

A model for the temporal behaviour of cohesiveness among concepts

B. SHEKAR

Department of Electrical Engineering, Indian Institute of Science, Bangalore 560012, India.

Received on February 7, 1989; Revised on July 20, 1989.

Abstract

In this paper we examine the involvement of time in the cohesiveness among functional concepts. We propose a graph-theoretic model for the same and show how it can be embedded in a knowledge base. We use this model in defining cohesiveness-coefficient of a concept and make use of it in stating a sufficient and necessary condition for the absence of simultaneous execution of these concepts.

Key words: Temporal behaviour, conceptual cohesiveness, graph models, function concepts, knowledge base.

1. Introduction

Classifying objects into groups based on measurements made on the objects is something that human beings perform routinely in many walks of life. Recognition of patterns, which is one of the fundamental activities of the human mind is the heart of any classification mechanism. Thus, one can say that pattern recognition is the basic problem in the wider field of artificial intelligence.

Classifying objects into groups is a widely studied topic. A measure to characterize the similarity (or cohesiveness) between objects is a necessary criterion for the activity of classification. A comprehensive review of various distance and similarity measures is provided by Diday and Simon¹ and Anderberg². Conventional measures of similarity are 'context-free'. Gowda and Krishna³ defined the so-called 'mutual nearest neighbourhood' distance measure to capture the context. Even such measures lacked the ability to capture the 'gestalt property'⁴ of objects. In order to capture this aspect, Michalski⁴ proposed the following similarity measure: Similarity $(A, B) = \text{function}(A, B, E, C)$ where A and B are the two objects under consideration, and E and C the environment and a set of pre-defined concepts, respectively. The inability to capture gestalt property along with some solutions to this problem has also been discussed by Watanabe⁵. Though the definition provided by Michalski is fairly general, the concepts he used were more at the physical level, such as 'colour = red' and 'height = tall'. Shekar *et al*^{6,7} explored the possibility of utilization of a

knowledge base in classifying objects. This knowledge base comprises: definitions of concepts that are functional in nature and that are at one or more levels higher than the physical descriptions, and cohesiveness among these functions. The concepts in the knowledge base are functions (or activities) that are possible with the help of objects that possess the necessary physical descriptions.

In this paper, we look at the behaviour of this cohesiveness with respect to time. More specifically, we look at the behaviour with respect to the temporal sequencing of concepts that constitute a parent concept in an N-tree⁶. We evolve a graph-theoretic model as a representation mechanism to reflect this temporal behaviour. The graph-theoretic terms that we use are defined by Deo⁸ with examples. Further, we define 'cohesiveness coefficient' and use the same in stating a necessary and sufficient condition for the absence (or presence) of mutual exclusion in the execution of parent concepts. It may be observed that 'time' is not an explicitly stated parameter in the definition of similarity measure defined by Michalski. Also, it does not feature in the context of functional cohesiveness^{6,7}.

2. Temporal behaviour of cohesiveness among concepts

Functional cohesiveness among concepts (objects) has been defined and classified into three generically different component-functions:

- (i) The component $fg(a, b, C)$ between objects a and b assigns a value 0 to the set $\{a, b\}$ if they are different in view of concept C , or 1 if they are similar in view of C .
- (ii) The growth-dependent component $fgd_i(n, a, m, b, C)$ assigns a real number based on the growth of the i th cluster that has a C -conceptual correspondence between n objects of type a and m objects of type b with respect to concept C .
- (iii) The contextual component $fc(A_1, \dots, A_i, B, C)$ assigns a value to the composite cluster (A_1, \dots, A_i, B) in view of concept C , where A_1, \dots, A_i are subclusters in general, and B is a new cluster generated. It assigns a different value in the absence of cluster B .

For a detailed discussion of the above component-functions refer to Shekar *et al*⁶.

This definition of functional cohesiveness does not include the effect of time. In other words, it has been tacitly assumed that two concepts A and B always remain cohesive to each other in view of concept C , if once they are found to be cohesive. Though there are many instances of such cohesiveness, there are many other instances where this is not the case.

