INTERNATIONAL DESIGN CONFERENCE - DESIGN 2008
Dubrovnik - Croatia, May 19 - 22, 2008.

DESIGN 2008

HIERARCHICAL DECOMPOSITIONS FOR COMPLEX PRODUCT REPRESENTATION

O.O. Ariyo, C.M. Eckert and P.J. Clarkson

Keywords: hierarchy formation, product breakdown, hierarchical structuring, product representation

1. Introduction

In many published design texts and standards, for example [INCOSE 2004, pp. 10-14] it is often advocated that hierarchical descriptions should be structured into successive levels of systems, assemblies, components and so. An underlying principle of this viewpoint is the assumption that products are readily decomposable into hierarchical descriptions. More importantly, there is a presumption that the resulting decomposition, irrespective of its characteristics, will satisfy its intended purpose. Yet, attaining a useful hierarchical product breakdown in practice can at times be challenging. Product representation is pivotal to creating models which aid reasoning about a design. Complex products such as jet engines typically have thousands of unique components. The use of single-levelled decompositions for product representation could lead to ineffective models which are either too difficult to manage due to their size or that are too abstract to apply to practical design problems. Hierarchical structured decompositions offer the capacity to manage large models effectively [Jarratt 2004]. However, in practice, sub-units within a hierarchy are defined purely in an arbitrary manner. This approach leaves a chance of creating product decompositions that fail to fulfil the objective of complexity management which had lead to adopting hierarchical grouping of components in the first place.

This paper suggests precautionary steps to avoid poor hierarchical product breakdowns by developing rules and guidelines to aid the decomposition process. These guidelines are based on a critical review of published literature and insights from an exercise carried out to investigate the different ways in which a motorcycle may be decomposed. This research was carried out for the purpose of creating hierarchical structured product representations that support reasoning about the potential effects of design change on components and systems. However, it is important to realise that the theories on hierarchies presented have much broader implications beyond change management since hierarchy concepts are at the core of many techniques and processes used for managing complexity.

2. Hierarchy theory

A review of literature on network structures was carried out (1) to clarify the meaning of the term hierarchy (2) to identify what types of hierarchy may be used for product representation and (3) to establish the defining characteristics of a hierarchical structure.

2.1 What is a hierarchy?

The term hierarchy implies a mode of structuring in which some aspects of a network are ranked higher than others. In some types of hierarchical structures, an additional condition of "*partitioning*" of nodes is imposed on the final structure. This means each lower level component is restricted to having a single "boss." This type of hierarchy is commonly referred to as a rooted "tree" (Figure 1a). In many design texts, hierarchies are used synonymously with *trees*, although not every hierarchy relation can be described as a tree. In fact, *partitioning* is not a condition for hierarchy formation [Johnson 2005]. Hierarchies can also be structured in levels as shown in (Figure 1b). Unless in cases explicitly stated, the term hierarchy as used in this paper encompasses both *trees* and *multi-levelled* descriptions.

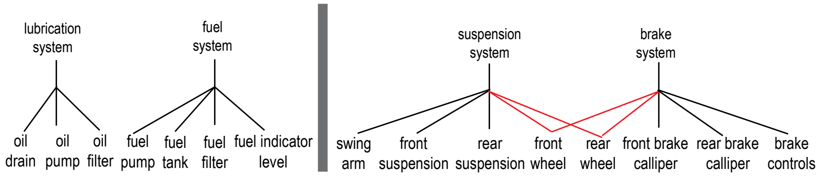


Figure 1. (a) Partitioned and (b) non-partitioned hierarchy

2.2 Types of hierarchy

The criteria used when creating a hierarchy is referred to as the abstraction principle, abstraction method or abstraction mechanism. There are three main types of abstraction mechanisms [Yoo and Bieber 2000].

- **Classification/instantiation hierarchies:** In such hierarchies, an abstract unit is defined based on properties that do not change in time. Sub-units in such a hierarchy are instances of an object. For example, a choice of a red or blue car forms different instances of the same object (i.e. a car). Lower levels of this hierarchy may consist of different shades of the red car.
- **Generalisation/specialisation hierarchies:** This is commonly known in the AI community as the 'is a' relation. Specialisation describes sub-units in a class, which share the same properties as others within that category but nonetheless have their own defining properties. This relation leads to a hierarchy of 'types.' An illustration of a generalisation/specialisation hierarchy of motorcycles is shown in Figure 2.

