Extensible Logger And TracEr

Concept paper

Author: Boris M. Vigman

Contact info: boris_vigman@hotmail.com
bv_o_gel@lycos.co.uk

Version 1.1 (ELATE 0.3.5)
November 2003

Table of Contents

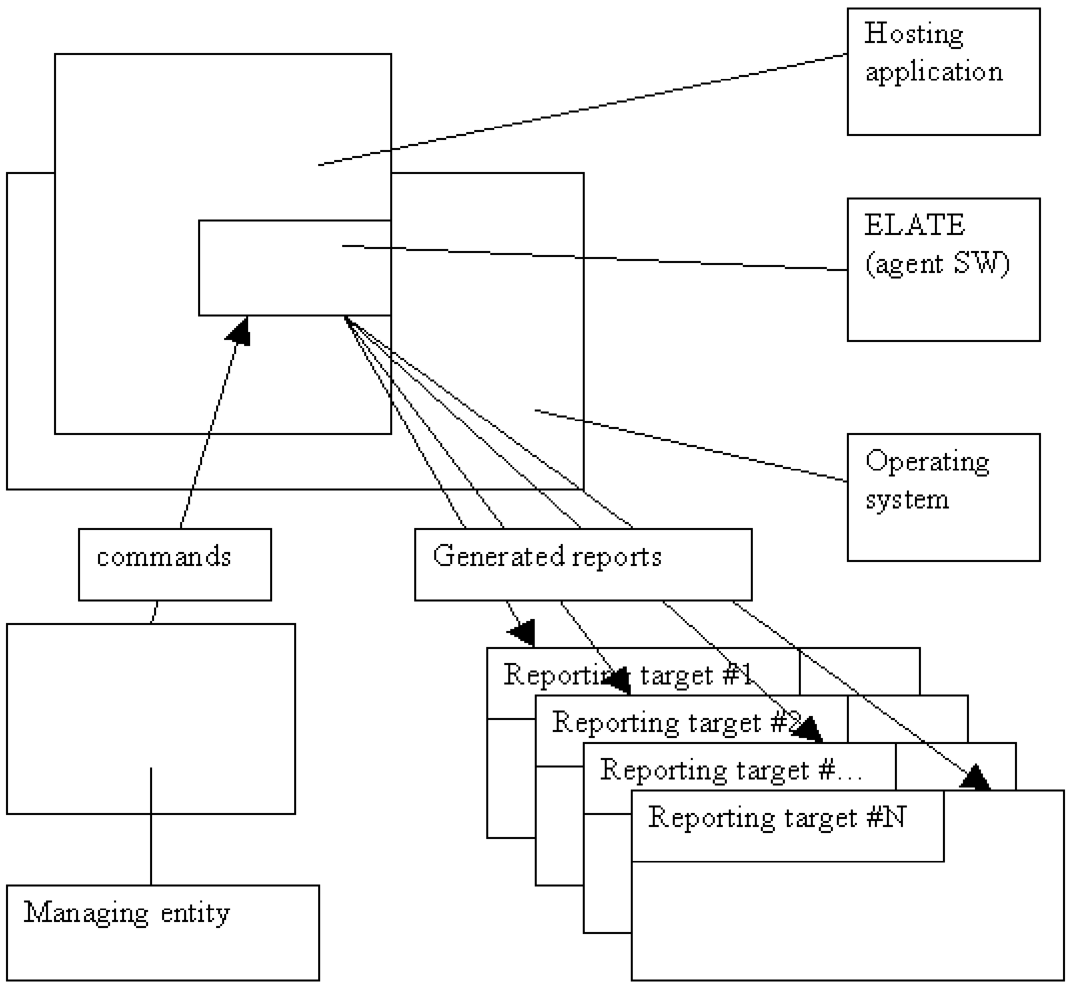
Introduction

ELATE is a software mechanism (written in C) that is intended to give more or less unified solution for emitting of messages (called “reports”) from inside the given software into the outer world for subsequent processing. This mechanism is designed with keeping in mind the next guidelines:

1. Maximally possible independency upon specific operating system;
2. Maximally possible independency upon specific HW architecture and configuration;
3. Maximally possible independency upon structure and content of forwarded information;
4. Maximally possible independency on characteristics and intentions of final consumers of generated messages (called “reporting targets”;
5. The mechanism must be controllable from the outside world – until it does not affect normal functioning of the hosting software;
6. Usage of this mechanism must be as simple as possible.

