see Table 5). This confirms the finding in Van der Helm et al. (1992). In that study, the I_{old} model and the I_{new} model were tested on a forced-choice grouping task for serial stimuli, similar to the ones in the experiment presently discussed, but with the following differences. First, several different types of stimulus element were used, i.e., not just one type (empty and full circles) as in Van Leeuwen and Van den Hof (1991). Second, the stimuli were selected more critically, so that the theoretical predictions discriminate more between the models. Third, instead of comparing the I loads of simplest codes with (dis)agreement among subjects about segmentations, the segmentations induced by simplest codes (see above, concerning Figure 1B) were compared directly to the segmentations given by the subjects. Thus, one gets a more direct test of the model assumptions. In that setting, the I_{new} model was shown to be highly significantly better than the I_{old} model, i.e., scored a five-times larger percentage of significantly correct predictions.

So, as far as the static-coding models are concerned, the I_{new} model is clearly superior. This implies not only support for the I_{new} load as an appropriate measure of Prägnanz, but since the I_{new} load requires transparent coding rules, it also implies support for the ISA rules as an appropriate set of coding rules.

The I_{new} model versus the Resonance model. Table 5 shows that the static-coding based I_{new} model scores the same as the dynamic-network based Resonance model for the intermediate-recall task, slightly better for the immediate-recall and the grouping tasks, and significantly better for both rating tasks. So one may reject Van Leeuwen and Van den Hof's (1991) conclusion that a static Prägnanz measure makes sense only if it is based on the notion of resonance. Nevertheless, the performance of the Resonance model requires some further discussion.

The results on the grouping task should be taken with a grain of salt. The Resonance model is not suited for intrapattern judgements such as are required in the grouping task, since it does not predict segmentations (see above). This was the reason for using agreement among subjects about segmentations. But therefore the actual segmentations as predicted by the I_{new} model are not taken into account, so that the I_{new} model may be incorrectly taken to perform well or badly. For instance, agreement about a segmentation may be predicted correctly, but that segmentation may differ from the predicted segmentation. So agreement among subjects yields an inaccurate test.

The two goodness-rating tasks require interpattern judgements, for which both the Resonance model and the I_{new} model are suited (again, see above). So the fact that the I_{new} model scores significantly better can be taken as a reliable result.

As was reported in Van Leeuwen and Van den Hof (1991), the intercorrelation between the ratings without any other task (unpracticed) and after two other tasks (practiced) is very high ($r = .945$, $p < .0001$). This suggests that *long-term* history (i.e., the other two tasks) hardly affects the judgements of subjects, and that a really dynamic network process does not have to account for it. This agrees with Mens (1988), who found that an SOA of several seconds between a prime and a stimulus is already large enough to prevent influence of the prime on the stimulus interpretation. Mens also found (Leeuwenberg, Mens, & Calis, 1985) that two stimuli presented with an SOA of less than about 30 ms are interpreted as one stimulus and independent of the *short-term* history (i.e., the presentation order). Only in case of *medium-term* history, i.e., between about 30 ms and 1,000–2,000 ms, Mens found (Mens & Leeuwenberg, 1988) some influence in the sense that good, but without-history-second-best, alternative interpretations may become almost just as preferred as the without-history-best interpretation. So a really dynamic network process could be useful in simulating such a medium-term history effect (see also next section).

Whereas the grouping and rating tasks involve a process of selecting a preferred interpretation, the two recall tasks involve a process of retrieving an already selected-and-stored preferred interpretation. For the recall tasks, the basic assumption is that good patterns are remembered better. Despite the shorter presentation times in the immediate-recall task, subjects made fewer mistakes than in the intermediate-recall task (see Appendix in Van Leeuwen & Van den Hof, 1991). Furthermore, the tasks not only yield the same low correlations with both the I_{new} model and the Resonance model (see Table 3), but also show a rather low intercorrelation ($r = .346$, $p < .01$), as was reported in Van Leeuwen and Van den Hof (1991).

