EB solys
Runtime analysis with DLT and beyond

Torsten Mosis, Software Architect, systemticks GmbH | 15th May 2019
Introduction
EB solys

• is a powerful tool or rather a construction kit for runtime analysis of complex software systems

• collects multiple streams of events, logs, and tracing information from different sources, combines them on a single timeline for analysis and visualizes the results

• has its roots in automotive IVI, but can be applied for other domains, such as IoT or consumer electronics

• has built-in support for a couple of GENIVI features, such as Automotive DLT, D-Bus and Franca IDL

• is based on Eclipse RCP and is running under Windows, Linux and MacOS

• is designed to be easy extensible and customizable

• is an open source project since October 2018, under EPL 2.0 and is maintained by systemticks
Brief History

- **08/2012**: Started as an internal project at EB for QNX and Win projects.
- **10/2015**: Support for DLT, D-Bus and Franca IDL.
- **03/2017**: Launched Target Agent as Open Source project.
- **10/2018**: Launched EB solys as Open Source project.
- **11/2013**: Target Agent running on Linux.
- **06/2016**: EB solys auto (no GUI).
- **08/2018**: Target Agent running on Android.
- **>= 2019**: Maintained by systemticks.
Objective

- Gain a **greater insight** into the **operational activity** of **complex software systems**

- Enable the development team creating a **joint system understanding**

- **Localize** and **isolate** errors **faster** and with significant **less workforce** than usual

Provide a **toolchain** with the focus on gathering **valuable runtime data** from **different sources** and setting them **in relation to each other**
## Architecture

<table>
<thead>
<tr>
<th>Data visualization</th>
<th>GUI</th>
<th>CLI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data preparation</td>
<td>Event Hooks</td>
<td>Scripting</td>
</tr>
<tr>
<td></td>
<td>Decoder</td>
<td></td>
</tr>
<tr>
<td>Data collection</td>
<td>Runtime Event Management</td>
<td>Importer</td>
</tr>
</tbody>
</table>
Example Setup with DLT

EB solys provides built-in features and APIs for the purposes of:

- demultiplexing
- visualizing
- filtering & searching
- aggregating
- decoding
- correlating
- transforming
- automating

across different data sources in a single place.
Customization and Extension Points

With **Event Hooks** one can intercept single runtime events and transform or restructure its data contents.

Hook in your own **Decoder**, that transforms arbitrary non-primitive data (e.g., binary payload) into structured readable text.

**Built-In Support:**
- D-Bus
- Franca IDL (incl. Contracts)
- Protobuf
- Json
- FIBEX (under development)

Provide your own **Communicator** to be able to connect to any trace provider.

**Built-In Support:**
- DLT daemon (TCP/IP)
- DLT via UART (under development)
- EB solys target agent (Linux and Android)

Use the Built-in **Script Engine** to add new functionality by accessing the EB solys raw data and UI resources through a powerful API.

Scripts are developed and executed directly within EB solys, which allows very short turnaround cycles.

Scripts are developed with Xtend

Provide your own **Importer** for any kind of log file

**Built-In Support:**
- Automotive DLT 4.0
- Target Agent Logs
- Android Logcat
- PDML
Features
Feature Overview

- **Graphical correlation** and analysis of target runtime data
- Hot spot analysis & **root cause isolation**
- Static and dynamic **communication analysis**
- GENIVI-ready with support for Linux, DLT, D-Bus and Franca IDL
- **Extension points** for custom importer and decoders to process any trace format
- **Built-In Script Engine** to add new functionality by accessing the EB solys raw data and resources
- **Multi-OS** (Linux, Android, QNX) target-agent with **extensible plug-in architecture**
- Out-of-the-box plug-ins available for analyzing system resources, inter-process communication and application traces
- **Integration** into continuous build and test toolchain with EB solys Auto mode
Data correlation from different sources

Select a Timemarker

Traces all around the Timemarker

Sent D-Bus message with content at that Timemarker

Resource consumption (CPU load) at that Timemarker
Stacked Decoding

Franca Decoder

D-Bus Decoder

DLT Decoder
Stacked Decoding

Franca Decoder

D-Bus Decoder

DLT Decoder
Sequence Logic Visualization
Structural Inspection
Data Analysis using scripting capabilities

- Feature Set can be enriched on-the-fly with (Xtend) scripts
- API to access all collected runtime data and most of the GUI elements
- Full-featured built-in editor with syntax highlighting, code completion, content assist, javadoc, etc.
- Scripts can also be executed in „live mode“ (callback scripts)
- Fully Java compatible
- Invocations from Xtend code into Javascript code possible. And vice-versa
Download & Installation
Installation & Development

• EB solys  https://github.com/Elektrobit/eb-solys

• Data Sources
  • DLT  https://github.com/GENIVI/dlt-daemon
  • Target Agent Linux  https://github.com/Elektrobit/eb-solys-target-agent
  • Target Agent Android  https://github.com/Elektrobit/eb-solys-android-agent

• Development Prerequisites
  • Java 8 Runtime Environment
  • Maven 3.3.1 or higher
  • Eclipse Installer
  • openjfx (for Linux only)
See also

• EB solys Introduction
  • Product Homepage
  • Technical article (german)

• EB solys & Franca Contracts
  • Contract-based software development with Franca
  • EB solys and Franca Contracts – An overview
  • Retrace and validate the usage of interfaces

• EB solys, DLT & Fluent Bit
  • Automotive logging on Linux with Fluent Bit, DLT and EB solys
    https://systemticks.de/2019/02/27/automotive_logging_1.html
  • Visualize Fluent Bit metrics with EB solys
    https://systemticks.de/2019/03/15/fluentbit_and_solys.html
Thank you! Any Questions?

Visit GENIVI:
http://www.genivi.org
http://projects.genivi.org

Contact:
torsten.mosis@systemticks.de
help@genivi.org
help@genivi.org