Consider the concept *writing*. Meaningful writing can be achieved with the help of objects that satisfy the concepts *marking*, *erasing*, and *cutting*. Thus, in view of the concept *cutting*, these three concepts display a high degree of cohesiveness. Here we associate the objects pencil, eraser, and blade with the concepts *marking*, *erasing*, and *cutting*, respectively (fig. 1). Now let us look at the activity of writing more closely. It is not difficult to observe that execution of the concept *marking* (such as marking some alphabetic symbol with a pencil) excludes the execution of the other two concepts at the same time. This should not be mistaken as 'only one at a time'. In fact this is an instance of manual execution of a concept.

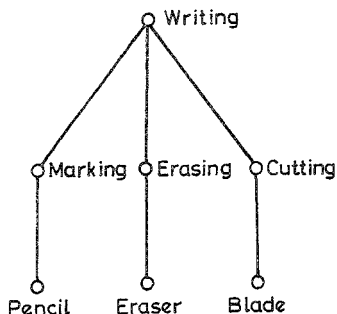


FIG. 1. An N tree.

There are many instances of automated execution that require more than one concept at the same time. Thus, we see that *marking* excludes *erasing* and *cutting*, and *cutting* excludes *erasing* alone because the object implying *marking* is involved with *cutting*. More generally, a constituent child concept of some concept may be needed only after the execution of some other concept. Consequently, the object implying this child concept is free to be used with/for any other concept. In other words, a concept *A* is cohesive to concept *B*, with respect to concept *C*, only if the execution of *B* has to simultaneously proceed along with the execution of *A*.

This analysis of cohesiveness will be especially helpful if the number of objects required by the concepts is more than the number available. Consider a situation where there is only one blade, and concept *cutting* is involved in two higher level concepts, namely, *shaving* and *writing*. Functional cohesiveness indicates that *writing* and *shaving* cannot be performed at the same time. Though this is true from a non-temporal point of view, knowledge of the temporal behaviour of the cohesiveness of constituent concepts can help in generating meaningful plans for the parent concepts. One remark is in order here. It is only the temporal dependency of concepts that we are investigating here, and not the time units involved in the execution of concepts. Consider the same writing example. All that we state is that the object (objects) implying *erasing* will be free when *marking* is being executed. Consequently, it can be used along with some other concept or all by itself. We never state that *marking* will be executed for some *t* units of time, and hence the involved objects will be free after *t* units of time. In fact this can never be stated in general except for a certain class of instances. One such instance is the winding of a mechanical wall-clock. The concept *winding* is executed at regular intervals of time and usually not after the clock has stopped.

Thus, the need to build this temporal information about concepts inside the cohesion forest⁶ is obvious. Cohesion forests have been discussed in detail by Shekar *et al*⁶. We build this temporal information by embedding a directed graph structure that we call as temporal graph (TG) in the cohesion forest. We define the temporal graph for a concept *k* that has

been defined in the cohesion forest, with the help of graph-theoretic terms:

$$TG(k) = G_1 \cup G_2$$

where

$$G_1 \subseteq V \times N \quad \text{and} \quad G_2 \subseteq (k \times N) \cup (N \times k)$$

where

$$V = \{v | (k, v) \in N\text{-tree}\} \quad \text{and} \quad N \text{ is the set of natural numbers.}$$

A few comments on the above definition are in order:

1. The temporal graph of any concept is made up of two generically different sub-graphs: G_1 that specifies the temporal sequencing of the constituent concepts of k , and G_2 that specifies the commencement and termination of the execution of k .
2. G_1 is specified as a relation on the cartesian product between the children of k (V) and the set of natural numbers (N). Natural numbers are used to specify the temporal sequencing of the child-concepts specified in V . Observe that G_1 may have more than one concept associated with the same natural number. This signifies parallel execution of more than one concept. Also observe that G_1 can never be an empty set, though the definition allows it. This is because any concept specified by the property descriptors at the leaf-level of an N -tree has a commencement and a termination of execution.
3. G_2 is specified as a relation on the union of two cartesian products. The product $k \times N$ specifies the commencement of k . This product specifies the edges that connect the node representing k to different time slots specified by N . These edges correspond to the different ways of commencing the execution of k . The product $N \times k$ specifies the different ways k can terminate. It may be noted that, though elements of G_2 are not linked to any element of V directly, the link is established through G_1 . Here again, G_2 can never be empty in practice.
4. It is worth noting that a temporal graph of a concept involves only those concepts that are children to it. These child-concepts themselves can have children. Thus, this temporal modelling of cohesiveness can span from the macro-level involving high-level concepts to the micro-level involving the objects specified by the instantiated physical descriptors.