In these recall tasks, the low correlations with the models suggest that the Prägnanz of the pattern to be recalled is not the only important factor – at least, if one accepts the I_{new} load and (to a lesser degree) the Resonance measure as indicators of pattern Prägnanz, on the basis of the rating tasks. This agrees with the between-task difference in the number of errors, which suggests that the recall of a pattern is influenced more by the complexity of the task than by the Prägnanz of that pattern. The low intercorrelation may suggest that in the intermediate-recall task, the complexity of the task depends on the Prägnanz of the target after which the previous target has to be recalled.

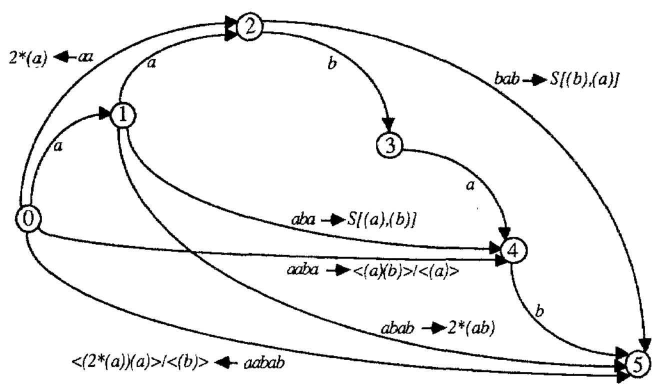

Fig. 6. In this directed graph, each edge represents a subseries of the 5-symbol series *aabab*. Only those subseries that consist of one symbol, or can be encoded entirely into one ISA form are represented. Thus, each path from vertex 0 to vertex 5 represents a possible code for the entire *aabab*. For instance, the path along vertices 0, 2, 5, reseriepresents the code 2*(*a*)S[(*b*),(*a*)]. In principle, the number of codes is combinatorially explosive. But this graph shows that they can be represented in a nonexplosive way, using the static interaction (common parts) between the codes

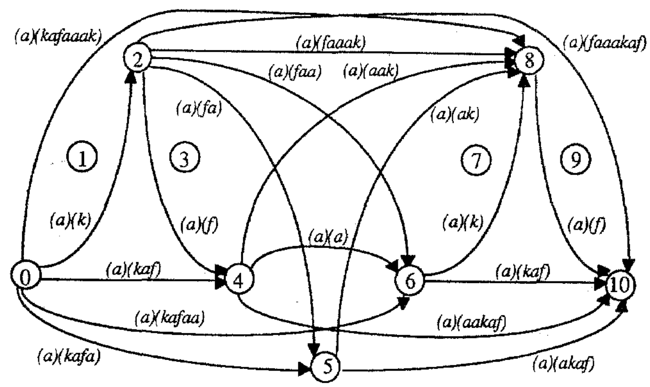

Fig. 7. The paths vertex 0 to vertex 10 in this directed graph represent all A forms <(*a*)>/<(x_1)(x_2)...(x_n)> into which the series *akafaaakaf* can be encoded. Each edge represents a possible pair (*a*)(x_i), so that e.g., the A form <(*a*)>/<(*k*)(*f*)(*a*)(*k*)(*f*)> is represented by the path along vertices 0, 2, 4, 6, 8, 10, and the A form <(*a*)>/<(*kaf*)(*a*)(*kaf*)> by the path along vertices 0, 4, 6, 10. In principle, the number of A forms is combinatorially explosive. This graph, however, shows that they can be represented in a nonexplosive way, using the static interaction (common parts) between the A forms

This seems to be an aspect that is well suited for simulation by means of a really dynamic network process (see also next section). For instance, the better the Prägnanz of a target (so, highly activated), the more difficult the previous target is recalled (as it lost much activation to the intervening target).

Dynamic versus static?

In the static-coding I_{new} model, visual perception is conceived of as a modular process with the raw pattern as input and a simplest interpretation as output. This interpretation is considered to be embedded in stored-knowledge structures, and therefore may play different roles, depending on knowledge (history, intention). So the static-coding approach does not deny a dynamic process, once the simplest interpretation has been embedded in stored-knowledge structures, but it is that simplest interpretation in which the static-coding approach is interested and which is not assumed to be dependent on knowledge (cf., e.g., Rock, 1985).

