[osmia rufus]

OSMIA

Title:	Open Source Medical Image Analysis
Acronym:	OSMIA
Date:	October 2003
Project Number:	IST-2001-34512
Deliverable:	D1.2 Project Review Report
Version:	1.1
Author(s)	All Consortium Members

Contents

1 Introduction

1.1 Objectives

The aim of this document is to provide a review of the OSMIA project (IST-2001-34512).

1.2 Scope

This document applies to all workpackages on the OSMIA project (IST-2001-34512) as described in the description of work.

1.3 Related documents

- OSMIA Description of work
- OSMIA Contract preparation forms (CPF)
- Work package Deliverables

1.4 Nomenclature

Within this document the following abbreviations have been used.

Abbreviation	**Member**
UoM	University of Manchester
DCU/VSL	Dublin City University/Vision System Labs
UoWO	University of Western Ontario
Voxar	Voxar Ltd.
EU	European Commission

2 Project Summary

The following is taken from the original proposal technical annex part B;

The aim of this project to provide an open source software development environment for medical image analysis research which facilitates the free and open exchange of ideas and techniques (dissemination) with a minimum of effort. An infrastructure within which long term research objectives can be achieved, co-operatively via incremental stages. Mechanisms will exist which enable clinical end users (non-developers) to access these state-of-the-art techniques at early stages, using interfaces appropriate to their level of expertise. These mechanisms will also provide commercially interested parties with the opportunity to rapidly assess the potential of research technologies for custom integration into applications, without the need to re-implement. This represents a co-operative model for open source and commercial developers to collaborate without restricting either community.

In the **Description of work** annex to the project contract the project objectives are made more specific. The following is a copy of the section **Project Summary** from page 4 of the annex given in *italics*,which has been annotated with comments describing how the project has achieved each aim.

The goal of this project is to establish the OSMIA release of TINA as **the** *environment of choice for medical image analysis research, by building on an existing, proven technology. In particular the consortium shall;*

- *Establish a developer community to support and encourage the adoption of OSMIA.*

  The infrastructure for the community has been established. In particular the resources made available in WP2 (see section 3.2) such as the web resources, mailing lists and repository are all necessary features of such a community. Although establishing an active external community has been delayed (the reason for this delay is discussed in section 2.1) the consortium is developing the 'OSMIA Foundation' in order to ensure that the community can evolve and to manage the software effectively (see section 3.5 and deliverable D5.2).

- *Validate and demonstrate the benefits of using the OSMIA release in research, ensuring the co-operative sharing of algorithmic ideas.*

  This validation has successfully been achieved with the work on work package 4, section 3.4

- *Specify and introduce a software interface enabling OSMIA executables to be driven from third party commercial products*

  The work on both work package 2 and 3, sections 3.2 and 3.3 respectively, have produced two interface mechanisms which enable this. The first, introduced on work package 2 was introduced by the code split allowed the software to be built on non X11/Posix compliant operating systems such as Microsoft Windows. The work on work package 3 introduced a Visual Studio build environment for the software allowing the libraries to be linked directly to thirdparty code. The Web Analysis Server was also developed on this work package. This system allows any user with a web-browser to interact with a version of TINA, on both a local machine as well as on a remote server. The benefits of this are discussed in section 3.3

- *Validate and demonstrate this interface as a means of enabling commercial organisations to rapidly evaluate and assess research techniques for exploitation and provide clinical end users with access to state-of-the-art research algorithms*

  The work on work package 3, culminating in the deliverable D3.5 demonstrates the success in reaching this objective.

- *Disseminate and promote the use of OSMIA at relevant exhibitions and conferences*

  Section 3.5 outlines the efforts made here. Deliverable D5.2 provides a complete description of the work undertaken.

In order to achieve a stable, supported open source release of OSMIA suitable for co-operative research the consortium will;

- *Install the existing TINA codebase onto a centralised access point (server) and provide repository services such as CVS for the distribution and contribution of code*

  Achieved under work package 2. section 3.2

- *Improve the client side installation of the system by introducing auto-configuration tools and extending the existing GUIs to include GTK and/or Tcl/Tk*

  Autoconfiguration was introduced under work package 2. section 3.2 as was the Tcl interface used for the Web analysis Server of work package 3. Also introduced in this work package (and used on work package 4) was the 'Swig' interface which enables the new TINA libraries to be linked to over 15 different interface languages. The GTK interface is approximately 50% complete but will eventually replace the existing XView interface now that Sun have adopted GTK as their preferred interface.