For the sake of convenience we denote an element of G_1 as an edge $(n_1, C, n_1 + 1)$ instead of (C, n) where $n, n_1 \in N$. n_1 and $n_1 + 1$ signify the beginning and ending of the time slot specified by n .

Any edge in this temporal graph represents a concept in the cohesion forest, the only exception being the edges coming out of the parent concept and the ones going into the parent concept. Consider the graph given in fig. 2. Here, the tree rooted at C has been decomposed into constituent concepts A and B . These constituent concepts have been further decomposed into their respective objects described by property lists P_A and P_B . The edges of the N -tree are represented as broken lines and the edges of TG are represented as solid lines. Directed edge $(1, 2)$ passes through concept A , and directed edge $(2, 3)$ passes through concept B . Edges $(C, 1)$ and $(3, C)$ represent the fact that this TG corresponds to the parent concept C . It should be noted that instead of such a pair, if we have an edge $(3, 1)$ passing through C , then it will be impossible to identify which concept is the child and which is the parent. Of course, one can refer to the corresponding N -tree and get this

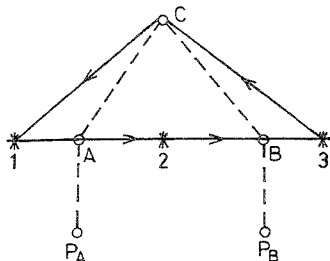


FIG. 2. A temporal graph.

information. This also will not be straight forward as each one of the concepts is a parent of some child node(s) in general.

Thus, any concept k_i in the forest, has a TG associated with it which we call $TG(k_i)$. The N -tree rooted at k_i is connected to $TG(k_i)$ only through k_i . More specifically, the connection occurs through an edge going out of k_i and through another coming into k_i . Every child concept at one level below the parent is a member of $TG(k_i)$. With these basic definitions, let us try to construct some TGs for the concepts *shaving*, *writing*, and *bathing*, and make some elementary observations on them.

Consider the N -tree of shaving given in fig. 3. This activity (*Sh*) has been decomposed into three constituent sub-activities, namely, lathering (*La*), looking (*Lo*), and cutting (*C*). Edge (*Sh*, 1) indicates that shaving commences with $TG(Sh)$ starting at vertex 1. Vertices 1 and 2 have parallel edges (1, *Lo*, 2) and (1, *La*, 2) between them. This indicates that the activities *Lo* and *La*, corresponding to looking and lathering, respectively, are simultaneously executed, and executed in the first time slot. These parallel edges are followed by another pair of parallel edges (2, *C*, 3) and (2, *Lo*, 3). This indicates that these two activities need to be executed in parallel after the previous pair of activities has been completed. Here, we observe that two concepts are simultaneously cohesive only if they are involved in at least two mutually parallel edges. We also observe that two concepts are sequentially cohesive if their corresponding edges are adjacent. In $TG(Sh)$, *La* and *C* are sequentially cohesive. So also are *La* and *Lo*. *Lo* and *La* are simultaneously cohesive. It is interesting to note that two concepts can be simultaneously and sequentially cohesive in the same TG. Henceforth, for ease of graphical representation, we do not depict the edges involving a concept if this concept is simultaneously cohesive with every other concept in the TG. *Lo* is an example of one such concept.

Consider the N -tree for the concept *writing* given in fig. 4. This concept may be decomposed into three constituent concepts *cutting*, *marking*, and *erasing* represented by labels *C*, *M*, and *E*, respectively. The two parallel edges (1, *C*, 2) and (1, *M*, 2) indicate that the marking object, namely a pencil has to be sharpened with the help of a cutting object, such

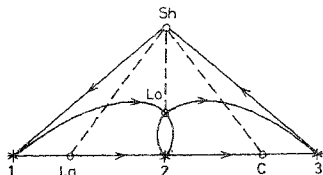


FIG. 3. TG (Sh).

