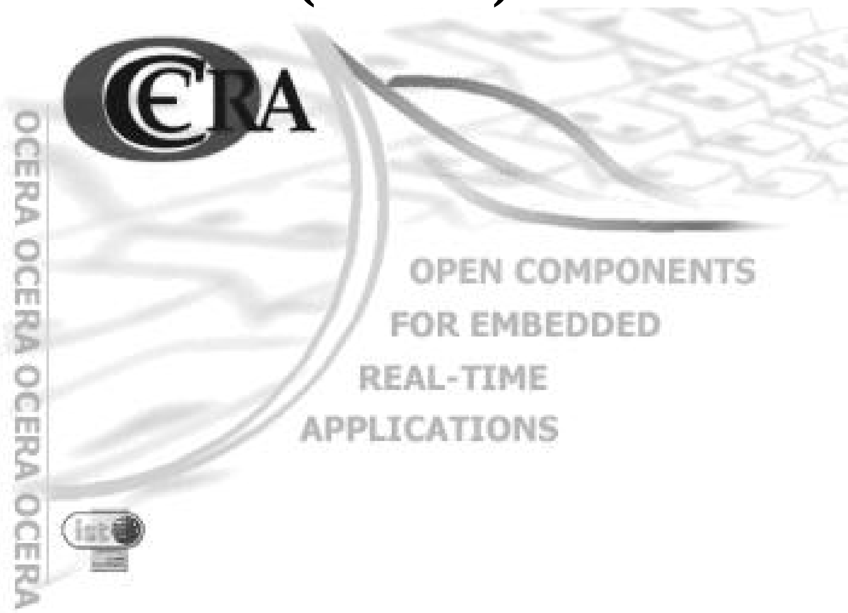
WP11 - Dissemination and Implementation (WWW)



D11.8 Exploitation plan (draft)

D11.8 Exploitation Plan (Draft)
by A. Crespo

Published March 2004

Table of Contents

Document Presentation

Project Coordinator

Organisation:	UPVLC
Responsible person:	Alfons Crespo
Address:	Camino Vera, 14, 46022 Valencia, Spain
Phone:	+34 963877576
Fax:	+34 963877576
Email:	alfons@disca.upv.es

Participant List

Role	Id.	Participant Name	Acronym	Country
CO	1	Universidad Politecnica de Valencia	UPVLC	E
CR	2	Scuola Superiore Santa Anna	SSSA	I
CR	3	Czech Technical University in Prague	CTU	CZ
CR	4	CEA/DRT/LIST/DTSI	CEA	FR
CR	5	Unicontrols	UC	CZ
CR	6	MNIS	MNIS	FR
CR	7	Visual Tools S.A.	VT	E

Document version

Release	Date	Reason of change
1_0	03/31/04	First release

Chapter 1. Introduction

This deliverable provides updates and revisions to the exploitation plan activities that were described in the Preliminary Exploitation Plan deliverable D11.5. After the completion of the second year of the OCERA project, the present document serves as controlling mechanism to the goals set by the OCERA consortium regarding the exploitation and dissemination of the project outcome. Additionally the deliverable contains an updated project status, namely the creation of the first OCERA prototype release that was concluded at the end of the second year.

As Open Source project, the exploitation is based more on the distribution of the development and the knowledge than sell the products.

This document describes both the Consortium exploitation plan and individual exploitation plans. This document is organised as follows: in the next chapter a review of the different products achieved in the OCERA project is detailed. Chapter 3 describes the Consortium exploitation plans and the individual pans of the each partner emphasizing the support after the project. Chapter 4 details the results of the dissemination through several internet access indicators.

Chapter 2. OCERA products typology

OCERA project was intended to produce components for the development of real-time embedded applications. Along the project the possible results of the project have been extended to fill additional requirements or to cover a more wide range of applications.

Although the second phase components are not included in the OCERA official release, from the purposes of the product description we will consider the first and second phase components. The products achieved in the OCERA project can be considered in different ways:

1. As a whole product: the OCERA kernel. It consists in the global project and it is distributed as a tarball including all the OCERA development oriented to the building of embedded systems for a wide range of applications: hard, soft and hard+soft real-time systems. This tarball also includes tools to configure the different options of the kernels to build the embedded system.
2. OCERA specific components. In general most of the components are very general (using POSIX system call interface) so, they can be used outside of the OCERA kernels and, from this point of view, can be integrated in other environments. On the other hand, some components are integrated in the RTLinux executive or Linux kernel and its usability and integration on other platforms is more difficult.
3. Additional OCERA components, like Gnat porting, permits the use of languages, like Ada, in RTLinux. These components add more possibilities to OCERA users to use other languages and tools when using OCERA.
4. OCERA Portings. This set of components permit the use of the OCERA developments when using other platforms (PowerPC and strongARM) than i386.
5. OCERA tools. A set of tools have been developed in order to give support during the design and validation to the OCERA components.
6. OCERA Documentation, training and case studies. This is an important result of the project which will produce a CDROM with all the project documentation, training activities and case studies using different approaches.

2.1. OCERA kernel

The OCERA kernel corresponds to the integration of the basic patches in the Linux kernel to be used as soft real-time kernel and the RTLinux executive with the inclusion of several basic components developed in the OCERA project. It is distributed as a tarball including the following parts:

- **kernel**: includes the distribution of Linux and RTLinux.
  1. The Linux kernel tree permits to standardize the development within OCERA and avoid integration problems as much as possible. Moreover, it provides a base kernel with the needed patches already applied (BigPhysArea, Low-Latency and Preemption patches) in order to solve the conflicts.
  2. RT-Linux Tree uses the same criteria than the Linux tree. The RTLinux-3.2pre1 is the basic version distributed in this product. RTLinux-3.2pre1distribution includes several OCERA components (POSIX Timers, POSIX Tracing, TLSF ....).
- **components**: includes the components code to be used in the application development. Components are organised in four basic directories: scheduling, quality

of service, communication and fault tolerance. Each component included examples, documentation and make files.