- *Introduce a range of documentation and community related support systems, such as email lists and a developer website*

  All achieved under work packages 2, section 3.2, see above too

- *Establish a user group of consortium members and other related parties in order to assess the systems implemented and ultimately bootstrap the user community*

  As mentioned earlier, although the user group has not yet been established outside of the project consortium, the groundwork has been done. We have made contact with a number of potential contributors and managed to distribute the code to the NeatVision user community (which includes both software developers and clinical users). Section 2.1 discusses the reasons for this delay. The section on work package 5, section 3.5, describes the 'OSMIA Foundation' and how this will be used to establish a robust community

- *Establish a public license for OSMIA*

  A BSD license has been adopted for the core library. This allows anyone to use the core library however they wish (including for commercial purposes). Users wishing to maintain other types of license of parts of their software can use 'project' components which are maintained separately.

In order to improve the clinical uptake of research techniques the consortium will promote OSMIA to commercial organisations by;

- *Introduce a open software interface using a technology such as a remote procedural call or object brookering to allow commercial or other products to execute functionality in OSMIA based software.*

  This objective has been achieved in work package 3, section 3.3 via the Web Analysis Server.

- *Implement a compatible interface in a leading commercial medical visualisation program and use it to evaluate algorithms in OSMIA for commercial exploitation*

  This objective has been achieved in work packages 2 and 3, sections 3.2 and 3.3 by the code split and introduction of Visual Studio build environments which have allowed Voxar to evaluate algorithms in TINA.

- *Establish exploitation conditions for algorithms in OSMIA (compatible with the public license)*

  This is addressed by both the adoption of a BSD license and the 'OSMIA Foundation' as described in section 3.5

Finally to disseminate OSMIA technology to both industry and academia the consortium ;

- *Promote OSMIA at exhibitions targeting clinical end users, commercial organisations as well as researchers, (ECR)*
- *Demonstrate OSMIA at conferences (MIUA, MICCAI) in Europe and the US.*
- *Distribute OSMIA with Linux distributions and with versions of medical visualisation software*

  All of these items are addressed in section 3.5

2.1 Technical development issue

One key technical development had an impact on several areas of the project. This section explains this development and how it effected the other work packages.

The original proposal was written from the viewpoint that the structure of TINA would largely remain unchanged. However during the first phase of the project (when the other partners started to review the state of the software) it became clear that the most logical process involved a large-scale re-design of the code structure. In particular, splitting the code into two code-bases one for the non-dependent pure C code libraries (tina-libs) the other for the X11/Unix specific code (tina-tools). This was a larger, more fundamental re-organisation than was originally planned, but was regarded as vital and therefore undertaken.

The effect of this on work package 3 was that it allowed a new, richer interface to the libraries to emerge. Without the constraints of the original entry points into the software the developer was free to access the full breadth of code TINA offered. This, together with the Visual Studio build environment has allowed non Unix developers to experiment with the software.

The restructuring delayed some work on work package 4 because it required the partners to wait for the new codebase to be made available and for them to become involved in some testing and debugging. It has also delayed establishing the user community as it was felt impossible to allow a wider community to become involved in such wholesale modifications, which needed to be completed on such a short time-scale. It was decided to delay such activity until the software was in an acceptable state to be acceptable to potential developers. For this reason the consortium is establising the 'OSMIA Foundation' to oversee future development and exploitation of the software as well as provide a focus for future dissemination of TINA and its allied software.