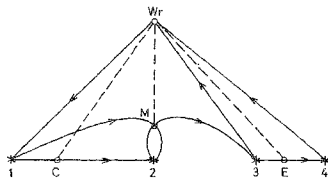


FIG. 4. TG (Wr).

as a blade. Subsequent to that, the marking process is the logical next step. This is represented by edge $(2, M, 3)$. The process of writing may be completed here, or it may be completed after the usage of an eraser. Edge $(3, Wr)$ represents the former and edge $(4, Wr)$ represents the latter. Edge $(3, E, 4)$ represents the time slot corresponding to the erasing activity. Thus we see that though *erasing* is present in the N -tree corresponding to concept *writing*, the activity of writing may get completed without the execution of *erasing*. Similarly, there may be more than one edge coming out of node Wr which indicates that there may be more than one way of commencing the execution of the concept corresponding to Wr . In general, the product of the in-degree and out-degree of any node k_i (with respect to $TG(k_i)$) in an N -tree reveals the number of ways in which k_i can be executed. Here it may also be observed that concepts C and E are not sequentially cohesive.

At this juncture, it is worthwhile to note the following to avoid any possible confusion in the interpretation of parallel edges. Parallel edges indicate the necessity of performing the corresponding activities simultaneously. As each activity that is represented as a concept (in an N -tree) can get decomposed into sub-concepts that correspond to sub-activities, and finally to physical descriptors that describe the relevant objects, any activity involves the relevant objects. Thus, parallel edges in essence signify the simultaneous execution of the corresponding concepts with the help of the relevant objects. If an activity requires two or more objects, then each of them must be performing a particular function that will consequently find a place in the cohesion forest. Let us assume that the function is not represented in the cohesion forest. We are only interested in any object from the functional angle. Thus, the parallel edges from the concept that is the parent of these physically specified objects that are in essence concepts (refer to Shekar *et al*⁶ for the recursive definition of concept) will involve these objects. As the object specification occurs at the leaf-level, the TG at that level will indicate the simultaneity at the micro-level which can augment the TGs at higher levels.

Consider the activity of bathing. A possible decomposition of this activity (Ba) into sub-concepts, namely, a *room* (Ro) for ensuring privacy; two units of *vessel* (Ve), one large one for holding a large quantity of water and a small one for transferring water from the large vessel on to the body; and *lathering* (La) for cleansing the body; is shown in fig. 5. As *wetting* (the term 'wetting' is used synonymously with the phrase 'to make the surface wet' for rhyme and brevity) and *vessel* are executed together to drench the body, we have parallel edges $(1, We, 2)$ and $(1, Ve, 2)$. Strictly speaking, edge (Ba, Ve) in the N -tree has to be

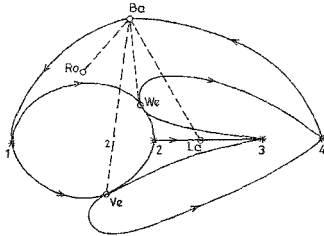


FIG. 5. TG (Ba).

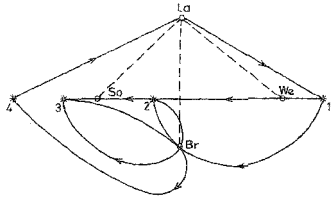


FIG. 6. TG (La).

replicated as there are two vessels involved. However, the need does not arise here as both the vessels are simultaneously active or simultaneously inactive. After drenching the body, we perform the activity of lathering. This is represented by edge (2, La, 3). Finally, we again drench the body. This is represented by edges (3, We, 4) and (3, Ve, 4). It may be observed that the concepts between vertices 1 and 2 get repeated between vertices 3 and 4.

As the concept *lathering* occurs in the bathing scenario and also in the shaving scenario, and it is a scenario in its own right, let us look at its conceptual decomposition. *Lathering* consists of concepts *soaping*, *brushing*, and *wetting* (we use the term 'soaping' synonymous to the phrase 'applying soap to the surface' for rhyme and brevity). This *N*-tree along with the associated TG is given in fig. 6. Lathering commences with the wetting of the brush. Consequently we have parallel edges (1, We, 2) and (1, Br, 2). Following this, we generate lather with the help of the wet brush and a soap. This is represented as (3, Br, 4). Thus we see that node *Br* is involved in every time slot, namely (1, 2), (2, 3), and (3, 4). If we remove the edges involving this node, then we will be left with a directed edge from vertex 3 to vertex 4 without involving any concept. Hence we do not exclude it.