SW interaction overview

The next drawing shows major entities that may interact with ELATE

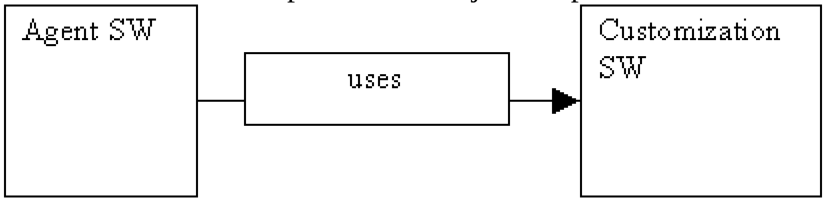


Brief interaction description

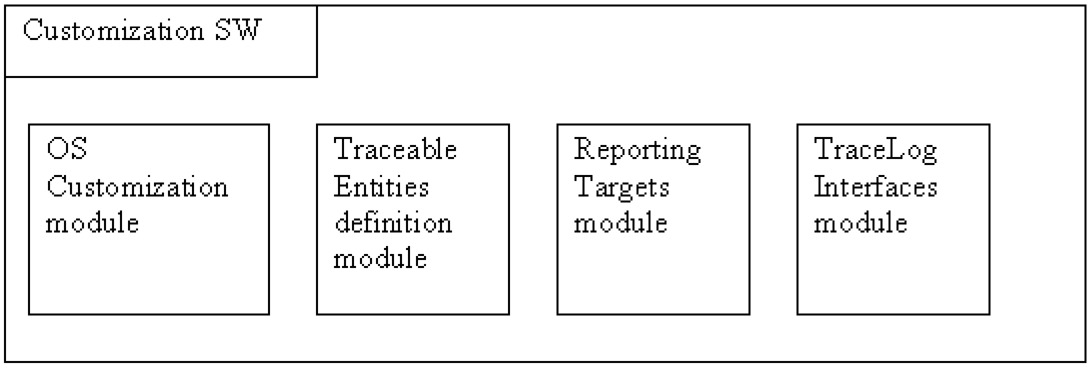
The ELATE SW is compiled together with the C/C++ code ("hosting application") that is expected to generate reports; into the code of hosting application calls to ELATE API are inserted. After the compilation the hosting application starts to generate reports (kinds of reports are predefined by the ELATE API) and to forward them to the bunch of "reporting targets".
At the same time the ELATE SW mechanism may be managed from the outer entity (task, file, workstation etc) in order to change its functionality in some way.

SW architecture overview

The ELATE SW is comprised of 2 major components:



The Agent SW is the core element of the SW – it collects and forwards the reports towards outer reporting targets.
Customization SW consists of several parts:

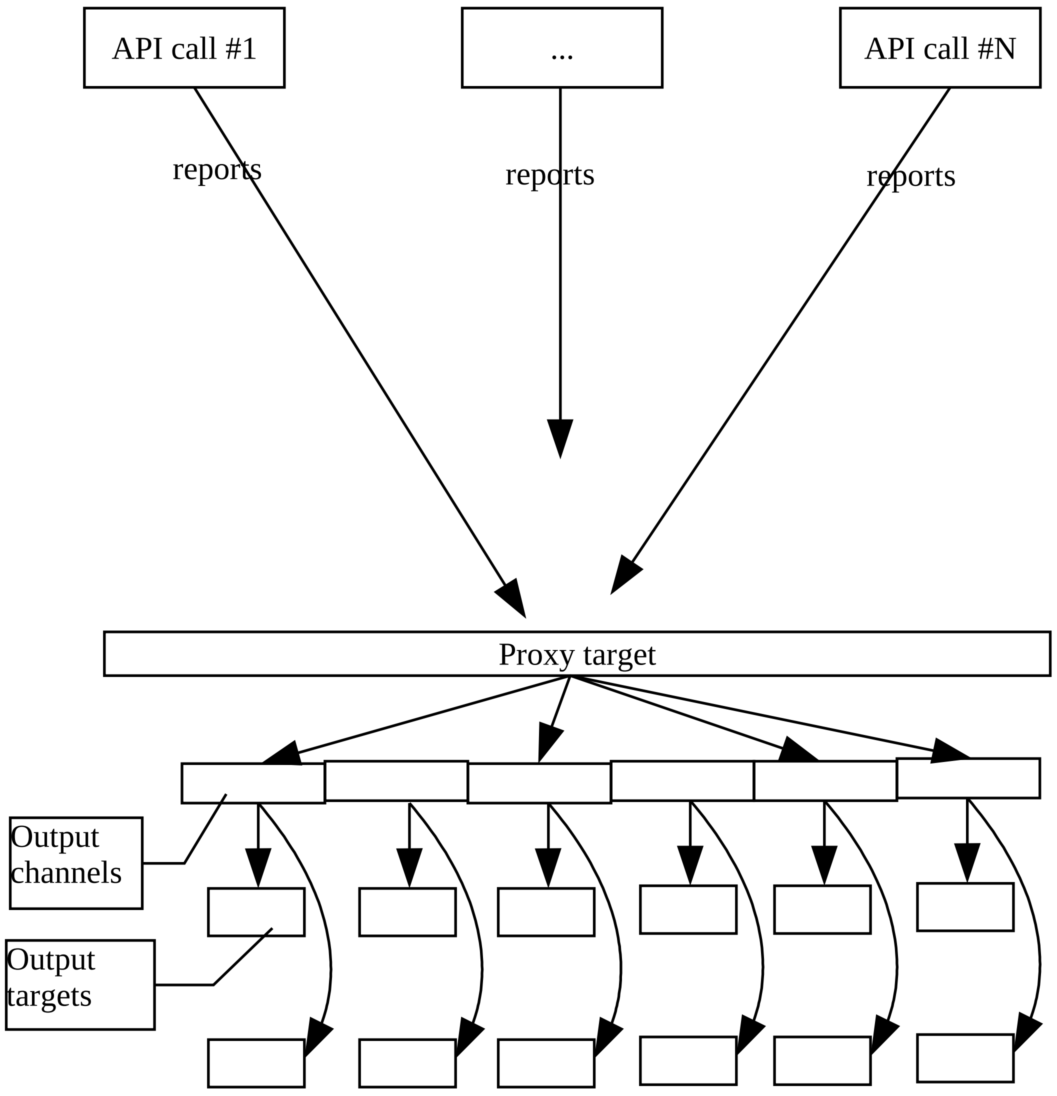


OS Customization module provides implementations of all OS-specific services that are required by ELATE.
Traceable Entities (TE) module provides definitions of all report types that re able for generation by ELATE in **its specific configuration**.
Reporting Targets module provides definitions and implementations of all output targets (console, another task, file, printer etc) that are used by ELATE in **its specific configuration.**
TraceLog Interfaces module provides set of macros used for generation of reports for each of two possible work modes, namely tracing and logging[1].

1 It must be underlined that ELATE may generate 2 kinds of reports: logs and traces. Those modes are independent so both kinds may be generated simultaneously, or one of them may be generated, or none. In the latter case ALL components of ELATE are totally removed from the hosting application at the compilation stage.

Interaction between ELATE components

The next diagram shows how different components of ELATE interact and cooperate between them.



Brief description of ELATE components interaction diagram

Report processing inside ELATE is the next:

1. ELATE API call generates the report;
2. Reports are posted to Proxy Target – an entity that hides real targets behind the unique interface;
3. Proxy Target forwards the report to predefined set of channels, each of which is associated with number of reporting targets (currently two targets per channel).