- **scripts**: contents the configuration files to build the target with the user selections.
- **applications**: offers the code of the developed applications in the OCERA project.
- **emdebsys**: It containts the embedded debian system distribution that can be used with the OCERA configuration to build a target.
- **stand-alone**: It contents the sources, examples and documentation to develop applications based on the stand-alone RTLinux.

At this moment, end of the second year, the OCERA kernel is based on the Linux version 2.4.18. New upgrades to next versions will maintain this structure. The RTLinux version is the rtlinux-3.2pre1, next releases will be included.

2.2. OCERA specific components

One of the efforts did by the Consortium was to produce components which can be used under different operating systems offering a POSIX interface. So, there is a high number of components at both levels (Linux and RTLinux) that can be used in different platforms. Some of these components have strong dependencies with other OCERA components at the kernel level. This is the case of those components which need full POSIX support. In this case, these components only can be used in other environment if it implements the POSIX services.

A list of components attending to its dependencies, with some examples, is detailed.

Independent POSIX components are components that do not need to use other OCERA components

- LinCAN: CAN driver configurable for Linux or Linux/RT-Linux. It is independent under Linux but uses malloc if it is compiled as a RTLinux thread/driver.
- ORTE: OCERA real-time ethernet for Linux, Rt-Linux, Windows, etc. It is independent under Linux but will use the RT UDP stack when it is ported to the hard real-time environment.

Components that need other external patches and Linux modifications

- Feedback scheduler

Components that need OCERA kernel adaptation

- FT components are deeply dependent on other OCERA components but do not modify the kernel by itselves, so the code is portable. The current implementation of facilities are located at two different levels :
  - Soft RT Linux OCERA (Redundancy Management) It depends on ORTE component
  - Hard RT Linux OCERA (Degraded Mode Management) It depends on LPTrace and basic POSIX components.
- Application Defined Scheduling (ADS) component requires to adapt the RT-Linux kernel to provide support to the external scheduling.

2.3. Additional OCERA components

The efforts on the OCERA project were not only in the direction of to develop components to be used in the OCERA framework but four other lines were drawn:

1. Porting the OCERA components to other hardware platforms. There exists several Linux portings to PowerPC and StrongARM, so small efforts were required to use them. It was not the case of the RTLinux. There exist a initial version for strongARM but not for PowerPC. In this case, ...

2. Porting other tools to RTLinux. Taking profit of the more complete POSIX interface of RTLinux achieved in OCERA, some compilers were adapted to produce code to be executed on RTLinux.
   1. GNAT Ada compiler has been adapted to produce code to be executed on RTLinux. In our approach, we maintain the RTLinux layer, which offers a POSIX.13 sub-set and we build Ada applications to be executed on top of RTLinux only by slightly modifying the GNAT run-time system and by adding the functionality needed to allow the execution of the Ada program as a Linux module.
   2. Virtual Java Machine. In order to evaluate the use of the dynamic memory allocator and the porting premises achieved in the RTLGnat, we have ported with small efforts a Virtual Java Machine on top of RTLinux. additional efforts are needed to offer a minimum quality porting of this JamVM. JamVM is a new Java Virtual Machine which conforms to the JVM specification.
   3. C++ support. Support for programs written in C++ requires dynamic memory management and other additional features that can be, with not too much effort, developed.

3. Exporting OCERA components to other environments: Several components have been included, at this moment, in other environments.
   1. The Dynamic Memory Allocator (TLSF) (DIDMA in previous versions) and POSIX Tracing have has been included in MARTE OS [MARTE OS] and the RTLinux-GPL distribution rtlinux-3.2pre1.
   2. The TLSF is considered to be included in the Linux 2.6 ?????
   3. Several OCERA POSIX components Signals, Timers, Message Queues, Barriers have been included in the RTLinux-GPL distribution rtlinux-3.2pre1.
   4. LinCAN (CAN driver) is fully utilizable in Linux kernels from 2.2.x, 2.4.x and 2.6.x series and RT-Linux version 3.2 or OCERA RT-Linux enabled system is required for hard real-time use. The following cards are supported:
      - Advantech PC-104 PCM3680 dual channel board
      - PiKRON ISA card
      - BfaD DIMM PC card
      - KVASER PCIcan-Q, PCIcan-D, PCIcan-S
      - KVASER PCcan-Q, PCcan-D, PCcan-S, Pccan-F
      - MPL pip5 and pip6
      - NSI PC-104 board CAN104
      - Contemporary Controls PC-104 board CAN104
      - Arcom Control Systems PC-104 board AIM104CAN

- IXXAT ISA board PC-I03
- SECO PC-104 board M436
- Board support template sources for yet unsupported hardware
- Virtual board

5. ORTE has been ported and tested on multiple platforms: Linux, RTLinux, Windows platforms (MinGW and MSCV), RTAI (RT-NET TCP/IP) and compatible with RTI NDDS. (RTLinux version requires RTL_UDP and RW-locks components)

4. Extending the possibilities of the OCERA components. MARTE OS has been modified to be included as a module in substitution of RTLinux. So, MARTE OS can be used as real-time executive in an OCERA hard and soft real-time system.

2.4. OCERA tools

In addition to the OCERA components, several tools have been developed to help users in the design and testing of the OCERA components.

In this category we can find:

- FTBuilder : this is a design & build tool that has been developed to help the user to build its fault-tolerant application. The FTBuilder is written in TCL/TK (version 8.3) and is totally independent of any other component.
- UniCAP is a tool developed by UniControls. It is a complex software tool for support of designing and application programming of distributed information and control systems. (See description of UniCAP in the Annex A).
  It consists of two main parts:
  1. Development environment for application programming.
  2. Real-time environment in target process control stations
- Verification of distributed systems using timed automata developed by CTU is a tool that permits the temporal properties of a discrete event system focusing on CAN models.
- Soft PLC (Programmable Logical Controller), compliant to IEC 61131-3 (Programming Languages for PLC), was implemented by CTU. It was adopted by FSMLabs as typical application of RT Linux
- Stand-alone tools are a set of programs that allows the debugging and tracing of embedded systems based on the stand-alone development.