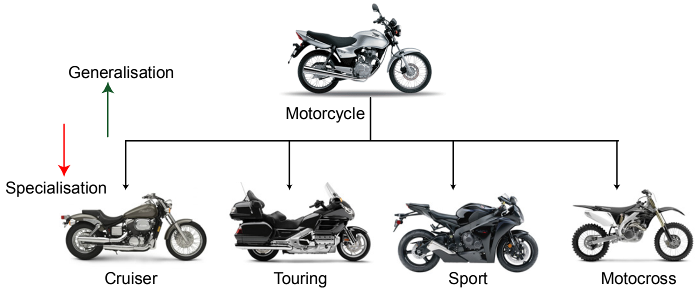


Figure 2. Generalisation /specialisation hierarchy

- **Part/whole hierarchies:** Commonly known in the AI community as 'part of' relation. It describes relations of parts that make up a whole. For example a car may consist of engine, body, chassis, and wheels. The engine is composed of lower level systems, which are then broken into more components.

The part/whole type of abstraction mechanism enables composition and decomposition of elements within a product. As a result, this type of abstraction mechanism is favourable for creating hierarchical structured descriptions of a product.

2.3 Conditions for hierarchy formation

Not all types of relationships between parts and wholes are hierarchies. This has been illustrated in numerous texts [for example, Johnson 2005] with the Simpson's finger conundrum which is explained as follows: If Simpson's finger is "part of" Simpson and if Simpson is a "part of" the Philosophy Department, is Simpson's finger a part of the Philosophy Department? In order to avoid being caught by such a conundrum, it is important that a part/whole hierarchy satisfy three primary conditions:

- Irreflexivity: an object cannot be part of itself directly or indirectly.
- Antisymmetry: this characteristic implies that if an object 'A' has a part-whole relation to an object 'B' and if an object 'B' has a part-whole relation to an object 'A,' then both A and B must be the same object.
- Transitivity: this implies that if 'A' is a part of 'B' and 'B' is a part of 'C,' then 'A' must be a part of 'C.'

Transitivity is a key condition for hierarchy formation; hence non-transitive part-whole relations (such as the Simpson's finger example) are not hierarchies. However, satisfying the condition of transitivity does not in itself ensure creation of useful hierarchical structured descriptions of products.

3. Requirement for hierarchical product representation

There are three main requirements that must be satisfied when creating hierarchical product representations.

1. Tree structure: The relations between parts and wholes should be represented as a tree. It is difficult to represent the content of a multi-levelled hierarchy without the edges between nodes crossing [Feiner 1988]. A tree structured representation enables a clear representation of nodes within a hierarchy.
2. Manageable steepness: This requirement is essential to fulfilling the objective of managing large product models effectively. It is important that the steepness of successive hierarchy levels is not too sharp or too flat. While there is yet to be research to determine the ideal steepness-index (e.g. appropriate whole-to-part ratio) for hierarchical product decompositions, common sense tells us that a breakdown which takes the form of a "broad and shallow" structure (as illustrated in Figure 3a) may not support the representation of important sub-systems within a complex product. Similarly, an excessively narrow hierarchical structure (Figure 3b) may also be impractical for product representation, since they require a lot of effort to build in addition to being cumbersome to use and difficult to manage. Unless this requirement is satisfied, it is difficult to justify the need for hierarchical structuring to product representation.