This starting point implies that the static-coding approach is less suited for explaining effects such as the medium-term history effect or the recall effect mentioned above. I therefore agree that a dynamic-network model may be a powerful instrument in simulating such effects. This simulation as such, however, is not yet an explanation. The explanatory power could come from a theory that forms the basis for the nodes, links, and activation-spreading processes, as used in the network.

Now, on the one hand, Buffart's (1986, 1987) theory about hierarchical completeness does not have implications with respect to activation-spreading processes. Furthermore, I have shown above that Buffart's theory is inconsistent, and therefore cannot be accepted as a basis for the nodes and links in the dynamic-network model in Van Leeuwen and Van den Hof (1991). On the other hand, remember that the concept of accessibility, specified by the notions *holography* and *transparency*, provides a sound and firm theoretical basis for the I_{new} model, i.e., for both the ISA rules and the I_{new} load. Furthermore, the I_{new} model showed the best experimental performance. These static results suggest that it might be relevant to develop a dynamic-network model that starts from the I_{new} model, as follows.

Consider a network in which the nodes represent all ISA codes, while the links represent so-called regularity relations and hierarchy relations between those ISA codes. These two kinds of relation can be based on the notions of holography and transparency, respectively, and would be more or less analogous to, respectively, the inclusion and substitution relations discussed above. Now, let the activation-spreading functions be such that, for a arbitrary first-input series (i.e., without history), fitting ISA codes with minimal I_{new} load end up with the highest activation value. Indications that this is feasible can be found in Van der Vegt et al. (1989). So, without the influence of history, this network model would simply simulate the I_{new} model. This way, the activation-spreading functions would not be just functions found by trial and error that simulate empirical data, but would be based on a theoretically and empirically supported starting criterion.

A probably more intriguing solution may now be available for the following problem. In the DIT-codes network, each possible DIT code is represented by one node in the network. This is unrealistic, as the number of DIT codes is combinatorially explosive and, for larger networks, would exceed the number of neurons in the human brain. According to Mellink and Buffart (1987), the number of nodes is about $4^k/3$ for a network with interpretations from length (number of symbols) 1 to k. A similar problem exists (or, rather, existed) for the I_{new} model since a simplest code has to be selected out of a combinatorially explosive number of possible ISA codes (cf. Hatfield & Epstein, 1985). For the I_{new} model, however, this problem has been solved on the basis of the concept of accessibility. That is, the simplest codes can be found with all possible codes being taken into account, but without each of those codes being considered separately (Van der Helm & Leeuwenberg, 1986, 1991; Van der Helm, 1988). The key to this solution is illustrated in Figures 6 and 7, showing graphs in which each edge represents a code part that may belong to several different codes, while each code is represented by a path in the graph. Then, in order to get a simplest code, the so-called *shortest-path-method* (Dijkstra, 1959) can be applied, requiring only a polynomial amount of processing time. So, this way of representing codes implies a nonexplosive amount of processing time and storage space.

Now, the way of representing codes, as given in Figures 6 and 7, might also be applicable in the ISA-codes network. That is, an interpretation is not represented by one node, but by a set of nodes or, in other words, by a *trace* in the network. Then, one still faces the so-called *binding-problem*, i.e., how to distinguish between different and possibly overlapping traces. For that problem one could turn to oscillatory mechanisms: each trace shows a periodicity in activation spreading, but different traces at different moments in time (see, e.g., Goebel, 1990; Crick & Koch, 1990).

For the ISA-codes network, this way of representing codes may not only avoid an explosive number of network-nodes, but may also affect positively the performance of the network. For, the essence of a dynamic-network model is that the result depends on the interaction of interpretations, and Figures 6 and 7 show a more sophisticated representation of the static interaction (common parts) of interpretations, which may enable a more sophisticated simulation of the dynamic interaction of interpretations. So all in all, the dynamic-network approach does not have to be seen as an alternative opposing the static-coding approach, but rather as an enriching alternative based on the same principles.

Summary

In an investigation into pattern Prägnanz, Leeuwenberg's (1969, 1971) static-coding approach (which uses pattern information only) was contrasted with the dynamic-network approach (which also uses history) initiated by Buffart (1986, 1987). Within this framework, I have discussed the experiment in Van Leeuwen and Van den Hof (1991), in which a static network measure of Prägnanz, namely the Resonance measure, was compared with the I_{old} load used in the static-coding approach until several years ago. The better performance of the Resonance measure was taken as a "static" indication for further development of the network model.