2.2 Feedback from mid-term review

The emphasis from the midterm review was on the exploitation of the software libraries, in particular with a view to providing a more commercially attractive *product*. To this end we have adopted a BSD license for the core libraries which allows them to be used in a product without modification. However, this only addresses the issue of *commercial permission*, it does not address the issue of *commercial quality*. Therefore, it was decided to adapt deliverable D3.5 (The Algorithmic Evaluation document) to include additional sections (beyond the analysis originally specified) which assessed the quality of the software for direct use in a product, such as Voxars Voxar 3D application. The conclusions from this analysis included a set of bare minimum issues which must be resolved before Voxar could even consider the software seriously for direct inclusion. These were;

1. Fix the global namespace pollution.
2. Remove the use of static state in functionality.
3. Fix the majority of warning messages produced by the Voxar preferred compiler.
4. Add module and function level comments.
5. Add basic test harnesses

Although much has been done to deal with these issues (recognising them has been an important first step) the consortium is aware that a steering group is needed to ensure these are attended to by the open source community. This is one of the reasons why the consortium is establishing the 'OSMIA Foundation' discussed in deliverable D5.2. This group has already begun to address some of the issues above. In particular retrospective code commenting and the fixing of warning messages are being addressed.

The issues raised in the mid-term review regarding establishing a user group have been discussed in the previous sections. It is also worth emphasising that **no algorithmic research has been undertaken during the course of this project**. In our contacts with potential contributors we have discussed what contributions they could make and allowed them to investigate how these could be made using TINA. Such discussion have allowed us to modify how the work on work package 2 progressed.

3 Work Packages

The following tasks were active during the period relevant to this report;

- WP1: Project Management
- WP2: Open Source Distribution
- WP3: Commercial Interface
- WP4: Community Evaluation
- WP5: Dissemination and Exploitation

In the following sections each work package will be discussed under four headings. **Objective** outlines from the original proposal what the main objective of this work package was. **Overview of work completed** presents details of the tasks undertaken in this work package. Were possible a task-by-task review is given. Finally, **Summary of achievements** concludes the section with a discussion of how accurately the work package has attained its objectives.

3.1 WP1: Project Management

3.1.1 Objective

The objective of this work package are to ensure the smooth running of this project.

3.1.2 Overview of work completed

- Organised 4 successful project management meetings
- Produced the proposal for the ‘OSMIA Foundation’ (see D5.2)
- Organised workshop for consortium partners held at UoM
- Secondment of personnel between Voxar and ISBE
- Organised technical workshops at UoM and Voxar
- Ensured circulation of relevant documentation to all partners
- Maintained the project website, on the TINA developer site
- Liaised with the EU in sending deliverables and organising review meetings

The following personnel have been appointed to the project by the relevant partners:

Partner	Personnel	Start	End	Total Effort
UoM	Tony Lacey	0	18	8
	Giovanni Bounaccorsi	4	18	14
DCU	Naser Prljaca	0	6	6
	Ovidiu Ghita	6	18	12
Voxar	Ian Poole	0	18	12
UoWO	John Barron	0	18	18

3.1.3 Summary of achievements

The smooth running and successful execution of the project.

3.2 WP2: Open Source Distribution

3.2.1 Objective

To transform the current TINA software distribution into a self consistent, easily maintained, uncluttered codebase with a robust installation procedure available on a variety of platforms. To provide mechanisms to facilitate the distribution and support of the new TINA codebase.

3.2.2 Overview of work completed

This section is perhaps more verbose than many of the others. This is because much of what has been achieved in this work package is not documented elsewhere.

WP2 Task 1: Server configuration.

A Linux based Dell workstation was procured and configured. The machine is hosted at UoM and managed by UoM staff who have extensive UNIX platform experience. A Debian linux installation was used as UoM hosts a local mirror of the Debian distribution making installation over the network very efficient. Debian is also a distribution with a high degree of competence in automatic update, particularly with regards to security patches. This achieved the aim of a well configured, easy to maintain system. This machine currently has an uptime of over 200 days.

The domain `tina-vision.net` was registered. The TINA name was maintained (as opposed to using a the OSMIA title) because of the long history of the software and its recognition in the field.

Software to support web hosting, email lists, software source control and automatic documentation generation was installed and configured on this server. The server installation includes the following packages;

- `Apache 1.3` : Open source web server.
- `PHP 4.1` : Web site scripting language
- `CVS 1.11` : Source control software.
- `Mailman 2.0.13` : Mailing list management software.

All software installed on the server is open source.

