A new architectural metric and its visualisation to support incremental re-architecting of large legacy systems

Reinder J. Bril and André Postma
Philips Research Laboratories Eindhoven (PRLE), Prof. Holstlaan 4 (Bldg. WL), NL-5656 AA Eindhoven, The Netherlands
{reinder.bril, andre.postma}@philips.com

Summary

In this paper, we present a new architectural metric, termed dominating ratio, together with an appropriate visualisation. The dominating ratio provides support for re-architecting in situations where existing metrics, like coupling and cohesion, fall short. A tabular representation with browsing facilities enables a systematic investigation of the software architecture for incremental re-architecting of large legacy systems. Such a means for visualisation is yet uncommon for quantitative information and novel for an architectural metric. Initial experience with the dominating ratio and its visualisation is described briefly.

Keywords

Software architecture, software architecture re-engineering, re-architecting, architectural metrics, architecture visualisation, architectural evolution, architecture recovery.

1. Introduction

Software architecture plays a vital role in the development of large software systems. For the purposes of maintenance, an explicit description of the software architecture of a system supports comprehension of it, amongst other things. Many large legacy systems do not have an up-to-date documented software architecture, however. Within Philips, we therefore gained experience with software architecture recovery (or reverse architecting) (see [7]). We further witness that software architecture control, securing the state-of-affairs (preventing further deterioration) and guaranteeing the quality of enhancements made during maintenance activities, is moving from state-of-research towards state-of-the-art (see [2]). Software architecture re-engineering (or software re-architecting), which is typically required for perfective maintenance activities, software architectural improvements, or as part of architectural evolution, is still in its infancy, however. Publications in this area are limited to general approaches and descriptions of first experiences with toy examples (see, e.g., [3], [6], and [8]). Clearly, there is a need for incremental software re-architecting support. Although automatic re-architecting is felt to be impossible in general, metrics (and their visualisation) can support software re-architecting by providing insight in the quality of a software architecture, and revealing the impact of changes to the architecture. The support provided by
coupling and cohesion, the two classical examples of architectural metrics, is too limited, however, as will be shown in this paper.

This paper introduces a new metric, called *dominating ratio*, which provides a unifying basis for coupling and cohesion, and provides support for re-architecting in situations where coupling and cohesion fall short. The metric has been inspired by [7] and is usable for impact analysis purposes in contexts as described in [6]. Furthermore, a novel way for visualising this kind of metric is proposed, facilitating a systematic investigation of the software architecture. Initial experience with the metric and its visualisation is described briefly.

# 2. Dominating Ratio

Coupling (or inter-connectivity of composites, i.e. sets of entities) and cohesion (intra-connectivity) are two well-known (architectural) metrics, used for re-clustering [8], among other purposes. Coupling ($C(x,y)$) and cohesion ($C(x)$) are the fraction of the maximum number of directed dependencies between disjoint composites $(x,y)$ and within a composite $(x)$, respectively.

Unfortunately, as will be illustrated in this section, coupling and cohesion do not provide adequate support for incremental re-architecting. Therefore, we define a new metric, termed *dominating ratio* ($DR(x,y)$). For identical composites (i.e. $DR(x,x)$), the metric is similar to cohesion. For disjoint composites $(x,y)$, the metric is the fraction of the maximum number of directed dependencies from the first $(x)$ to the second composite $(y)$. For nested composites $(x,y)$, the metric is defined similarly as for disjoint composites by taking the constituent and its complement within the constituting entity (effectively excluding intra-constituent dependencies). For completeness, a formal definition of coupling, cohesion and dominating ratio is given in the appendix.

![Figure I: Strictly layered system S and Subsystems B and C](image)

Legend:
- **A**: entity
- **→**: dependency (uses relation)
The distinguishing properties of the dominating ratio are: (1) it is asymmetric, (2) it excludes reflexive uses-pairs of finest grain entities, (3) it is defined for any two entities, (4) it optionally crosses multiple levels in the decomposition structure of a system (see [1]), and (5) it is defined relative to a tree-cut.