2.5. OCERA documentation, training and case studies

OCERA project documentation will offer the possibility of training to the future users and developers. To support training, we will develop a training documentation and CD-ROM containing the Open Source deliverables and the documentation. The preparation of the training documentation will consist in documentation rewriting of each module to create the training documentation, to add new components not developed by OCERA, and to gather the development tools and utilities used for development.

All the documentation will be merged into a USER'S GUIDE and a PROGRAMMER'S GUIDE. Each component developer being responsible of the PROGRAMMER'S REFERENCE MANUAL and MANUAL PAGES (UNIX way) for the developed components. The training documentation will consist in different case studies to show the use and facilities of the OCERA components. Each case study will contain assignment, solution and its explanation/discussion.

The creation of the training/example CD-ROM will be done using the Documentation and development tools gathered in Training and Technical Support work-package , and the Source Code of the up-to-date version of the software. The CD-ROM will also include an installable version of the software with RTLinux and the developments tools, with the goal to facilitate the installation of a RTLinux/Ocera development system and/or the installation of a minimal set of components for embedded systems or tiny systems. The component installation will be driven by an installation software. We will also design a CD-ROM jacquet and documentation to make the Ocera CD-ROM and documentation attractive and easy to use and to read.

The CDROM will allow to install a debian 3.0r1 on a PC with all the development tools that are necessary for developing RT application above OCERA, compiling OCERA, using ADA, cross compile for ARM and POWERPC.

This CDROM will be available both as ISO image on Internet and as a physical CDROM for distributing within workshops, sending to research groups or technical software industry departments or sell through technicals bookshops.

Chapter 3. Exploitation plans

In this chapter we describe the actions planned for the exploitation of the OCERA results. These actions are described considering the Consortium and the specific plans of each partner and will consider the targeted groups for dissemination.

3.1. Consortium exploitation

OCERA results are disseminated in different forum to promote its knowledge and use. For what concerns to the dissemination and exploitation of the results of the OCERA project, several mechanisms have been to be adopted by the members of this consortium.

OCERA results are disseminating through:

- Paper publication in workshop, conferences and seminars on the subjects of real-time systems, operating systems and open-source forums. Until now, papers related with OCERA project have been published in the following events:
    - **Real-time Community:** IEEE Real-Time Systems Symposium, IEEE Euromicro Conference on Real-Time Systems, International Conference on Reliable Technologies Ada-Europe, IEEE Conference on Real-Time Computing Systems and Applications, IFAC Workshop on Real-Time Programming, Control and Decision Conference CDC2003.
    - **Embedded Systems:** IEEE Real-Time and Embedded Technology and Applications Symposium,
    - **Linux Community:** Real-Time Linux Workshop, Linux Devices,
    - **EC Events:** Participating in FP6 Projects
- Organization of international and national events (workshops, conferences and seminars) in the real-time field. The last edition of the Real-Time Linux Workshop was hosted in Valencia and sponsored by OCERA project.
- Promotion of OCERA results and integration of these results in other platforms. Several activities have been carried out to incorporate OCERA specific components in other environments:
    - Real-Time Linux-GPL is the major receptor of these OCERA components. Several components are integrated in the last distribution release (TLSF dynamic memory allocator, POSIX components, etc).
    - MARTE OS: MARTE OS has adopted TLSF as dynamic memory allocator and has also included in the last version some components as: RT_UDP. Additionally, the posting of MARTE OS as Linux module is being used by several MARTE OS users.
    - SHaRK operating system has included some OCERA components as the dynamic memory allocator.
    - RTAI is being porting some OCERA components. They have shown a high interest in the adaptation of more components.
    - Communications components are being ported to different Linux versions and are distributed as independent components.
- Initiatives of trainings intended for the acquisition of the competence on embedded Linux

- OCERA web site: One of the major aspects in dissemination is OCERA web. In section 4, we show the OCERA web site traffic and the OCERA project in search engines results.

3.2. Partner exploitation

Moreover the global exploitation of OCERA results, each partner is disseminating its results considering its activity. Also each partner has defined a specific exploitation plan.

3.2.1 UPVLC

The exploitation plan of the UPVLC consists in the following activities:

- UPVLC is currently the web site maintainer of the OCERA web site. We plan to follow this activity after the OCERA project finalisation.
- The RTLinux-GPL version is allocated in our servers and we coordinate the version upgrading and releasing. In this line, our purpose is to continue including OCERA components in the distribution and improving them.
- As part of our research activity, we plan to work after the project in the project topics. The components developed by UPVLC are the core of our research interest. So, new improvements on the OCERA topics will be included in future versions.
- In conjunction with MNIS, SSSA and CTU, we plan to release future OCERA versions after the end of the project.
- In order to promote the OCERA results in the industry, a Spin-off has been created with UPVLC people involved in the OCERA project. The company is called OS3 (http://www.os3sl.com) and it is oriented to develop industrial applications in the area of Spain and Portugal. Valencia-based OS3 is already working on several real-time Linux projects and has extensive RTLinux experience, as its staff are core developers of the RTLinux GPL version and are Ocera participants. The people involved in this spin-off are Pau Mendoza, Josep Vidal and Alejandro .
- Dissemination through publications is one of our main interest.

3.2.2 SSSA

The exploitation plan of the SSSA consists in the following activities:

- Technical publications
- The IRIS bandwidth server is going to be exploited porting it to other platforms.
- Feedback scheduling in OCERA is very general and can be used in other environments.
- In conjunction with MNIS, SSSA and UPVLC, we plan to collaborate in future OCERA versions after the end of the project.

3.2.3 CTU

The exploitation plan of the CTU consists in the following activities:

- Communication components have been developed to be used not only in the OCERA framework but in several platforms. The integration of communication components in several platforms will continue after the end of the project.