With the same experimental data, however, I have shown that the more recent static-coding measure of Prägnanz, namely the I_{new} load, performs better than the Resonance measure. The I_{new} load is derived from the concept of accessibility (Van der Helm & Leeuwenberg, 1991), which is also the basis for choosing the transparent holographic ISA rules as the only coding rules appropriate for obtaining allowed pattern interpretations. The good performance of the I_{new} load also supports the ISA rules, as the I_{new} load requires transparent coding rules. I have argued that these "static" results suggest the development of a network model different from the one proposed in Van Leeuwen and Van den Hof (1991). This is relevant the more since I have shown that Buffart's (1987) theory about hierarchical completeness is inconsistent, and therefore cannot be accepted as a basis for the network model in Van Leeuwen and Van den Hof (1991).

A network model, based on the concept of accessibility, might be shaped such that the activation-spreading functions yield a simulation of the I_{new} model (ISA rules plus I_{new} load), in case the influence of history is not (yet) present. This way, the activation-spreading functions would be based on a theoretically and empirically supported starting criterion. Furthermore, in the network, the allowed interpretations would not have to be represented by one node each, but might be represented by oscillating traces in the network. This way, all allowed interpretations can be represented without the implementation of a combinatorially explosive number of nodes. Then, moreover, the network might enable a more sophisticated account of the dynamic interaction of interpretations and thus may even yield an enrichment of the static-coding approach.

References

Boselie, F., & Leeuwenberg, E. L. J. (1985). Birkhoff revisited: Beauty as a function of effect and means. *American Journal of Psychology, 98*, 1–39.

Buffart, H. F. J. M. (1986). Gestalt qualities, memory structure, and minimum principles. In F. Klix & H. Hagendorf (Eds.), *Human memory and cognitive capabilities: Symposium in memoriam Hermann Ebbinghaus* (pp. 189–204). Amsterdam: North-Holland.

Buffart, H. F. J. M. (1987). Seven minus two and structural operations. In E. E. Roskam & R. Suck (Eds.), *Progress in mathematical psychology* (Vol. 1, pp. 117–149). Amsterdam: North Holland.

Buffart, H. F. J. M., Leeuwenberg, E. L. J., & Restle, F. (1981). Coding theory of visual pattern completion. *Journal of Experimental Psychology: Human Perception and Performance, 7*, 241–274.

Collard, R. F. A., & Buffart, H. F. J. M. (1983). Minimization of structural information: A set-theoretical approach. *Pattern Recognition, 16*, 231–242.

Crick, F., & Koch, C. (1990). Towards a neurobiological theory of consciousness. *Seminars in the Neurosciences, 2*, 263–275.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik, 1*, 269–271.

Goebel, R. (1990). Binding, episodic short-term memory, and selective attention, or why are PDP models poor at symbol manipulation? In D. Touretzky, J. Elman, T. Sejnowski, & G. Hinton (Eds.), *Connectionist models, proceedings of the 1990 summer school.* San Mateo, CA: Morgan Kaufman.

Hatfield, G. C., & Epstein, W. (1985). The status of the minimum principle in the theoretical analysis of visual perception. *Psychological Bulletin, 97*, 155–186.

Helm, P. A. van der (1988). Accessibility and simplicity of visual structures. Ph. D. Thesis, University of Nijmegen, Nijmegen, The Netherlands.

Helm, P. A. van der, & Leeuwenberg, E. L. J. (1986). Avoiding explosive search in automatic selection of simplest pattern codes. *Pattern Recognition, 19*, 181–191.

Helm, P. A. van der, & Leeuwenberg, E. L. J. (1991). Accessibility, a criterion for regularity and hierarchy in visual pattern codes. *Journal of Mathematical Psychology, 35*, 151–213.

Helm, P. A. van der, Lier, R. J. van, & Leeuwenberg, E. L. J. (1992). Serial pattern complexity: irregularity and hiearchy. *Perception, 21*, 517–544.