Below, we show how the dominating ratio provides adequate support for re-architecting in several situations common for large legacy systems, where coupling and cohesion fail to do this. We illustrate the situations using a simplified (strictly) layered unbalanced system S as an example (see Figure I and II).

2.1. **Identify asymmetric relations**

Many large software systems have a layered module architecture. The notion of a layered structure and its advantages with respect to ease of development have already been described in [4]. The benefits of systems with a layered architecture in terms of development effort and costs have been verified empirically in [10]. A (degenerated) example of a (strictly) layered system S and a subsystem B is shown in Figure I(a) and (b), respectively. Note that asymmetry (e.g. B1 uses B2 but not vice versa) is an inherent property of layering. The coupling metric does not reflect this asymmetry, however. The coupling between finest grain entities B1 and B2 is equal to the coupling between B2 and B1 (C(B1, B2) = C(B2, B1) = ½). The dominating ratio identifies such an asymmetric relation (DR(B1, B2) = 1/(1×1) = 1 and DR(B2, B1) = 0/(1×1) = 0).

2.2. **Identify non-connectivity of an entity**

In Figure I(c), a subsystem C is shown with two constituents C1 and C2. Let’s assume that these two constituents are the finest grain architectural entities under consideration. Although C contains these two constituents, C1 and C2 have no inter-relationship. This is not reflected in the cohesion of C, which is unequal to zero (C(C) = 2/(2×2) = ½). The dominating ratio for this situation is zero (DR(C, C) = 0/(2×(2−1)) = 0), hence reflecting the fact that the constituents of C are independent. This is accomplished by excluding reflexivity of the finest grain architectural entities in this metric.

2.3. **Identify candidates for movement**

In Figure I(b), B2 is both used by and using other entities within B. On the other hand, B1 is not used by and B3 does not use other entities within B. Hence, B1 and B3 are candidates for movement to subsystem C and A, respectively. Preferably, a metric assists in identifying candidates for movements. This can be done by defining the connectivity of a constituent with respect to (its complement within) a constituting entity, e.g. B1 and B, and vice versa. Coupling between B1 and B and vice versa is undefined, whereas the dominating ratio is
defined ($\text{DR}(B_1,B) = 2/(1 \times (3-1)) = 1$ and $\text{DR}(B,B_1) = 0/(3-1) \times 1 = 0$). For any entities $X$ and $Y$, where $X$ is part of $Y$, $\text{DR}(X,Y) = 0$ (respectively $\text{DR}(Y,X) = 0$) indicates that $X$ is not using (respectively not used by) entities within $Y$, and hence, identifies $X$ as a candidate for movement. An entity $X$ which is not using (respectively not used by) other entities within $Y$ is termed a “bottom” (respectively a “top”) of $Y$.

2.4. Optionally crossing multiple levels

Coupling and cohesion are defined for arbitrary sets of leaf entities in [8] (i.e. independent of the decomposition structure of a system), and the re-clustering approach described leads to a balanced system, when applied repeatedly [8]. On the contrary, the decomposition structure of a system has been taken as one of the starting points in the definition of the dominating ratio. The dominating ratio is applicable for both balanced and unbalanced systems, and defined across multiple levels. The decomposition structure of $S$ is shown in Figure II.

2.5. Definition relative to a tree-cut

How to define the connectivity of $S$? Figure II shows (all) 5 conceivable tree-cuts. A tree-cut is a line crossing each path from the finest grain architectural entities (e.g. $B_2$) to the root (i.e. $S$) exactly once, where each path is crossed “through” an architectural entity ([11]). Hence, there are 5 different ways to define the connectivity of $S$ corresponding with these 5 tree-cuts! Stated in other words, the value of connectivity is relative to a particular tree-cut, and the dominating ratio is defined as such.

A relative definition may come as a surprise, but is felt to be natural when considering the fact that the decomposition structure of $S$ could also be extended to entities at the programming level. To our knowledge, (explicitly) defining an architectural metric relative to a tree-cut is novel.
3. Visualising the dominating ratio

The dominating ratio can be calculated for every pair of (architectural) entities of a system. Given a large system (where the number of architectural entities may be in the order of magnitude of 8K; see [2]), how to visualise such a large amount (64M values for the dominating ratio) of information?

