

# EDOS project: developing the Technology for future Linux Distributions

Paulo Trezentos<sup>1</sup> ([Paulo.Trezentos@caixamagica.pt](mailto:Paulo.Trezentos@caixamagica.pt)),  
Stéphane Laurière<sup>2</sup> ([slauriere@mandriva.com](mailto:slauriere@mandriva.com))

1- Caixa Mágica / ADETTI / ISCTE 2- Edge IT / Mandriva

Poster Contact: Paulo Trezentos, Caixa Mágica Software, Edifício Espanha – Rua Soeiro Pereira Gomes, Lote 1 - 8º F, 1600-196 LISBOA, Portugal, +351 217 921 260

## Introduction

EDOS stands for *Environment for the development and Distribution of Open Source software*.

EDOS is a research project funded by the European Commission as a STREP project under the IST activities of the 6th Framework Programme. The project involves universities - Paris 7, Tel Aviv, Zurich and Geneva Universities -, research institutes - INRIA - and private companies - Caixa Magica, Nexedi, Nuxeo, Edge-IT and CSP Torino.

The project aims to study and solve problems associated with the production, management and distribution of open source software packages.

EDOS is creating a new generation of methodologies, theoretical models, quality models and testing framework specifically tailored to opensource software (OSS) engineering and to P2P distribution of code and binaries over the Internet.

In this Poster we introduce EDOS, the work developed and the results.

## Goals

EDOS wants to propose innovative solutions for dealing with three key processes at stake in the production of a Linux distribution: (i) software dependency management, (ii) automated and manual testing, (iii) code and binaries distribution over the Internet. The expected benefit is a dramatic increase of the engineering productivity of Linux editors.

EDOS address packages in the following perspectives:

- *Dependencies*: Software packages may need other software packages to run, and often they don't tell exactly what other packages they need but leave a large room for choice. Also, some software packages cannot be installed at the same time. This makes the job of tools that automatically download required software packages difficult. Distribution maintainers want to make sure that there is always a way of selecting available packages to correctly install every piece of software they include, and that users can upgrade their systems without losing functionality.
- *Downloading*: Users need to download software packages from somewhere. That requires a lots of bandwidth and puts strains on mirrors that host those packages. This problem would be better solved with peer-to-peer methods.
- *Quality assurance*: The complexity of the quality assurance process increases exponentially with the number of packages and the number of platforms. To maintain the workload manageable, Linux distribution developers are forced to reduce system quality, reduce the number of packages, or accept long delays before final releases of high quality system.
- *Metrics*: Following the "release early, release often" philosophy, free and open source software is always in constant development and any serious project has many versions floating around : older but stable versions, and newer versions with new features but with more bugs. Free software can be of wildly varying quality. We are defining quality metrics, assessing their relevance and implementing them.

## Methodology

While the number of available opensource applications and the number of OSS contributors and users are growing at a rapid pace, the production of large scale opensource software such as a Linux distribution raises issues pertaining to (i) the management of software dependencies (ii) the process of automated or manual testing of thousands of applications (iii) the delivery of large piece of code and binaries over the Internet to millions of people.

The research conducted at European scale within EDOS project tackles these issues by bringing together on the one hand theoretical specialists in the fields of constraint programming and formal methods.

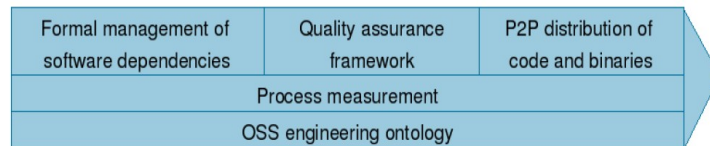


Fig. 2 – EDOS Layers

EDOS is organized in the following workpackages:

- **WP1 -Project Management**
- **WP2 - Formal Management of Software Dependencies:** The goal of this workpackage is to lay the foundations for a set of new generation tools to facilitate the task of gathering all the information about software packages and in particular allow to manage dependencies between them.
- **WP3 - Testing Framework and Quality Assurance Portal:** The goal of the workpackage is to research and experiment solutions which will ultimately allow to dramatically reduce the costs and delays of quality assurance in the process of building an industry grade custom GNU/Linux distribution or custom application comprising several. It will design, implement and experiment an integrated quality assurance framework based on code analysis and runtime tests, which operates at the system level.
- **WP4 - P2P Distribution Of Code and Binaries:** The goal of this work package is to investigate scalable and secure solutions to improving the process of distributing data (source code, binaries, documentation and meta-data) to end-users. The key issue in the code distribution process is the ability to transfer a large sized code base to a large number of people. In the case of Mandrakelinux, for instance, this entails copying a code base of 20 Gigabytes to a community containing up to 4 million users (i.e., the number of installed versions of Mandrakelinux). This community is growing so the problems have to be addressed. Currently the process is quite slow, as it takes 48 hours to copy from a master server to all mirror servers. This creates a latency problem that leads to inconsistencies at the user and developer side. This in turn can create awkward dependencies at the module level in future releases. This work package will test and evaluate two alternative architectures for data distribution that address the issue of latency and consistency.
- **WP5 - Metrics and Evaluation:** The goal of the project is to develop technology and products that will improve the efficiency of two key processes and one system. The two processes are the generation of a new version of a distribution from the previous version and the production of a customized distribution from an existing one. The system is the current inefficient mechanism of mirroring the Cooker data that needs to be replaced by a more efficient system. In the end, we want to demonstrate that we have indeed improved the processes and the system. Thus our goal is to define a set of metrics to measure the efficiency of the processes in question. These metrics will include man power as measured in man months and elapsed time.