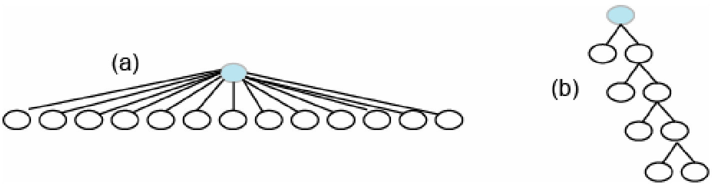


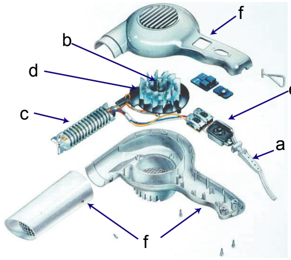
Figure 3. (a) Broad and shallow (b) Deep and narrow tree structures

3. Product breakdown completeness: Unlike the two previous requirements, this condition is induced by the background of change management against which this research is carried out. This need not be the case for hierarchical product breakdown intended for other purposes. The product breakdown is the reference-point against which inferences about the effects of design changes to components are made. Due to space limitations, a hairdryer model is used to show component connectivity (see Figure 4). It is important that the sum of the elements within a breakdown, considered when attending to a change query, constitute a relatively "complete" description of the product being assessed. The term "complete" in this case implies that the decomposition encompasses all the major assemblies and components within the product, system or module being investigated (as shown in Figure 4).

Having established the conditions and requirements for hierarchy formation, there are three main questions that arise. These are:

1. How can a tree-like product breakdown be applied to complex products?
2. How can the steepness of a hierarchy be adjusted to suitable levels?
3. How does one ensure a consistent degree of "completeness" for all levels in the hierarchy?

In order to answer these questions it was important to investigate the decomposability of a complex product.



Hairdryer	a	b	c	d	e	f
a power supply	-	x	x			
b motor	x	-		x	x	x
c heating unit	x		-	x	x	x
d fan	x	x	x	-		x
e control system			x	x	-	x
f casing	x	x	x	x	x	-

Product decomposition

Figure 4. Component connectivity within a hairdryer

4. Investigating product decomposability into hierarchies

As a way of assessing the factors that influence both tree-structuring and steepness in hierarchies, a breakdown of components within a motorcycle was carried out by the first author who is educated as a mechanical engineer to a graduate level. During this process, the challenges encountered in decomposing the motorcycle were noted. The decomposition guidelines provided at the end of this paper are based on the findings from this study.

4.1 Approach taken for assessing product decomposability

The hierarchy was formed using the part/whole abstraction mechanism. The decomposition of wholes into parts was transitive. It was assumed that a "part" may be categorised as belonging to one or more "wholes" i.e. multi-levelled hierarchies. The resulting product breakdown consisted of three levels of part descriptions, each having a different degree of granularity.

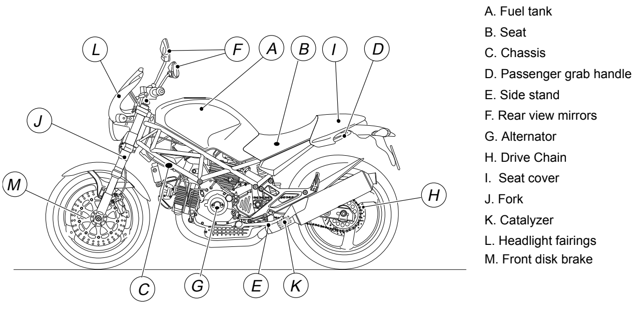


Figure 5. Decomposing a motorcycle into components

At the lowest (or finest) of the three decomposition levels, the description of the motorcycle consisted predominantly of single components such as *Pistons* and *Con-rods*. This level of decomposition consisted of 45 components; some of these are shown in Figure 5. Smaller product parts such as bolts, nuts, wires and cables were abstracted at the level of sub-assemblies. The middle level was abstracted at the level of sub-systems and sub-assemblies (e.g. *Lubrication system*). The choice of high-level descriptions consisted mainly of systems such as the *power and transmission system*. The final hierarchical breakdown consisted of 6, 14 and 45 elements on the respective levels. Naming conventions for product parts were based on descriptions available in the parts catalogue, the owner's manual and the service manual of the particular product.