Thus, from the above discussion and the definition of TG, we observe that no TG can contain a cycle if the entry and exit edges are removed. This results from the fact that time always moves forward and never backward. Even when the same concept is involved at a later point in time, the corresponding edge passes through that concept but the edge is not routed back. For example, in fig. 5, the edges between vertices 3 and 4 are identical to the edges between vertices 1 and 2. Still we do not create an edge (2, La, 4). This is because every vertex in the TG represents a time instance and an ascending sequence of edges represents increase in time. One other observation that can be made is regarding the two types of sequential cohesiveness among concepts, namely, bi-directional cohesiveness and uni-directional cohesiveness (fig. 4). One cannot execute the concept of erasing without having executed the concept of marking. In other words, *erasing* is cohesive to *marking*. On the contrary, *marking* is not cohesive to *erasing* because *erasing* is not a necessary preceding activity for *marking*. There are TGs where it is true the other way round also. Consider the TG given in fig. 5. While it is true that the concept represented by *La* requires the execution of the concept represented by *We* to precede it, it is not true that *wetting* requires the

execution of *lathering*. However, in this TG the execution of *lathering* must be succeeded by the execution of *wetting*. Thus they are bi-directionally cohesive. Observe that cohesiveness is with respect to the scenario under consideration and not independent. This is the reason why *lathering* is dependent on *vessel* also.

Sequential cohesiveness can be at various levels. In the shaving scenario, concept *wetting* that is a constituent concept of *lathering* is two time slots away from concept *cutting*. Hence, we can say that concept *cutting* is 3-cohesive to *wetting*. Observe that simultaneous cohesiveness can be addressed as 0-cohesiveness. Though this definition of cohesiveness may seem unimportant in a scenario, it assumes importance between scenarios. Consider the TGs given in figs 3 and 4. In TG(*Wr*), we know that *cutting* is 0-cohesive, whereas in TG(*Sh*) *cutting* is 1-cohesive. We can straight away say that both of them can be executed simultaneously with the presence of just one blade. This level-of-cohesiveness information has to be defined by taking the entire root-to-leaf decomposition of an *N*-tree into consideration. Thus, TG(*La*) should also be considered while assigning cohesiveness levels. This can also play a vital role in staggering (from the temporal view point) the execution of concepts.

It is not difficult to observe that the relative definition given for level-of-cohesiveness can be made absolute by selecting the base-level concept, *i.e.*, the concept with respect to which level is being computed, as the one that occurs earliest in the TG. In such a case, we call it as 'cohesiveness coefficient'. We give below the definition of this coefficient, denoted by $\text{coef}(C, \text{TG}(k_i))$, for a concept *C* in a temporal graph TG(k_i) containing *C*. The definition makes use of graph-theoretic terms involved in directed graphs⁸.

$$\text{Let } G(k_i) = \text{TG}(k_i) - \{(k_i, p), (q, k_i) \mid p, q \in \text{TG}(k_i)\}$$

$$\text{Coef}(C, \text{TG}(k_i)) = 0 \quad \text{whenever edge } (w, C, x) \in G(k_i) \quad \text{and} \quad d^-(w) = 0;$$

$$\text{Coef}(C, \text{TG}(k_i)) = d(w, y) \quad \text{whenever edge } (y, C, z) \in G(k_i).$$

Here, ' d^- ' is the in-degree of the specified node and ' d ' is the distance between the specified nodes.

