DELTA software technology

AMELIO
LOGIC DISCOVERY

Analyzing a Big Ball of Mud

The clean design of an application often gets lost over decades. Applications arise where design specifications are not met. Such applications are difficult to maintain and even more difficult to modernize. AMELIO Logic Discovery helps identify design violations in an iterative process.

From clean design to Big Ball of Mud

How could this have happened? When the so-called legacy applications were developed, there was a clean design. For example, it was determined which domains or sub-applications exist, which programs belong to which domain and how communication between the domains should take place. However, several decades have passed since then. The applications were passed from developer to developer, they were adapted, expanded and revised, new sub-applications were added, often under time pressure. The clean design has been lost. Communication is unstructured, and domains directly access artefacts that are within the purview of other domains. Often, the communication channels are neither documented nor known. In short: a big ball of mud has emerged.

However, if it is unclear who is accessing certain artefacts, then unwanted side effects of changes are inevitable. During modernizations, it becomes difficult to replace partial applications or domains with newly written or standard software as well. In the worst case, resulting errors only appear at runtime. For a safe and efficient maintenance or modernization, it is important to define the domains, their interfaces, and the permissible interactions, and to analyze, where design violations occur. This analysis can be performed automatically using AMELIO Logic Discovery.

Iteratively detect design violations

AMELIO works both model- and rule-based. First, all sources of the application are converted into models. The rules for analyzing the design violation can then be executed on these models. These rules can, for example, describe which communication between programs is permitted, what restrictions there are on the use of copybooks, or who, how, may access which database or file. The set of rules is customer-specific and is defined in an iterative process together with the customer. First, the simple, rough rules are specified and then further refined .

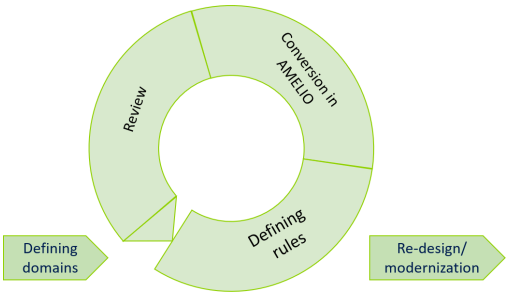

Figure 1: Iteratively detect design violations

AMELIO
LOGIC DISCOVERY

DELTA software technology

Who interacts with whom, when & how?

First, you need to determine which domains exist and which program belongs to which domain. The assignment of programs to domains can be described by rules, often based on program names, for example. In the next step, AMELIO determines the program trees, that means which program calls which other program. Programs that do not have a caller are potential entry-level programs. For all detected program calls, AMELIO checks whether the involved programs belong to the same domain or not

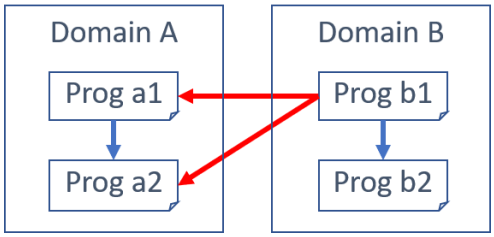

Figure 2: Communication across domain boundaries (red)

Is he allowed to do this?

AMELIO initially assumes that communication between the domains is not permitted. Therefore, it is now necessary to define which types of interaction across domain boundaries are permissible.

- Service programs: the program is a service through which programs from other domains can request or write data. The program can be defined as a whole service or some functions of it.
- Program pairs: it is specified that certain program pairs from different domains may interact with each other.
- Further rules can be defined in the next iteration steps.

AMELIO uses these rules to assess the interactions between the domains and to represent critical interactions separately.

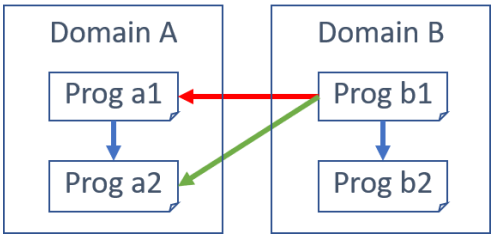

Figure 3: Communication across domain boundaries (red prohibited, green permitted)

Copybooks

To modernize an application, it may be necessary to analyze and reorganize the use of copybooks. To do this, AMELIO first determines which copybooks are used by the individual programs. Next, it is analyzed whether a copybook is used by programs from different domains. If this is the case, it will be determined for each line of the copybook, in which domains it is needed.

If there are copybooks that are used by programs from different domains, it is necessary to define in the review step, which uses are permitted and which are not.

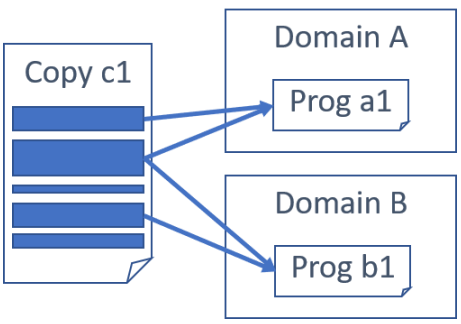

Figure 4: Copybook with usage in different domains

DELTA software technology

AMELIO
LOGIC DISCOVERY

Data thieves – caught red-handed

The original clean design has clearly defined, who has read or write access to a database. However, upon closer inspection, you realize that there are one or two data thieves in the application now. This refers to programs that have access to databases, to which they should not actually have (direct) access. Such data thieves can also be identified automatically. The databases are assigned to the different domains. The rule is: only programs from the same domain may access a database. AMELIO determines for all databases, which programs have read or write access to the databases, with their respective gaps or segments. Access from another domain is highlighted accordingly.

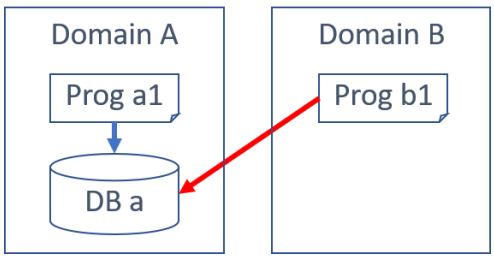

Figure 5: Database accesses across domain boundaries (red)

Analyzing a Big Ball of Mud

AMELIO Logic Discovery can analyze interactions and dependencies in an application. An iterative process can also be used to analyze, whether this application violates design specifications. This is the basis for maintaining, refactoring or modernizing applications efficiently and successfully.

DELTA software technology

Delta Software Technology GmbH

Eichenweg 16, 57392 Schmallenberg, Germany
phone +49 2972 9719-0
e-mail info@delta-software.com

www.delta-software.com

AMELIO Logic Discovery

Understand COBOL and PL/I applications: Reduce costs and risks for maintenance, modernization, and new implementations.

www.delta-software.com/amld

AMELIO Logic Discovery

Order Number: MT21106.01 – July 2025