4.2 Factors that influence tree structuring during hierarchy formation

From the study, there are two main properties of products that affect how "parts" are classified into "wholes" within a hierarchy structure. These are:

- Separability: this is the potential for a part to be removed from a whole. Some parts are inseparable from their whole. For example, the fuel tank of some aircrafts cannot be physically separated from the wing. This type of function sharing is more in common products with a highly integral architecture as opposed to those structured into modules.
- Shareability: this is the ability for a part to belong to two or more wholes. This tends to arise when a component performs a function in more than one system. Parts that are not shared are exclusive to wholes. A part in a complex product is likely to be shared amongst many systems.

Both of these issues arise due to the architecture of products. The main implication is that, depending on the strictness of the criteria used to group "parts" into "wholes," it is difficult to decompose a product into a tree structure.

4.3 Factors that influence "steepness" during hierarchy formation

The assessment of decomposability of a motorcycle revealed four primary factors that caused variations in the steepness of hierarchical product breakdown. Each of these factors is discussed in turn.

4.3.1 Abstraction levels and granularity selection

A choice over the level of granularity at which a sub-system is abstracted is partly objective and partly subjective. Objectivity arose from architectural constraints. For example, the piston of a motorcycle is a part of its engine. Variations in hierarchical structures arise due to subjective views of the level of granularity at which a sub-system could be abstracted. The level of granularity chosen for abstracting parts and wholes has implications on the number of nodes within the product model and consequently on the steepness of the resulting hierarchy, including its flatness and depth.

4.3.2 Abstraction principle adherence

The steepness of a hierarchy is also influenced by the ability to stick to one abstraction mechanism. When hierarchically decomposing the motorcycle, it is difficult to adhere to a single abstraction principle of structuring based on parts and wholes. There were instances where it was necessary to switch between abstraction mechanisms. For example, multiple instances of one type of component such as the *Temperature sensors*, *Thermal sensors*, etc., were better grouped as "*Sensors*" even though these components were not physically "part of" the same assembly. This manner of hierarchical structuring is a generalisation/specialisation abstraction mechanism.

4.3.3 Preferential perspectives in hierarchical grouping

According to Balcerak and Dale [1992], a particular set of relations forms a "perspective." A single set of components may be related in many ways. Examples include systems, assemblies, modules, or even information flow. Due to space limitations, an example of a hairdryer is again used to illustrate these differences in hierarchical decompositions (see Figure 6). Depending on the perspective considered when decomposing a product, the steepness of the resulting hierarchy will vary. However, it is important to note that the choice of a specific perspective is dependent on the goal for which the hierarchy is built and perceptions about what can be achieved decomposing with a particular structuring perspective.

4.3.4 Decomposition convention - Top-down vs. Bottom-up

Nodes on hierarchies are two-faced entities. They act as wholes in relations to lower-level parts and as parts for wholes on higher levels. However, the process for hierarchical grouping is neither strictly a top-down or bottom-up decomposition method. The top-down approach allows for products to be decomposed in instalments. More and more detail can be added on each layer of the hierarchy. A bottom-up approach requires that lower-level parts are listed first after which such parts are aggregated into wholes at higher levels. Regardless of which of these approaches is taken, the choice of component granularity on each successive level of the hierarchy is partly subjective thereby creating a potential for variation in the steepness of the resulting hierarchy.

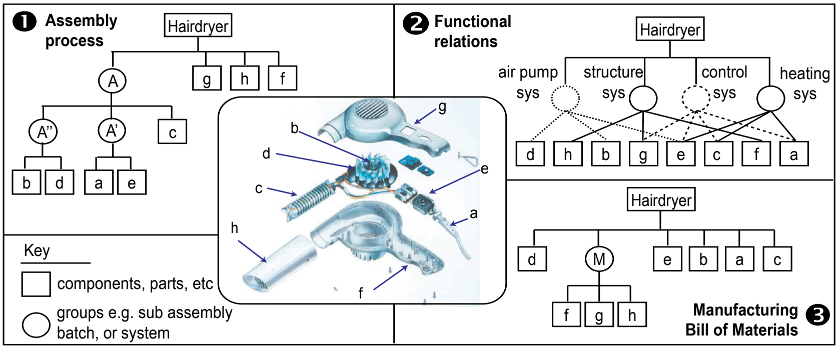


Figure 6. Multiple clustering perspectives for complex products

4.4 Factors that affect product breakdown completeness

Attaining a relatively complete breakdown of components within a product can be challenging. For example, the earlier mentioned issue of "separability" can sometimes lead to a failure to account for non-physical components or duplication of entities already listed. Even in cases where a full description can be achieved at a particular level of granularity, obtaining such complete descriptions at other levels of granularity can be challenging. Two main issues related to product description completion highlighted during the study are:

- **Naming conventions:** Grouping of components into "chunks" sometimes lead to wholes that do not have a name. A lack of a name may lead to communication ambiguity and misguided interpretation of concepts. When component names are more specific, "more is inherently understood about the artefact" [Greer et al., 2003]. A shared understanding of design technologies (through the use of names) helps to facilitate information transfer.
- **Maintenance of hierarchies:** The problem of completeness in product breakdown extends beyond the attainment of hierarchical product descriptions. Modifications to products sometime necessitate an update to product representations. Depending on the properties of a hierarchy such as its flatness, its depth, the number of nodes and the degree of component sharing between wholes, it is not always easy to update the hierarchy to reflect the changes without introducing other forms of inaccuracies into the model.

As it was the case with managing the steepness of hierarchies and obtaining tree-structured decompositions, it is important to develop guidelines, which aid in satisfying the requirement of describing a product completely.

5. Hierarchical structuring of product descriptions

As stated earlier, hierarchical product breakdown intended for change assessment needs to satisfy three requirements (1) a tree structure, (2) manageable steepness and (3) a relatively complete decomposition of the product being assessed. The study described in the previous section brought into light various factors that may affect the possibility of satisfying each of these requirements. In this section, techniques for attaining product breakdowns that meet all of these requirements are discussed.

5.1 Tree structure formation

One way of attaining a tree-structure during hierarchical decompositions is through the introduction of a partitioning condition. This enables grouping parts as belonging to a single whole. For example, a hierarchy may be partitioned into groups for which a component is a primary member. However, depending on the purpose for which the decomposition is intended, it is important to realise that enforcing such a partitioning condition may require leaving out some information dependencies between parts and wholes. For instance, the piston of an engine may be grouped under the combustion system for which it is a primary member; thereby failing to include information about its dependency with other secondary groups such as the lubrication system. Irrespective of the choice of partitioning condition adopted, it is important that the resulting hierarchy satisfies the condition of irreflexivity.

5.2 Hierarchical breakdown steepness modification

With regards to satisfying the second requirement i.e. managing the steepness of the hierarchical product breakdown, the steepness of a hierarchy depends on the ratio by which "parts" are allocated to "wholes." Although it is important that successive levels of component descriptions are chosen at a suitable abstraction level, it may not be sufficient to ensure tree structures have the desired steepness. Two more techniques for modifying the steepness of hierarchies are as follows:

- **Change in abstraction principle**: In a situation where a single component appears on more than one occasion in a product breakdown, the steepness of a hierarchy can be modified changing from a part/whole to a generalisation/specialisation abstraction principle. This non-adherence to a single abstraction mechanism enables modification of the product breakdown structure. However, it is important that the resulting decomposition still satisfies the condition of transitivity for such a breakdown to be classified as a hierarchy.
- **Hierarchical grouping:** The manner in which components are decomposed into tree-structures can also help to create hierarchies with a favourable steepness. Depending on the degree of strictness to which the partitioning condition is applied, a tree structure may also be used to vary the structure of a hierarchy as illustrated in Figure 7.

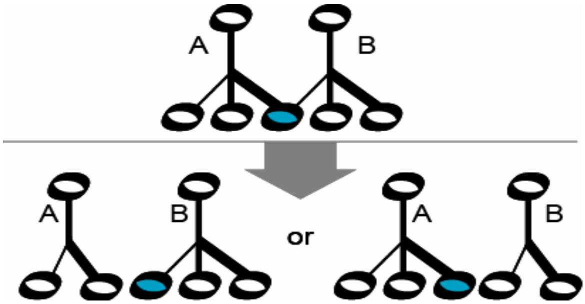


Figure 7. Variation in hierarchical decompositions

5.3 Product breakdown completeness

To ensure that elements within a breakdown constitute a relatively complete description of a product, it is important that successive levels of hierarchies are contrasted against each other. The descriptions at a system level must accommodate all lower level components and must satisfy the condition of antisymmetry. There are situations where a "part" does not belong to any "whole." In such cases, a lower-level component may be repeated at a high level to ensure completeness.