As an example, consider the TG given in fig. 3. To determine a base-level concept, we identify vertex *w* whose in-degree is 0 in $\text{TG}(k_i) - \{(k_i, p), (q, k_i)\}$ for all *p* and *q*. Here, this happens to be node 1. Now any concept *Z* such that $(w, Z, s) \in \text{TG}(k_i)$ can be chosen as the base-level concept. In this case, *Lo* and *La* happen to be base-level concepts. Thus they have a cohesiveness-coefficient of 0. Hence $\text{Coef}(C, \text{TG}(\textit{Sh}))$ has a value of 1. This coefficient is specially useful in stating the following necessary and sufficient condition for the absence of simultaneous execution of concepts. It is stated using the first-order predicate calculus.

$$(\forall k_i)(\forall k_j)((\exists C)(C \in \text{TG}(k_i) \wedge C \in \text{TG}(k_j)) \wedge$$

$$\text{coef}(C, \text{TG}(k_i)) = \text{coef}(C, \text{TG}(k_j)) \Leftrightarrow \neg \text{simul}(k_i, k_j).$$

It should be noted that this condition is based on the assumption that the objects present are just enough for the execution of k_i and k_j individually, and not simultaneously. The forward implication states a sufficient condition for the existence of non-simultaneous

execution of k_i and k_j . In other words, if there exists a concept C that is a member of $TG(k_i)$ and $TG(k_j)$, and its cohesiveness coefficients are the same in both the TGs, then it is not possible to execute k_i and k_j at the same time. If there does not exist any concept that satisfies the antecedent, then it is possible to execute both the concepts at the same time.

It is important to note that the above condition does not reflect a transitive behaviour among concepts. Consider concepts E , F , and G . There may not be any concept with the same cohesiveness coefficient that is common to E and F . Consequently E and F can be executed simultaneously. The same may be true with F and G also. However, this may not be true with E and G . Observe that the above condition overrides the condition given by Shekar *et al*⁶. There, the membership of C in the two N -trees corresponding to k_i and k_j was sufficient to introduce mutual exclusion. Here, we add more power to it by stating that it is the cohesiveness coefficients that determine mutual exclusion and not mere membership alone.

3. Conclusions

In this paper, we try to abstract the temporal nature present in the cohesiveness among concepts. We propose a graph-theoretic model called 'temporal graph' which reflects temporal sequencing of concepts and embed this graph in the cohesion forest. This in essence is an acyclic-directed graph connecting the concepts that are the children of the parent concept representing the root of the N -tree. We elucidate this model with the help of a few examples and make some observations on its characteristics.

This directed graph helps us to define two types of cohesiveness among concepts, namely, parallel cohesiveness and sequential cohesiveness. We go further to generalize them. This results in the definition of 'cohesiveness coefficient' of a concept in a temporal graph. This attribute helps us to look at simultaneous execution of concepts defined in the cohesion forest more clearly. We identify a necessary and sufficient condition for the absence of simultaneous execution in a given pair of concepts. This condition is more powerful than the condition stated in an earlier paper⁶. However, for extra power, we need extra information in the knowledge base. This should be resident in the knowledge base as cohesiveness coefficients. Here, we make an assumption that a directed edge represents one time unit. In the most general sense this may not be true. It is not difficult to refine the model further to include that also. We have not done it here as it does not contribute to the furthering of conceptual clarity as regards execution of concepts.

References

1. DIDAY, E. AND SIMON, J. C. Clustering analysis, in *Digital pattern recognition*, Communication and Cybernetics Series, Vol. 10, (ed.), K. S. Fu, 1976, pp. 47-94, Springer-Verlag.
2. ANDERBERG, M. R. *Cluster analysis for applications*, 1973, Academic Press.

3. GOWDA, K. C. AND KRISHNA, G. Disaggregative clustering using the concept of mutual nearest neighbourhood, *IEEE Trans.*, 1978, **SMC-8**, 888-895.
4. MICHALSKI, R. S. Knowledge acquisition through conceptual clustering: A theoretical framework and an algorithm for partitioning data into conjunctive concepts, *Inter. J. Policy Analysis Inf. Systems*, 1980, **4**, 219-244.
5. WATANABE, S. *Knowing and guessing: A quantitative study of inference and information*, 1969, Wiley.
6. SHEKAR, B., MURTY, M. N. AND KRISHNA, G. A knowledge-based clustering scheme, *Pattern Recognition Letters*, 1987, **5**, 253-259.
7. SHEKAR, B., MURTY, M. N. AND KRISHNA, G. Pattern clustering: An artificial intelligence approach, *Proc. 10th Int. Jt Conf. on Artificial Intelligence*, Milano, Italy, pp. 214-216, August 1987.
8. DEO, N. *Graph theory with applications to engineering and computer science*, 1970, Prentice-Hall.