Hochberg, J. E., & McAlister, E. (1953). A quantitative approach to figural "goodness." *Journal of Experimental Psychology, 46*, 361–364.

Koffka, K. (1935). *Principles of Gestalt Psychology* (1962). 5th ed., London: Routledge & Kegan Paul.

Leeuwen, C. van, Buffart, H., & Vegt, J. van der (1988). Sequence influence on the organization of meaningless serial stimuli: Economy after all. *Journal of Experimental Psychology: Human Perception and Performance, 14*, 481–502.

Leeuwen, C. van, & Hof, M. van den (1991). What has happened to Prägnanz? Coding, stability, or resonance. *Perception & Psychophysics, 50*, 435–448.

Leeuwenberg, E. L. J. (1969). Quantitative specification of information in sequential patterns. *Psychological Review, 76*, 216–220.

Leeuwenberg, E. L. J. (1971). A perceptual coding language for visual and auditory patterns. *American Journal of Psychology, 84*, 307–349.

Leeuwenberg, E. L. J., & Buffart, H. F. J. M. (1984). The perception of foreground and background as derived from structural information theory. *Acta Psychologica, 55*, 249–272.

Leeuwenberg, E. L. J., Mens, L., & Calis, G. (1985). Knowledge within perception: Masking caused by incompatible interpretations. *Acta Psychologica, 59*, 91–102.

Lier, R. J. van, Helm, P. A. van der, & Leeuwenberg, E. L. J. (1993). Integrating global and local aspects of visual occlusion. Submitted.

Mellink, H., & Buffart, H. (1987). Abstract code network as a model of perceptual memory. *Pattern Recognition, 20*, 143–151.

Mens, L. (1988). Primary perception of stimulus structure. Ph. D. Thesis, University of Nijmegen, Nijmegen, The Netherlands.

Mens, L., & Leeuwenberg, E. L. J. (1988). Hidden Figures are ever present. *Journal of Experimental Psychology: Human Perception and Performance, 14*, 561–571.

Rock, I. (1985). Perception and knowledge. *Acta Psychologica, 59*, 3–22.

Simon, H. A. (1972). Complexity and the representation of patterned sequences of symbols. *Psychological Review, 79*, 369–382.

Vegt, J. van der, Buffart, H., & Leeuwen, C. van (1989). The "structural memory": A network model for human perception of serial objects. *Psychological Research, 50*, 211–222.

Appendix

This appendix and the appendix in Van Leeuwen and Van den Hof (1991) together provide the necessary theoretical and experimental data. In the following table, the symbolized patterns are given together with the minimal I_{old} load and the minimal I_{new} load, as computed by the algorithm PISA (Van der Helm et al., 1986, 1991) using only the ISA rules. Generally, several minimum codes are possible; below, a minimum code is given for the I_{old} load, and one for the I_{new} load but only if it cannot be the one for the I_{old} load. The total set of patterns consists of 128 patterns; here, only the first 64 patterns from the appendix in Van Leeuwen and Van den Hof (1991) are given; the second 64 patterns can be obtained from the first 64 patterns in reversed order, by changing symbol *a* into *b* and symbol *b* into *a*.
a represents an empty circle, and *b* a full circle.