6. Conclusion and Further work

Hierarchical structuring of product descriptions has been presented as a technique for representing complex products. However, there is more than one way of decomposing products hierarchically. *Guidelines* for attaining meaningful hierarchical product decomposition are summarised in Table 1.
These guidelines have been applied to decomposing the High Pressure (HP) rotor of a jet engine in an industrial setting. By adopting rules and *recommendations* outlined in Table 1, the hierarchy structure created was useful for managing the scale of the product breakdown within the model. It also enabled assessment of extensive number of change queries beyond extents achievable with single-levelled product breakdown. Further work is required to enable refinement of these guidelines provided. An investigation into the appropriate limits for whole/part ratio will further assist model builders in structuring product descriptions into hierarchies.

Table 1. Guidelines for hierarchical decomposition of complex products

	Requirements	Rules	Recommendations
1	**Tree structure** – Parts and wholes should be structured into a tree to enable clear representations nodes within a hierarchy.	Tree structured representations of a product breakdown must be satisfy the condition of **irreflexivity**.	In a situation where a "part" is member to more than a single whole, a partitioning condition should be used to modify the hierarchy into a tree structure. For example, each "part" may be assigned to a single "whole" for which it is a primary member.
2	**Manageable steepness** - A sensible degree of steepness must be attained in order to manage the complexity of the resulting hierarchical structure	In attempt to manage hierarchy steepness, it is important to note that the product breakdown must be **transitive**.	Care must be taken when deciding on the level of granularity for successive hierarchy levels to avoid it being too broad or too narrow.
			Hierarchical structuring should not be limited by the need to adhere to an abstraction principle.
3	**Product breakdown completeness** - The description at each level of the hierarchy should be a relatively complete representation of the same entity.	Successive levels of product description must satisfy the condition of **antisymmetry**.	*Inseparable* components (e.g. an aircraft wing which is also a fuel tank) may be represented using distinct entities, provided it can be noted as having the same physical embodiment.
			Successive levels of product description should be contrasted against each other to ensure completeness.

References

Balcerak, K. J. and Dale, B. G., "Engineering Change Administration: the Key Issues", Computer-Integrated Manufacturing Systems, Vol. 5, No. 2, 1992, pp. 125-132.

Feiner, S., "Seeing the forest for the trees: Hierarchical display of hypertext structure", In Proceedings of the Conference on Office Information Systems., 1988.

Greer, J. L., Stock, M. E., Stone, R. B. and Wood, K. L., " Enumerating the Component Space: First Steps Toward a Design Naming Convention for Mechanical Parts", ASME Design Engineering Technical Conferences, Chicago, IL, 2003.

INCOSE, "Systems Engineering Handbook: A "What to" Guide for all SE Practitioners", International Council on Systems Engineering., 2004.

Jarratt, T. A. W., "A Model-Based Approach to Support the Management of Engineering Change", PhD Thesis, CUED, Cambridge University

Johnson, J., "Multidimensional Multilevel Networks in the Science of Complex Systems", Proceedings of the ECCS 2005 Satellite Workshop: Embracing Complexity in Design, J. Johnson, T. Zamenopoulos and K. Alexiou, The Open University, Milton Keynes, 2005.

Yoo, J. and Bieber, M., "Towards a Relationship Navigation Analysis", Proceedings of 33rd Hawaii International Conference on System Science, HICSS, IEEE Press., Washington, D.C., 2000.

Owolabi Ariyo
PhD Research Student
University of Cambridge, Engineering Design Centre
Trumpington Street, Cambridge, CB2 1PZ, United Kingdom
Tel.: +44 1223 332758
Fax.: +44 1223 766963
Email: ooa24@cam.ac.uk
URL: http://www-edc.eng.cam.ac.uk/people/ooa24.html