- In conjunction with MNIS, SSSA and UPVLC, we plan to collaborate in future OCERA versions after the end of the project.
- Publications
- CTU intends to find national projects and industrial contracts to continue development of both open source and proprietary communication stacks for RT Ethernet and CAN. These two protocols represent middle term aims in the area of the system software development. Further CTU intends to focus on the development of tools for embedded systems (e.g. Processor Expert distributed by Motorola and implemented by UNIS, Czech SME). CTU contribution will be in the development of beans/components for standardized communication protocols and control systems.
- CTU as a leader of Embedded Systems group in new national center of excellence CAK (50 full time employees, see www.c-a-k.cz) became a centric contact point in the Czech Republic, joining both universities (CTU Prague, UTIA Prague, VSB Ostrava, VUT Brno) and companies (Unicontrols, UNIS, Siemens AVG) in the area. The objective is to foster competition among PhD. students, young researchers and developers; to generate interesting scientific topics originated in the application area; and to collaborate on application projects. As a part of follow up activities we intend to continue in the development of demonstrators (mobile robot, helicopter model, laser game, aquariums, …) to attract new students interested in the embedded systems. OCERA architecture is a point of departure for this development work.
- Scientific work will concentrate on verification and optimization in general (specifically on timed automata, time Petri Nets, static scheduling for FPGAs, EDF with communications and hybrid Petri Nets). OCERA framework has proven its constructive role in defense/argumentation/rejection of these research topics.

3.2.4 CEA

The exploitation plan is described considering the different product lines:

Fault-tolerance components

The maintenance of the fault-tolerance components will involve the debugging of the WP6 components will be insured during one year after the end of the project and compatibility with future version of OCERA will be insured.

Also, fault_tolerance is a very large domain and further development of such facilities has multiple implications in many different directions such as :

- design and verification tools,
- code generation and debugging environments,
- basic middleware building blocks,
- fault-tolerant communications,...

This would require the constitution of a consortium to pursue the work started.

Among the plans for further exploitation of OCERA results. It is intend to work in collaboration with a CEA team working on UML design of real-time applications.

The objective is to enter a fault-tolerant profile within a UML CASE tool that will produce code for fault-tolerant applications using an adaptation of the fault-tolerance model developed in OCERA.

This team is currently participating in a new open project for UML modeling named TOPModl.

CEA Exploitation plans for other results of OCERA

CEA-LLSP is a research laboratory that develops design environments for real-time embedded applications. Research areas :

- Modeling of concurrent real-time systems (soft and safety critical)
- Analysis and verification of real_time application models
- Code generation and compilation techniques
- Specific middleware supporting high level programming model (real-time active objects)
- Execution environments (Virtual machine)

In this context we have developed the ACCORD/UML environment that provides support for modeling/design/code of real-time embedded applications. The current target systems are UNIX, LINUX, VxWorks for mono machine applications

Among future directions are :

- Management of deployment and distribution
- Introduction of fault-tolerance management
- Optimized code generation for target architectures dedicated to embedded systems.

Among OCERA results and apart from fault-tolerance, two main axes can thus be foreseen for further exploitation:

- ORTE components will be tested as a possible component for distributed ACCORD/UML environments
- Standalone OCERA RTLINUX could be a target environment

Commercial exploitation

The CEA-List robotics laboratory develops and distributes a generic robot control software named TAO2000. TA02000 is the core software used in several industrial robotic products. It is designed and distributed by industrial partners of CEA-List such as Haption (www.haption.com). As a CEA spin-off, Haption benefits from CEA technology transfers, hence from a know-how of thirty years of research in force-feedback systems.

From the very beginning of the project, Haption expressed a strong interest towards Ocera developments. TAO2000 is now based on the VxWorks Real-Time Operating System from WindRiver, which involves high licensing costs from both the development point of view (VxWorks development license cost is about 10k Euros for a single target) and the final product point of view (VxWorks runtime license cost is about 1k Euros, billed to the end-user).

To reduce both development and final product costs, Haption intends to use RTLinux and Ocera developments as the new operating system for next controller generation of its leading product, the Virtuose haptic device. As a first step towards this direction, Ocera WorkPackage 9 (validation on platform - robotic application) should present the feasibility of porting the current Virtuose software from VxWorks to RTLinux/Ocera. As a second step, Haption will resume and enhance the Ocera project robotic application in order to release an industrial software by the end of 2005. The VxWorks and the RTLinux/Ocera software versions will be merged into a single source code for further maintenance and upgrades. Distribution of the Virtuose product using the RTLinux/Ocera software will depend on licensing and patenting issues, however. Haption expects a fast evolution of RTLinux low-level layers so that it can circumvent the FSMlabs patent, which introduces strong uncertainties regarding overall application source code protection.

The CEA-List electronics laboratory designs and develops hardware devices dedicated to robotic products, such as intelligent I/O cards for low-level robot control. Such devices are being used in the robotic products (such as the Virtuose hardware controller) of several industrial partners of CEA-List. Having invested much effort in deported control architecture for several years, the electronics laboratory expresses a strong interest for the RTLinux Standalone component. This should allow deep enhancements of low-level control capabilities of intelligent I/O devices, and also high-level control directly performed by the device, instead of a dedicated controller/computer. Such device requires real-time capabilities from the OS that a standard embedded linux kernel cannot provide. Current products are based on hardware solutions (Digital Signal Processors) that do not offer a lot of flexibility and which strongly constraints the device features. The electronics laboratory at CEA-List expects a lot from the Standalone RTLinux component and intends to develop a hardware device based on it. This device should integrate an ARM processor based architecture with an embedded Standalone RTLinux to allow both low-level and high-level control of a 1 to 3 axis robot device connected to it (we assume that ARM should not allow a more than 3 axis high-level control, considering first analysis). An example of such kind of 3-axis robot device is now in development at CEA-List for the MUVII IST project (IST-2000-28463, http://www.hpclab.ceid.upatras.gr/muvii). Feasibility tests of such architecture should start in the middle of 2005 and first prototypes are expected by the end of 2006. The device should feature USB2 connectivity allowing deported control and compatibility with almost any computer system.