Pattern	I_{old} + code		I_{new} + code	
aaaaaaa	2	7*(a)	1	
aaaaaab	3	6*(a)b	2	
aaaaaba	4	5*(a)ba	3	
aaaaabb	4	5*(a)2*(b)	2	
aaaabaa	5	4*(a)b2*(a)	3	
aaaabab	4	<(4*(a))(a)>/<(b)>	3	4*(a)S[(b),(a)]
aaaabba	5	4*(a)2*(b)a	3	
aaaabbb	4	4*(a)3*(b)	2	
aaabaaa	4	S[3*((a)),(b)]	2	
aaabaab	5	3*(a)S[(b)(a)]	3	
aaababa	5	3*(a)2*(ba)	4	
aaababb	5	3*(a)<(b)>/<(a)(b)>	4	
aaabbaa	5	aS[2*((a))(b)]	3	
aaabbab	5	a<(a)(b)>/<(ab)>	4	3*(a)2*(b)ab
aaabbba	5	3*(a)3*(b)a	3	
aaabbbb	4	3*(a)4*(b)	2	
aabaaaa	5	2*(a)b4*(a)	3	
aabaaab	5	S[(2*(a)b), (a)]	4	
aabaaba	5	aS[S[((a)),((b))]]	3	
aabaabb	5	a<(ab)>/<(a)(b)>	3	S[2*((a)),(b)]2*(b)
aababaa	5	S[2*((a))(b),(a)]	3	
aababab	4	a3*(ab)	4	
aababba	5	a<(a)(b)>/<(ba)>	4	2*(a)bS[(a)(b)]
aababbb	5	a<(a)>/<(b)(3*(b))>	4	2*(a)ba3*(b)
aabbaaa	5	S[2*((a))(b)]a	3	
aabbaab	5	S[2*((a))(b)]b	3	
aabbaba	5	a<(ab)>/<(b)(a)>	4	2*(a)2*(b)S[(a),(b)]
aabbabb	5	a2*(a2*(b))	3	2*(a)S[2*((b)),(a)]
aabbbaa	5	S[2*((a))(b),(b)]	3	
aabbbab	5	aS[(ab)(b)]	4	2*(a)3*(b)ab
aabbbba	4	<(a)(4*(b))>/<(a)>	3	2*(a)4*(b)a
aabbbbb	4	2*(a)5*(b)	2	
abaaaaa	4	ab5*(a)	3	
abaaaab	4	<(a)(4*(a))>/<(b)>	3	aS[(b)2*((a))]
abaaaba	5	S[S[((a)),((b))],(a)]	3	
abaaabb	5	a<(b)>/<(3*(a))(b)>	4	ab3*(a)2*(b)
abaabaa	5	S[S[((a)),((b))]]a	3	
abaabab	5	S[S[((a)),((b))]]b	3	
abaabba	5	<(ab)>/<(a)(b)>a	4	S[(a),(b)]S[(a)(b)]
abaabbb	5	<(a)>/<(b)(a)>3*(b)	4	
ababaaa	5	a<(b)>/<(a)(3*(a))>	4	aS[(b),(a)]3*(a)
ababaab	5	a<(b)(a)>/<(ab)>	4	S[(a),(b)]S[(b)(a)]
abababa	4	a3*(ba)	3	S[S[((a)),((b))],(b)]
abababb	4	3*(ab)b	4	
ababbaa	5	a<(ba)>/<(b)(a)>	4	S[(a),(b)]2*(b)2*(a)
ababbab	5	aS[S[((b)),((a))]]	3	
ababbba	5	aS[(ba)(b)]	4	S[(a),(b)]3*(b)a
ababbbb	4	<(a)>/<(b)(4*(b))>	3	S[(a),(b)]4*(b)
abbaaaa	5	a2*(b)4*(a)	3	
abbaaab	5	a<(b)(3*(a))>/<(b)>	4	a2*(b)3*(a)b
abbaaba	5	a<(b)(a)>/<(ba)>	4	S[(a)(b)]S[(a),(b)]
abbaabb	5	aS[2*((b))(a)]	3	
abbabaa	5	<(ab)>/<(b)(a)>a	4	S[(a)(b)]b2*(a)
abbabab	5	aS[(b)(ba)]	4	S[(a)(b)]S[(b),(a)]
abbabba	5	S[(a)2*((b)),(a)]	3	
abbabbb	5	S[(a)(b)]3*(b)	3	
abbbaaa	5	a3*(b)3*(a)	3	
abbbaab	5	<(a)>/<(3*(b))(a)>b	4	a3*(b)2*(a)b
abbbaba	5	a<(b)>/<(b)2*((a))>	4	a3*(b)S[(a),(b)]
abbbabb	5	S[(a2*(b)),(b)]	4	
abbbbaa	4	<(a)>/<(4*(b))(a)>	3	a4*(b)2*(a)
abbbbab	4	<(a)>/<(4*(b))(b)>	3	S[(a)2*((b))]b
abbbbba	4	S[(a),(5*(b))]	3	
abbbbbb	3	a6*(b)	2	