Experience within Philips ([2]) revealed that a tabular representation with browsing facilities is an effective and efficient way for a systematic investigation of the module architecture of a system (and complements a graphical representation). The tabular representation is based on the decomposition structure of a system and browsing corresponds with horizontal and vertical traversals through the decomposition tree. We use similar visualisation means for the dominating ratio as well. In our view, such a means facilitates a systematic investigation of the software architecture for incremental re-architecting of large legacy systems. We have extended an existing module architecture browser (MAB; see [2]) with the dominating ratio, yielding a single means for visualising both quantitative (and qualitative) structural module architecture information and metrics.

Note that it is quite uncommon to represent quantitative information by means of tables ([9]). To our knowledge the usage of a tabular representation with browsing facilities for visualising architectural metrics is novel.

Below, we show how the dominating ratio is presented by the MAB and briefly describe the basic means for navigation provided by the MAB.

3.1. Presentation

The basic form in which the MAB displays information is a table. Figure III shows the system S and a screen shot of the MAB visualising the dominating ratio.

![Figure III: System S and Dominating Ratio visualised by the MAB](image-url)
The values shown in Figure III(b) correspond with the tree-cut through the finest grain architectural entities in Figure II. The dominating ratios from one dominating entity (the system S) to one dominated entity (also the system S) are displayed. There is a row for each child of the dominating entity and a column for each child of the dominated entity. The cells of the table contain the dominating ratio between the corresponding children multiplied by the so-called multiplier (displayed directly above the table). Hence, the corresponding dominating ratio is obtained by dividing the cell value by the multiplier. A table may contain both empty cells (e.g. $\text{DR}(B,C)$) and cells containing zero (e.g. $\text{DR}(C,C)$). Empty cells denote the absence of uses relations, whereas cells containing zero denote the presence of reflexive uses relations only. The layout of a table is determined by the order of its rows and columns, which may be provided by a user by means of a layout relation. The fact that S is a strictly layered system (see Figure I(a)) follows immediately from the table shown in Figure III(b): the bottom-left triangle is empty and the top-right triangle contains only values adjacent to the diagonal.

3.2. Navigation

The MAB provides, amongst others, means to zoom-in and zoom-out on rows, columns and cells of the table and to exchange rows and columns in a table. As an example, the tables displayed in Figure IV(a) and Figure IV(b) are the result of clicking on B in the row and column (i.e. zoom-in) in the table in Figure III(b), respectively. Similarly, Figure III(b) is the result of clicking on S in the row of Figure IV(a) or the column of Figure IV(b) (i.e. zoom-out), respectively.

Figure IV: Screenshots of the MAB

Zoom-in and zoom-out are ways to traverse vertically through the decomposition structure. It is possible to traverse horizontally through the decomposition structure by clicking on the names of siblings (e.g. C above or A below B in Figure IV(a)). Note that Figure IV(a) is the result of exchanging the rows and columns in Figure IV(b) (and vice versa). The fact that $B_3$
is a bottom of $B$ and $B_1$ is a top of $B$ follows from Figure IV(a) and Figure IV(b), respectively.

4. Initial Experience

In this section, we briefly report upon a small experiment conducted with the dominating ratio using the MAB. The goal of the experiment was to identify candidates for movement in order to resolve layering violations in a large software-intensive legacy system termed SOPHO. SOPHO may be viewed as a family of private branch exchanges (i.e. telephony systems in a business environment), which is developed and maintained by Philips Business Communications since the early 1980s. The core of that system consists of approximately 5 K files (containing 2.5 MLOC written in C++), 35 K includes between files, and 8 K architectural entities (i.e. subsystems, components, modules and files), organised in an unbalanced tree (representing the decomposition structure of the system) with a depth ranging from 5 to 12.

4.1. Results

Amongst others, the following results were found during the experiment:

- **different kinds of entities are candidate for movement:**
  
  Some of the layering violations could be resolved by moving individual files, modules, and sometimes even components. Candidates for movement include "tops" and "bottoms" (see Section 2.3), which may be easily identified with the help of the dominating ratio.