The server backup is done via two mechanisms. The first is using disk mirroring. The machine has two 80GB disks, the second of which acts as a mirror of the first updated every night. Secondly the machine has an on-board tape unit onto which the primary disk is backed up each week. A rotational tape system is employed, with off-site copies.

WP2 Task 2: Repository configuration.

A CVS repository was established on the server and made available accessible using the address `cvs.tina-vision.net`. A repository usage policy was devised and made available to all users. A detailed guide on using the repository was written and made available to all (this can be accessed from the developer website). All project partners were granted write access to the repository. Anonymous read access was provided to allow public access to the software.

WP2 Task 3: Alpha release debug.

At the start of the project OSMIA alpha release of TINA was made available to the relevant partners. This release was largely a snapshot of the version of TINA in use prior to the start of the OSMIA project. The aim was to establish the limitations and problems with the existing release. Using feedback from the partners involved in **WP4 Task 1** it became clear that the current mechanisms for distribution were inadequate. The details of these inadequacies were used to drive the the development of the strategies employed in **WP2 Task 4**. In particular the problems identified can be summarised as;

- Complex installation procedure requiring good UNIX knowledge and understanding of TINA build architecture.
- Unnecessary interdependency between graphical and non-graphical code.

- Inconsistencies in file conventions, such as file naming.
- Complex include strategy - resulting in over includes.
- Documentation useful, but not kept up to date with code changes.

Based on this information the repository skeleton was established and released as the pre-alpha open source distribution, available within the CVS repository, meeting milestone **WP2.M1**. This reorganisation is discussed in more detail in **WP2 Task 4**.

WP2 Task 4: Release mechanisms.

The TINA 5 codebase was re-organised into two sections: the libraries, denoted *tina-libs*, and the graphical tools, denoted *tina-tools*. The aim of this split was to establish *tina-libs* as the core library system containing all of the algorithm code, without any dependency on higher level functionality which could limit its use to a particular operating system. The code from the original source distribution was split into these two areas. The GNU Autoconf source code configuration system was chosen as the configuration mechanism for both parts of the distribution. The build structure of both areas was developed to utilise Autoconf.

WP2 Task 5: Community resources.

Two websites were setup on the acquired server. The first is the main website at `http://www.tina-vision.net`. This site was designed to be easily accessible for a range of parties from those new to TINA wanting to find out more about it to those familiar with the software wanted to get details of new projects etc. As emphasised in the proposal, the goal was not to completely populate this site with information, but to establish the framework and style templates to ensure that the ongoing development of TINA could be presented in a consistent and accessible manner.

A simple, clean structure to this site has been used with the layout and structure kept consistent. The site can be viewed with any HTML 4 compliant browser, requiring no extra plugins or supporting tools. Updates to the site are regularly tested on a wide variety of browsers and platforms including terminal based text browsers such as `lynx` and `w3m`. For ease of management the site is controlled using CVS. This provides two benefits. First it allows the site to be updated by any permitted user, with modifications being made and checked locally before the main site is changed. Secondly it allows a site backup to be easily maintained.

Other features of the main web site include automated access to the internal memo system. Within the local TINA developer group at UoM we maintain a collection of documents referred to as the ‘blue memo system’. This is a collection of documents written in regard to the algorithmic research done by TINA developers and represents the shared knowledge of the science supporting many of the algorithms within TINA. The system is now accessible from the main website and is automatically kept up-to-date with the main library. We have also made available a library of images which we have found useful for research and teaching. Again this system is automated. With both of these systems, the infrastructure setup to support the distribution of information by the local TINA development community at UoM will be used to support distribution of information by the wider community as it develops.

A second website has been introduced to support the needs of the developer community. This site can be accessed via `http://developer.tina-vision.net`. The purpose of this site is to separate developer specific information from the main website. As stated at the header of this site;

This site is intended as repository for ideas, discussions and even bits of code relating to development of the TINA open source image analysis environment. Although open for all to see this site is primarily entended to support the developer community and will probably be of little us to others.

Both sites are built using PHP which has allows for automation and simple configuration, again reducing the maintenance overhead. A single configuration file can be modified to change many of the global configuration settings.