The workpackages are integrated in the following architecture

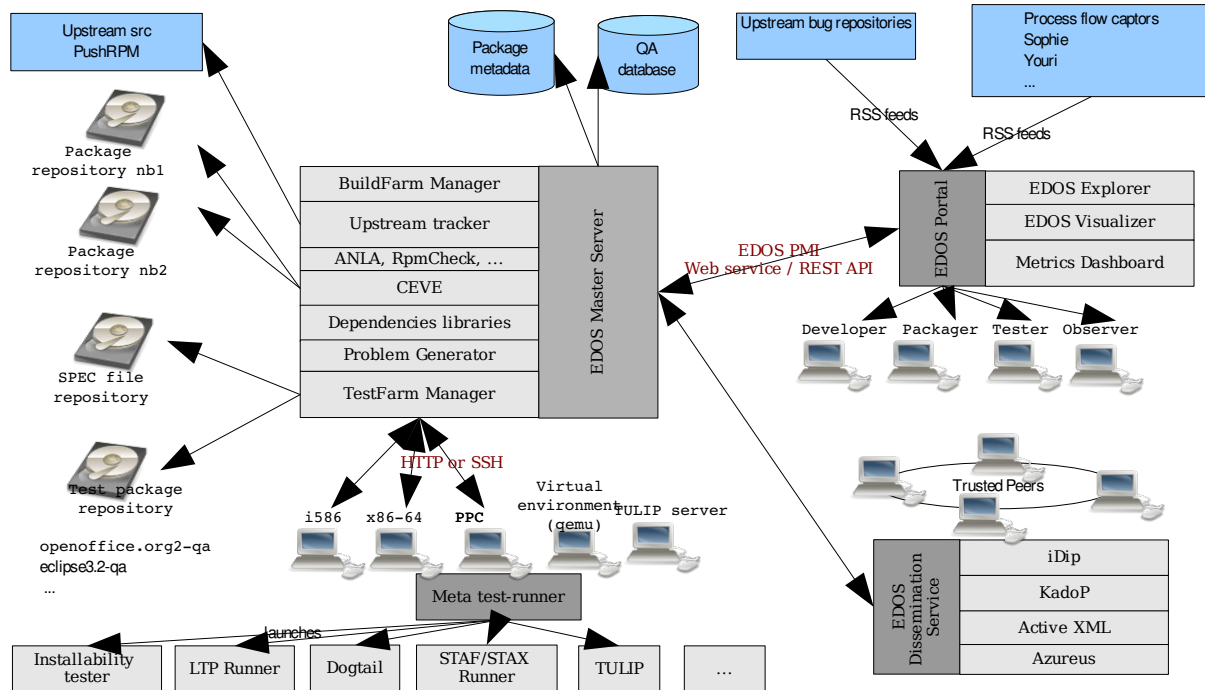


Fig. 2 – EDOS Architecture (Source: EDOS WP3 – François Deschelles)

## Results accomplished

Results were accomplished so far in different areas.

### Formal management of software dependencies:

EDOS Consortium has issued a formal proof that checking the installability of a single given package with respect to a given distribution is an NP-complete problem equivalent to SAT. A tool chain streamlining the management of package dependencies has been produced:

- C EVE is a package parser that outputs the dependency metadata in XML or SQL.
- EDOSLib provides a framework for describing and analysing dependency relationships among packages.
- C PProblemGenerator generates a constraint programming problem for the installability of the package. The problem can then be submitted to a CPP or SAT solver.
- edosdebcheck and edosrpmcheck are fast, efficient, integrated tools that allow to check for installability problem a full package repository.
- the history tool and the web exploration tool anla make it possible explore the evolution of packages in a repository over time, find explanation for installability problems, compute installation candidates, and more, including measuring the approximate SAT temperature of any Linux package, which is an indicator for spotting packages with intricate dependency webs.
- a publicly available “package weather service” has been running on the EDOS infrastructure for the Debian distribution
- Meta-installer enhancements like Apt-get rollback feature that allows the rollback of a previous apt operation.

### EDOS testing framework consists of:

- a generic XML format for describing test suites pertaining to the installation, update or functional testing of a set of packages
- a metatestrunner that takes XML test descriptions as input and that delegates the testrun to an underlying test runner

- TULIP: TULIP stands for “Testing Upgrades of Linux Images Program”. TULIP’s purpose is to drive upgrade tests of various Linux distributions to ensure both fine grained QA at the package level and testing the reliability of the upgrade result.
- a web portal taking over the workflow associated with the creation of tests, the submission of test reports and the display of associated metrics.
- Upstream tracker mechanism that allows the update of a RSS feed with the new versions released. A plugin for Trac had been published.

#### *EDOS P2P dissemination system*

EDOS team has designed a P2P architecture specifically tailored to the distribution of fast changing interdependent source code packages.

The system builds upon the KadoP infrastructure (Knowledge and Data in Peer to Peer) and on BitTorrent technology. KadoP relies on distributed hash tables technology, XML indexing and query optimisation techniques, and on the paradigm of ActiveXML documents, to enable the publication and efficient largescale querying of XMLcentred content in a P2P environment.

#### *EDOS metrics*

A transverse measurement effort is conducted for defining indicators of the production, the testing and the dissemination processes of a Linux distribution.

The indicator definitions are represented as two RDF schemas: MOAP and MOALD, respectively standing for “Metrics Of A Project” and “Metrics Of A Linux Distribution”. The objective of this effort is to create customizable dashboards providing live statistics on the pulse of opensource projects in general and of Linux distribution projects in particular.

The measurement effort also consists in designing the EDOS Project Management Interface. that models the OSS engineering processes. The PMI provides a sound foundation for a new kind of IDE specifically tailored to OSS engineering.

#### **Conclusions**

EDOS project will end in July 2007 and results are already available. More results are expected to be published meanwhile.

Joint projects of research teams and industrial partners can foster open source products, such as Linux distributions, in a unique way.

European open source software industry can take the lead in different fields and provide the market with valuable tools that provide efficiency, cost-saving and independence.