- **moving candidates is not always the right way for resolving layering violations:**
  
  Some of the identified candidates for movement should not be moved. By moving those entities, they would become part of a constituent to which they do not belong conceptually. Note that in these situations automatic re-architecting tools would yield structures which are awkward from an application domain expert’s point of view.

- **generic entities prevail specific entities as candidates for movement:**
  
  A common reason for layering violations turned out to be that functionality introduced at a particular layer became of a more general nature during the evolution of the system. Rather than moving those entities “down” in the structure, they were simply used by lower layers. We hardly identified entities which should have been more specific (i.e. moved “up” in the structure).

- **many violations can not be resolved by simply moving architectural entities.**
4.2. Lessons Learned

The experiment revealed that the dominating ratio and its visualisation by means of the MAB is effective to identify candidates for movement in order to resolve layering violations. Amongst others, the following additional insights were gained during the experiment:

- **means for modification of information desired:**
  As its name reflects, the MAB is a module architecture browser. The MAB provides no means to modify either a uses relation or a part-of relation. In some situations such modifications are desirable. E.g. whenever a set of mutually dependent entities is a candidate for movement, it is desirable to be able to create a virtual entity containing those mutually dependent entities (changing the part-of relation).

- **dominating ratios presented by the MAB did not always reflect a user’s intuition:**
  Currently, the MAB presents values of the dominating ratio relative to the tree-cut through the finest grain architectural entities. This choice did not turn out to be self-evident in many situations. As an example, consider Figure I(c) and assume that neither C₁ nor C₂ are the finest grain architectural entities (i.e. both have constituents). In such a situation, \( DR(C,C) \) is unequal to zero, despite the fact that C₁ and C₂ are independent.

Acknowledgements

We thank Wim J. Christis, Rob C. van Ommering and Tobias Rötschke for their comments on a previous version of this paper, and Wim Decroix for extending the MAB so as to include calculation and visualisation of the dominating ratio.

References


Appendix: Formalisation

The formalisation presented in this appendix is based on relation algebra ([5]), and uses “oblique” lifting ([11]). The module architecture of a system $S$ is described by a triple $<E, U, P>$: a set of (architectural) entities $E$, a uses relation $U$ (describing the dependencies between finest grain entities) and a part-of relation $P$ (describing the decomposition structure of $S$). We assume that $E$, $U$, and $P$ are non-empty.

**Coupling and Cohesion**

Let $G_1$ and $G_2$ be sets of finest grain entities (or leaves) of $S$. Coupling for disjoint sets $G_1$ and $G_2$ ($\mathcal{C}(G_1, G_2)$) and cohesion ($\mathcal{C}(G_1)$) are defined by:

- coupling:
  $$\mathcal{C}(G_1, G_2) = \mathcal{C}(G_2, G_1) = (\mid U \mid_{dom} G_1 \mid_{ran} G_2 \mid + \mid U \mid_{dom} G_2 \mid_{ran} G_1 \mid)/(2 \times |G_1| \times |G_2|)$$

- cohesion:
  $$\mathcal{C}(G_1) = \mid U \mid_{car} G_1 \mid / \mid G_1 \mid^2$$

where $U \mid_{dom} G_1$, $U \mid_{ran} G_1$, and $U \mid_{car} G_1$ denote the relation $U$ restricted to the set $G_1$ in its domain, range, and carrier (i.e. both domain and range), respectively.

Consider the formula for coupling. The term $\mid U \mid_{dom} G_1 \mid_{ran} G_2 \mid$ in the numerator represents the number of uses relations from the leaves in $G_1$ to the leaves in $G_2$ (the other term is similar). The denominator represents the maximum number of possible uses relations between the leaves in $G_1$ and the leaves in $G_2$. 

400, April 1998.