3.2.5 Unicontrols

UniControls has developed the UniCAP tool which is a complex software tool for support of designing and application programming of distributed information and control systems.

UniCAP was used as a design tool and as a real-time application in target process control stations in following applications:

1. The high-pressure Gas Turbocompressor Plant Kralice
   Technology: High pressure gas transport in the main gas pipe line from Russia to Germany
   Size of distributed system: 6 nodes
   Location: the Czech Republic, South Moravia
2. The high-pressure Gas Turbocompressor Plant Kourim
   Technology: High pressure gas transport in the main gas pipe line from Russia to Germany
   Size of distributed system: 6 nodes
   Location: the Czech Republic, Central Bohemia
3. The high-pressure Gas Compressor Plant Jablonov
   Technology: High pressure gas transport in the main gas pipe line from Russia to Germany
   Size of distributed system: 14 nodes
   Location: Slovakia

4. Underground Gas Storage Tvrdonice
   Technology: Seasoner natural underground gas storage
   Size of distributed system: 73 nodes
   Location: the Czech Republic, South Moravia

Plan of commercial exploitation

The UniControls exploitation plan for the OCERA results is:

1. Operating systems Linux+RTLinux should completely replace OS9 as base RTOS in UniControls for applications in industry from 10/2004.
2. Savings are planed for development tools and for licence fees. Our estimations is to produce about 70 nodes per year and to save about 400 EUR per node for license fees.
3. Linux+RTLinux enables to produce critical applications because source codes of the operating systems are available, which is usual requirement of certification authorities (Office for Avionics, Office for Rails, Office for nuclear safety...)
4. Better portability of applications (POSIX standard) should bring indirect savings.
5. UniControls plans to use UniCAP for applications in industry and to sell UniCAP including documentation and trainings as a complex tool for support of designing and application programming of distributed information and control systems.

3.2.6 MNIS

The exploitation plans of MNIS are centered in the dissemination of the OCERA project. The base for the dissemination is the CDROM and the case studies developed in WP10.

Moreover, MNIS will collaborate on the OCERA distribution maintenance and technical support.MNIS would also need the same sort of collaboration agreement with the partners developing the specific components, ORTE, LINCAN, FT, may be on a peer to peer basis through the usage may or may not be included in the development of an application.

Commercial exploitation plans

MNIS expects to commercially exploit the results of the OCERA project by providing technical support to industrial firms wanting to use OCERA kernel or some OCERA components in embedded systems or in control command systems.

To provide this support, MNIS, is already involved in the training and technical support package as workpackage leader and rely on a 5 years experience in providing RTLinux CDROM and support in France.

MNIS wants to focus on the European market, especially telecommunication and aeronautics market.

The training and technical support provided will be of two levels:

1st level is a direct intervention of MNIS for developing applications above OCERA or give training and courses to the client.

2nd level is to provide the clients with the possibility of adapting OCERA to its exact needs by using the services of the components developers.

The Universities involved in the OCERA project will provide the last level support. They will provide the ultimate expertise on the kernel and components in case of: new porting, adaptations, new features. A service and commercial agreement must for this be found between the different partners.

3.2.7 VisualTools

Visual Tools commercial exploitation of the project results will be based, mainly, on the OCERA resource reservation components and the OCERA embedded system development and deployment environment. As we pointed out in other contexts, Visual Tools product requirements are of a soft real-time nature.

Visual Tools expects to include OCERA resource reservation components into its custom GNU/Linux kernel so we can guarantee the appropriate behavior of the system in different load conditions. As videosurveillance systems, we must be able to guarantee the customer that alarm situations will be handled correctly, the corresponding images per second recorded in HD or transmitted through different media, according to configuration. Particularly, the CBS component permit us to be sure that alarms will be handled as expected even in overload conditions such as users connected for live video steaming, backup processes active, etc...

Visual Tools is exploring also low cost platforms based on the Xscale processor (to which OCERA resource reservation components are ported). In these low-cost systems it is even more difficult to guarantee the user that the correct actions will be taken in case a alarm is fired. Traditionally we have guaranteed such behaviour through a worst case analysis, wasting processing power "for if an alarm fires". The dynamic programmability of resource management component will permit us to profit XScale processing capabilities in normal conditions. We will be able to let more users to connect for live video streaming, for example, and in case an alarm is fired, program the CBS server to reserve the needed CPU bandwidth to handle the alarm.

On the control center side, we have other particular needs. The control center can receive several alarms and video streams at the same time, each video stream having different framerates and compression quality characteristics. For what alarms concern we can still make use of CBS to assure the customer proper alarm handling, but the live video streaming functionality is more complex; ideally we should be able to assign different CPU bandwidth to each stream depending on its needs. The feedback scheduling component fits perfectly in this case because it will give more CPU bandwidth to the thread/process that needs it depending on its stream characteristics (framerate and quality).

There are other cases in which Visual Tools could benefit from OCERA components. Sometimes our systems must be connected to other equipments with hard real-time constraints, e.g. connection to an on-board computer in a train or an specific bus. The integration of hard and soft real-time systems achieved in OCERA allows us to develop on the same GNU/Linux kernel and avoid the necessity of ad-hoc kernels or special hardware.

Finally Visual Tools will make use of the development and deployment environment offered by OCERA to produce our embedded systems, whether based on XScale or x86 processors.

Chapter 4. Dissemination indicators

The OCERA web site was designed for dissemination purposes, so the success of this goal can be, at this moment, evaluated in terms of number of accesses to the different parts of the web. Of course, this is not a criteria for the evaluation of the project, it is only a indicator of the success or not of the dissemination process using the web. We concentrate the analysis in the last year period (2^{nd} project year April 2003 - March 2004).

Several indicators can be used to analyze the results:

- General statistics
- Number of visitors
- Geographical location for the visitors
- Hits par week
- Number of downloads and Most requested files
- OCERA in Sourceforge
- OCERA in Goggle

4.1. General statistics

This is a summary of the web reporting.

