

IN THIS ISSUE

1. [AMELIO Logic Discovery: Understanding COBOL and PL/I-Applications](#)



2. [The Cobbler always wears the best Shoes](#)



3. [AmAVaG - Understand your Generating Systems](#)



Get in touch with us



Delta Software Technology GmbH
 Eichenweg 16
 57392 Schmallenberg, Germany

phone +49 2972 9719-0
 fax +49 2972 9719-60
 e-mail info@delta-software.com

www.delta-software.com

1 AMELIO Logic Discovery: Understanding COBOL and PL/I-Applications

We extend our product range for the software modernization. With



our new product AMELIO Logic Discovery we now offer an analysis tool that helps you to better understand your COBOL and PL/I applications and to extract the relevant functionality from your extensive applications. With its analyses, statistical evaluations and varied functions for displaying the application logic AMELIO Logic Discovery supports you whenever it is necessary to understand existing applications, e.g. when deciding for a modernization project or in projects for re-implementation.

(Technological) Generation Change

The generation change has begun, on the human level as well as on the technological level. Young developers are confronted with applications designed some 20 or 30 years ago and implemented with languages and paradigms which are hardly taught today.

Already the original systems were large and complex, since then they have been changed, extended and rebuild whereby size and complexity have further increased. As a result of this, even experienced COBOL and PL/I developers have problems to understand the applications as a whole.

The applications precisely implement the core business functions of the enterprises. The business logic is still up-to-date but the implementation is out-of-date. To develop this business logic, huge investments have been made. If the existing applications would be “thrown away” the same would apply to the investments. If you re-specify the applications without understanding the existing applications there is a danger that parts will be overlooked or specified unlike they have been implemented before. That’s why it is essential to understand what an application does and how, which algorithms and calculation rules have been implemented and how – detached from language-dependent implementation details.

AMELIO Logic Discovery answers these and other questions and thereby helps to understand existing applications and to master the generation change.

Making Application Logic Visible

AMELIO Logic Discovery helps you to understand your COBOL and PL/I applications: what are the applications doing? How do they do it? Which algorithms and calculation rules have been implemented and how?

The first step for understanding an application is to transfer it into a language-neutral model. For the collected data already first dependencies can be identified. The gathered information is stored, e.g in a repository.

By means of the collected information the application can be inventoried and statistical analyses can be performed. Furthermore, ad hoc requests can be made and the information can be navigated on.

But to really understand the applications the information must be connected and logical conclusions must be drawn from it. We don't leave the developer alone with the task to draw conclusions from the mass of data that is stored in the repository. Instead, AMELIO Logic Discovery provides another abstraction layer where functional abstractions can be performed and logical dependencies can be determined. The knowledge produced by the reasoning engine can arbitrarily be extracted, extended or modified.

In this way AMELIO Logic Discovery helps you to understand your application.

Really Understand Software

With our AMELIO Modernization Platform we have proven that our tools can perform large and complex software transformations fully automated.

Now we make this matured technology available to our customers for the analysis of their COBOL and PL/I applications.

AMELIO Logic Discovery helps to understand the existing COBOL and PL/I applications and thus reduces the costs for re-implementation of the existing functions and for the modernization of the applications.

As AMELIO LogicDiscovery helps to understand applications, the aim of our new, BMBF-supported project AmAVaG is to analyse generator systems. Further information on that can be found in our article "AmAVaG – Understand your Generating System".

2 The Cobbler always wears the best Shoes

The Cobbler's Children are Best Shod: The Solution of the MDD Paradox

[[Daniela Schilling](#)] Generators and domain specific languages (DSLs) accompanied by models are the central elements of model-driven development

but paradoxically they themselves get developed by using the conventional methods instead of model-driven ones. As a result, the development of generators and DSLs is considered as a time-consuming and error-prone hermetism. In this article we present a solution which solves this paradox and significantly reduces development effort and complexity.



The article has been published in [OBJEKTspektrum 06/2012](#).

[The english translation can be found here.](#)

Further Information

[HyperSenses](#) - Integrated system for model driven development of DSLs and software generators.

3 AmAVaG - Understand your Generating Systems

For decades the development of complex and business critical applications is automated via macro-based generation techniques – as Delta ADS, FSP SWT01, IBM VisualAge Pacbase or Micro Focus Application Master Builder. The basis are generating systems consisting of default processors and macros, individually developed macros, configurations and further artefacts. These generating systems have the same complexity as the generated systems themselves. But missing or insufficient documentation as well as the age-related retirement of the developers result in decreasing knowledge of the generating system and adaptations, modernisation or rewriting of the systems get more and more complicated.

Automated Architecture and Variability Analysis of Generating Systems (AmAVaG)

Within the [project AmAVaG](#) – promoted by the Federal Ministry of Education and Research – we

develop tools which automatically analyse and represent the architecture of individually developed macro-based generating systems and transform the results into models. Thus the following questions are addressed:

- which generating components got used in a generating process
- which parts of different generating components are responsible for the generation of a certain part of code
- which dependencies and excluding criteria exist between generating components
- how do certain values of the configuration effect the generating process
- and further more

Thus we help you to regain the knowledge over your generating system and thereby provide the basis for its modernisation. By the way you also get the re-documentation of your generating system.

Be part of it!

In addition to the usage of specifically created test data we are looking for partners who create applications by macro-based generating techniques and who provide parts of it for the project. Of course your data will be dependably made anonymous and kept in confidence. Be part of it early on, influence

NEWSLETTER

ISSUE DECEMBER 2012



the development, tell us your requirements for such an analysis tool! We promptly present you results and seize your suggestions.

Be part of it...

Further Information

AmAVaG - Automatic Architecture and Variability Analysis in Generating Systems

Delta ADS - Application development for COBOL and PL/I

Participate at AmAVaG now...



Funding number: 01IS12043

More newsletters and our newsletter administration can be found here:
www.delta-software.com/newsletter