Dominating Ratio

Hiding entities up to a particular tree-cut $T$ has been described in depth in [1]. Set $T$ is defined as the set of all entities on the tree-cut $T$. We consider a system $S_T$ which is derived from system $S = \langle E, U, P \rangle$ and tree-cut $T$, and which is described by a triple $\langle E_T, U_T, P_T \rangle$. The uses relation $U_T$ is obtained from $U$ by “existence-oriented” lifting $U$ to tree-cut $T$ and removing all reflexive uses relations from it. The set of entities $E_T$ (respectively the part-of relation $P_T$) is obtained from $E$ (respectively $P$), by removing all elements below tree-cut $T$ from $E$ (respectively $P$). The following sets are defined for notational convenience, where $A, B \in E_T$, and $P^*$ denotes the reflexive transitive closure of $P$ ($L_T(B)$ and $L_T(B \setminus A)$ are defined similarly):

- leaves of $A$ w.r.t. tree-cut $T$: $L_T(A) = \{ \text{dom}(P^* \downarrow \text{dom} T \downarrow \text{ran} \{A\})\}$;
- leaves of $A$ excluding the leaves of $B$ (i.e. the complement of $B$ in $A$) w.r.t. tree-cut $T$: $L_T(A \setminus B) = L_T(A) \setminus L_T(B)$.

The dominating ratio $\text{DR}_T(A,B)$ for the four situations - where situations 2 and 3 represent the nested cases ($P^+_T$ denotes the transitive closure of $P_T$) - is defined as:

1. $A = B$:
   $$\text{DR}_T(A,A) = \frac{\{U_T \downarrow \text{dom} L_T(A) \downarrow \text{ran} \{L_T(A) \downarrow L_T(A)\} \times \{L_T(A) \downarrow 1\}}{\{L_T(A) \downarrow L_T(A)\} \times \{L_T(A) \downarrow 1\}}$$
   for $\{L_T(A)\} > 1$ (and $\text{DR}_T(A,A) = 0$ otherwise)

   By using $\{L_T(A)\} \times (\{L_T(A)\} - 1)$ in the denominator (rather than $\{L_T(A)\}^2$) reflexivity is excluded.

2. $\langle A,B \rangle \in P_T^+$:
   $$\text{DR}_T(A,B) = \frac{\{U_T \downarrow \text{dom} L_T(A) \downarrow \text{ran} L_T(B \setminus A) \downarrow \{L_T(A) \downarrow L_T(B \setminus A)\} \times \{L_T(A) \downarrow L_T(B)\}}{\{L_T(A) \downarrow L_T(B)\} \times \{L_T(A) \downarrow L_T(B)\}}$$
   for $\{L_T(B \setminus A)\} \neq 0$ (and $\text{DR}_T(A,B) = 0$ otherwise)

   By taking the complement of $A$ within $B$ (i.e. $L_T(B \setminus A)$), intra-constituent dependencies (i.e. of $A$) are excluded.

3. $\langle B,A \rangle \in P_T^+$:
   $$\text{DR}_T(A,B) = \frac{\{U_T \downarrow \text{dom} L_T(A \setminus B) \downarrow \text{ran} L_T(B) \downarrow \{L_T(A \setminus B) \downarrow L_T(B)\} \times \{L_T(B) \downarrow L_T(B)\}}{\{L_T(A \setminus B)\} \times \{L_T(B)\}}$$
   for $\{L_T(A \setminus B)\} \neq 0$ (and $\text{DR}_T(A,B) = 0$ otherwise)

4. Others (A and B are disjoint):
   $$\text{DR}_T(A,B) = \frac{\{U_T \downarrow \text{dom} L_T(A) \downarrow \text{ran} L_T(B) \downarrow \{L_T(A) \downarrow L_T(B)\} \times \{L_T(B) \downarrow L_T(B)\}}{\{L_T(A)\} \times \{L_T(B)\}}$$

For sets $G_1$ and $G_2$ containing leaf entities $L_T(A)$ and $L_T(B)$, coupling of $G_1$ and $G_2$ may also be expressed as $C(G_1, G_2) = (\text{DR}_T(A,B) + \text{DR}_T(B,A))/2$. By ignoring reflexivity (in both numerator and denominator of $C(G_1)$), $C(G_1)$ is similar to $\text{DR}_T(A,B)$. Hence, the dominating ratio unifies coupling and cohesion.