Item	Value
Hits	321653
Total Data Transferred	6.64 gigabytes
Total Visiting Users	49813
Time Period	March 30, 2003, 01:31 AM to April 05, 2004, 07:03 AM
Average Hits per User	6.46
Average Users per Day	133.55
Average Data Transferred per Day	18.22 megabytes
Hits cached by Client	32092 (9.98%)
Report generated on	April 05, 2004 at 12:05 PM
Incomplete downloads/file requests	10793 (3.36%)
Log spans a period of	373 days
Total failed requests	13955 (4.34%)
Unique IP Addresses	12604
Average Data Transferred per User	139.68 kilobytes
Average Hits per Day	862.34
Average Data Transferred per Hit	21.63 kilobytes
Each user has visited approximately	3.95 times

Hits on Pages	109255
Hits on Files	179711
Hits on Images	18732

4.2. Number of visitors

The most relevant information is the number of visiting users (49.813 users) and the average hits par day (133.55). The number of unique IP addresses is lower (12.604 IP addresses) can be consequence of the use of a proxy.

With respect to the people who has revisited the web after the first visit, it is significant that the number of times that each user has visited the web is approximately 2.33.

4.3. Geographical location for the visitors

The following table shows the domain distribution of the users. Only domains with a number of hist greater than 100 has been listed.

Domain Name	Hits	Percentage
Commercial (.com)	91997	36.98%
Network (.net)	43698	17.57%
Spain (.es)	27959	11.24%
Educational (.edu)	16537	6.65%
France (.fr)	11871	4.77%
Italy (.it)	7081	2.85%
Germany (.de)	5188	2.09%
Poland (.pl)	4838	1.94%
Venezuela (.ve)	3874	1.56%
Czech Republic (.cz)	3031	1.22%

4.4. Number of downloads and Most requested files

OCERA deliverables and components can be downloaded from several different sources: OCERA web site (www.ocera.org), Sourceforge web site (www.sourceforge.org/ocera) and RTPortal (rtportal.disca.upv.es) where the UPVLC offers the unstable developments.

To detect which are the most interested documents downloaded, next table shows the list of the **Deliverables** sorted by the number of hits.

File Name	Hits
/archive/deliverables/ms1-month6/WP1/D1.1.pdf	455
/archive/deliverables/ms1-month6/WP2/D2.1.pdf	411
/archive/upvlc/public/reports/architecture/Arch_db.pdf	332

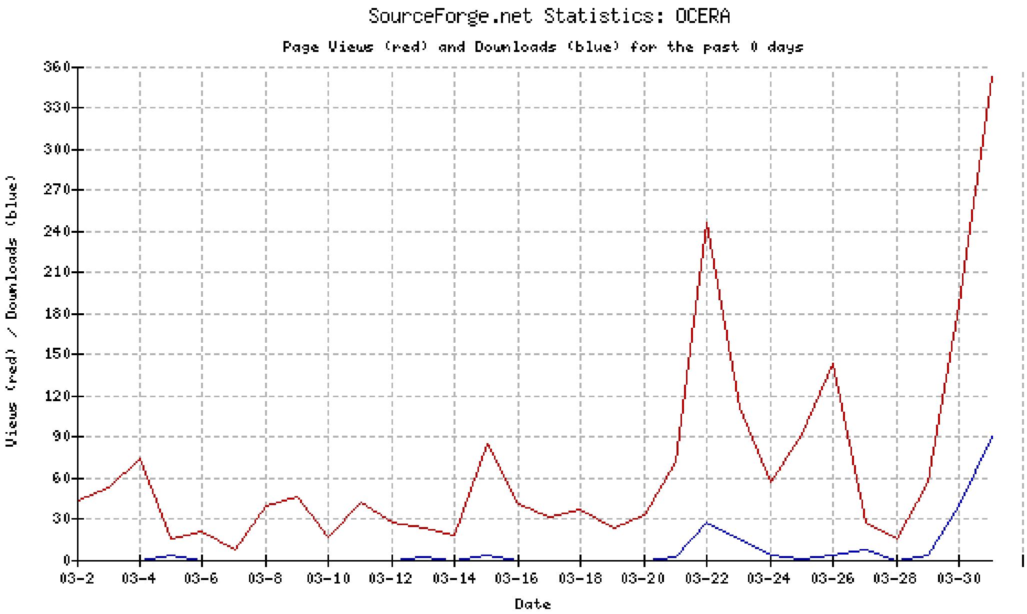
/archive/deliverables/ms1-month6/WP3/D3.2_Not_in_Open_RTOS/D3.2.pdf	325
/archive/deliverables/ms2-month12/WP5/D5.1.pdf	299
/archive/deliverables/ms2-month12/WP9/D9pc1.pdf	203
/archive/deliverables/ms1-month6/WP3/D3.1_Feedback/D3.1.pdf	201
/archive/deliverables/ms2-month12/WP7/D7.1.pdf	195
/archive/deliverables/ms1-month6/WP3/D3.3_New_Approaches/D3.3.pdf	192
/archive/deliverables/ms1-month6/WP11/D11.1/D11.1.pdf	161
/archive/deliverables/ms2-month12/WP9/D9rb1.pdf	155
/archive/deliverables/ms2-month12/WP4/D4.1.pdf	153
/archive/deliverables/ms2-month12/WP9/D9mm1.pdf	134
/archive/deliverables/ms2-month12/WP5/D5.2_rep.pdf	134
/archive/deliverables/ms1-month6/WP13/D13.1_1/D13.1_1.pdf	134
/archive/upvlc/public/reports/whitepaper/whitepaper.pdf	108
/archive/deliverables/ms1-month6/WP12/D12.1_Assesment/D12.1.pdf	105
/archive/deliverables/ms2-month12/WP10/D10.2.pdf	99
/archive/deliverables/ms2-month12/WP4/D4.2_rep.pdf	88
/archive/deliverables/ms2-month12/WP7/D7.2_rep.pdf	86
/archive/deliverables/ms2-month12/WP10/D10.1.pdf	71
/archive/deliverables/ms1-month6/WP11/D11.5/D11.5.pdf	67
/archive/upvlc/public/components/ptrace/ptrace-1.0/doc/rtl-pt1.0.pdf	66
/archive/deliverables/ms1-month6/WP11/D11.2/D11.2.pdf	56
/archive/deliverables/ms2-month12/WP6/D6.1.pdf	54
/archive/upvlc/public/components/ptrace/ptrace-1.0/doc/posixtrace.pdf	53
/archive/deliverables/ms2-month12/WP6/D6.2_rep.pdf	52
/archive/deliverables/ms2-month12/WP8/D8.1.pdf	50

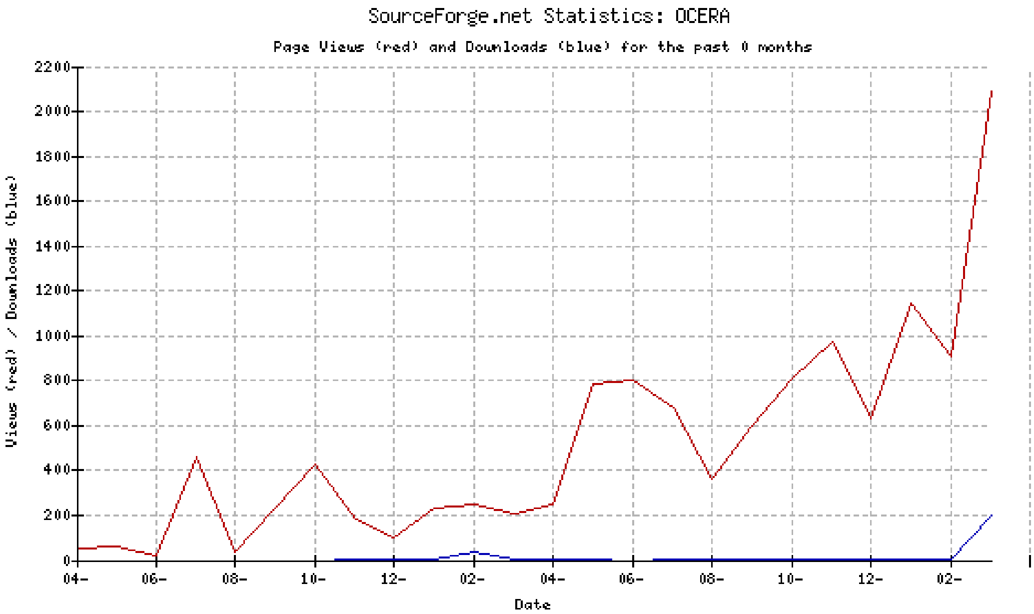
With respect to the components, next table shows the number of downloads of some components that are distributed in a independent way. Other components (as POSIX Timers, POSIX Signals, etc) are included in the new release of the GPL RT Linux.

File Name	Downloads
RTLGNAT 3.14	936
RTLGNAT 3.15	868
Stand-alone	626
POSIX Timers	612
POSIX Signals	611
Dynamic Allocator	245
Application scheduler	124

Sourceforge statistics

The page views and downloads for the last month are shown in the next graphic.





The same information from the beginning of the project is shown in the next figure.

4.5. OCERA in Google

One important indicator is the visibility of the project through the Google motor. Several search strings have been searched in Google in order to look the position to any OCERA reference. A search of "OCERA" in google produces a total of **4,890** references.

Next table shows some of the results of the search using general or specific string.

Search Pattern	Position	Number of references
Scheduling components real-time	1	268000
"Resource management" components real-time	1	105000
Fault tolerance components	1	262000
Communication components real-time	2	887000
POSIX Timers	1	24500
POSIX Signals	3	64200
Real-Time Components	20	3800000
Real-Time Components Embedded systems	14	494000
Memory allocator real-time	1	19200
RT-Ethernet	1	232
CanOpen real-time	18	4700
CBS scheduler	1	3500
Fault tolerance components	1	226000
Scheduling components real-time	1	268000

Date of this information: 05/04/2004

References:

[ACT] ACT-Europe [http://www.act-europe.fr/]

[Jaluna] http://www.jaluna.com/

[MARTE OS] http://marte.unican.es/

[JamVM] Java Virtual Machine http://jamvm.sourceforge.net/

[ShaRK] http://shark.sssup.it/

ANNEX A: UniCAP porting to Linux+RT-Linux environment

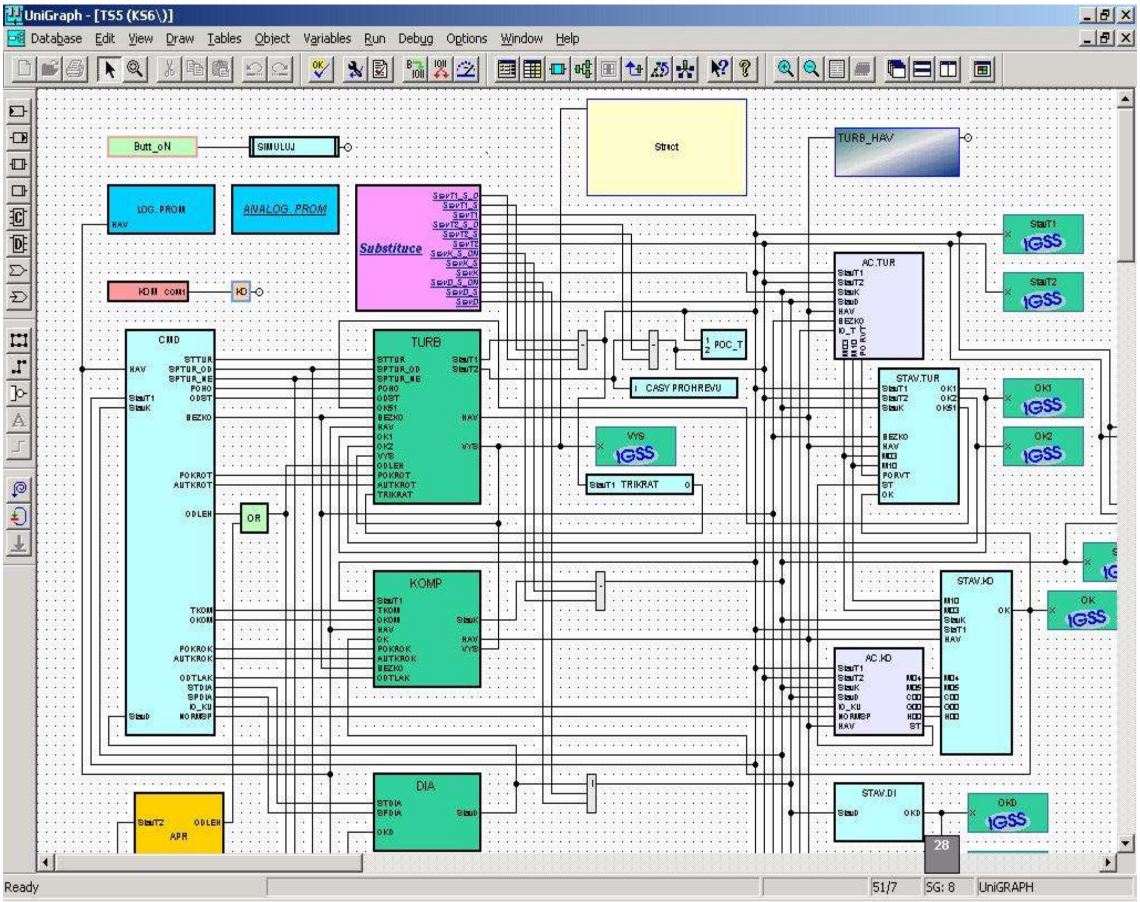
1. Introduction

UniCAP is a complex software tool for support of designing and application programming of distributed information and control systems.

It consists of two main parts:

1. Development environment for application programming. Application programmer can use a graphic editor for design an application in FBD or SFC languages.

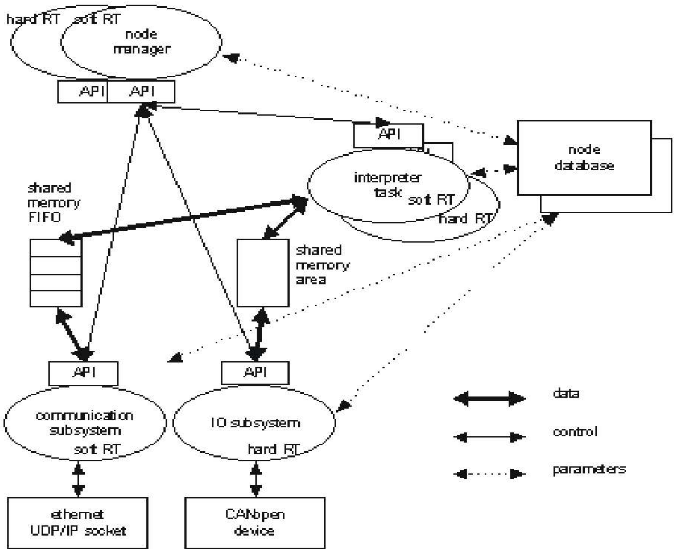
Main control algorithm of turbocompressor gas plant Kralice in FBD language:



2. Real-time environment in target process control stations

Control algorithms of process control applications are compiled, downloaded into process control stations (nodes of a distributed system) and executed in an environment of a real-time operating system. This part of UniCAP will be ported in frame of OCERA project from OS9 operating system to Linux+RT-Linux operating system.

2. Software architecture of UniCAP in proces control stations



Node manager

Node manager maintains all information regarding node as control system subject. Node database with interpreted code is maintained here and algorithm interpreter is controlled in terms start, stop interpretation, attach/release IO subsystem and communication subsystem. Configuration/reconfiguration of IO subsystem and ethernet communication subsystem is controlled from here. Node manager will contain hard RT task responsible for accurate node timing and soft RT task responsible for node management.

Control algorithm interpreter

Control algorithm interpreter uses code, prepared in node database, and performs actions as specified by this interpreted code. Interpreted code is prepared offline in development and debugging tool running on host PC and downloaded into target system. Application control is described in languages according to IEC 61131 and then compiled. Download of interpreted code is controlled by node manager. Control algorithm interpreter contains IO subsystem interface and communication subsystem interface. Interpreted code is divided into interpreted "tasks", which are represented with Linux tasks. For critical control algorithms are these tasks placed in hard RT, the rest is placed in soft RT.

Ethernet communication subsystem

Ethernet communication subsystem transports messages via ethernet network using UDP/IP sockets. Subsystem take care for message confirmation, timeout evaluation and communication retries. Subsystem keeps and maintains net topology information, which is obtained with broadcast messages among nodes in topology.

IO subsystem with CANopen communication

IO subsystem connects algorithm interpreter with remote IO modules connected to node with CAN bus and protocol CANopen. IO modules provides direct technology control. CANopen protocol is realized by CANopen device and connected to the subsystem with its API library. CANopen subsystem is configured according to node database and can act as CANopen master or slave, as a master can manage several CANopen remote IO nets with attached modules. Used nets and modules with appropriate parameters are specified in database.

3. Plan of porting to Linux+RT-Linux environment

Node manager

RT space placement: both hard RT and soft RT

SW interface: Node manager API

Control algorithm interpreter

RT space placement: both hard RT and soft RT
SW interface: API for interpretation control, Node databases with interpreted code

Ethernet communication subsystem

RT space placement: soft RT
SW interface: Communication subsystem API library

IO subsystem with CANopen communication

RT space placement: hard RT
SW interface: IO subsystem API library