Two further resources have been installed to support more freeform community interaction. Firstly two mailing lists have been started. A developer list with the mail address `developer-list@tina-vision.net` and an announcement list with the mail address `announce-list@tina-vision.net`. Both lists are archived and can be controlled and maintained from the main website. The developer list has been very useful even within the confines of the project partners.

The second resource which is an extension to the developer site is the Wiki. A wiki is a piece of server software that allows users to freely create and edit Web page content using any Web browser. Wiki supports hyperlinks and has a simple text syntax for creating new pages and crosslinks between internal pages on the fly. This resource is being used to allow collaborative documents to be edited such as the CVS users guide.

The software documentation infrastructure was introduced. The documentation for the software has been designed to fit into four main sections. **The Programmers Guide** is to include programming guidlines and examples for using TINA. The infrastructure for this uses the `texinfo` processing system so that both web-based and hard copy output can be produced. The **Library Reference** system uses the `doxygen` system to generate an index linked library of the software directly from the code. This enables the manual to remain up to date with the software. A HTML version of the output is made available from the website. Two smaller documents have also been initiated. The **FAQ** list and the **Coding Policy** documents provide help and advice on a variety of coding issues.

WP2 Task 6: Code update.

During this task the entire codebase for TINA was upgraded to version 5. This involved restructuring all files (over 500 C files). On each file the tasks included;

- Reorganising and renaming to agree with the new file naming convention.
- Introducing a standard header containing license disclaimer.
- Restructuring the header file system, pairing each source file with a ‘sister’ header file.
- Removal of some of the more obvious programming misdemeanours such as local prototype specifications

Each of these involved expert intervention and thus the manual editing and organisation of the files.

The library was also wrapped using the ‘Swig’ auto-wrapping package to provide bindings to over 15 other languages. The java bindings have proved vital in the work on work package 4 (section 3.4).

WP2 Task 7: Integration.

The developments under work package 3 (section 3.3) were incorporated into the TINA 5 code base as discussed in D3.4 and D3.5

WP2 Task 8: Beta release debug.

This phase of the work package has largely concentrated on the issues arising from the *commercial quality* review section of D3.5. In particular the aim has been to reduce the number of compiler warnings and to provide code-level documentation using the doxygen system described earlier.

3.2.3 Summary of achievements

- Complete refactoring of TINA codebase
- Introduction of new automatic configuration system
- Introduction of new software distribution system including a CVS repository with usage policy
- Installation of two supporting website infrastructures and other community resources
- Code building on a wider number of targets include MacOSX and Microsoft Windows 2000
- Introduction of new documentation systems
- Addition of automatic language binding system to allow TINA libraries to be called from over 15 languages

3.3 WP3: Commercial Interface

3.3.1 Objective

The objectives of this work package can be summarised as;

1. To provide a software environment that would allow potential commercial users of TINA to rapidly to evaluate algorithms and routines.
2. To provide a software environment to allow commercial users of TINA to interface TINA directly with their codebase, for either prototyping or product delivery.
3. To identify specific areas of TINA which Voxar could use to enhance its products, and develop these into prototypes based around Voxars flagship product - Voxar3D.
4. Based on experience from the above, assess the usability and value of the TINA libraries in a commercial software engineering environment.

3.3.2 Overview of work completed

The work undertaken in this package deviated from what was originally envisaged. As discussed in section 2.1 the code split into tina-libs and tina-tools has meant that one direct interface emerged naturally.

In meeting the above objectives we have;

- **Targeting objective 1:** Developed a Web Analysis server allowing users to experiment with the TINA libraries from any platform either remotely or locally. The details of this system and the technologies used are discussed in D3.4.
- **Targeting objective 2:** Created a build environment for the TINA libraries based on Visual Studio, being the most common development environment used commercially on the Windows platform. This required modifications to the library to address operating system and compiler compatibility issues. We then proceeded to write C++ conversion and wrapper routines to interface between TINA and Voxar imaging types. This work is covered in deliverable D3.3.
- **Targeting objective 3:** Based on workshops and discussions between ISBE and Voxar, including demonstrations of TINA via ”tinatool” and the Web Analysis Server, we selected the following areas as most valuable to Voxar3D;

  a) Tissue segmentation,

  b) Boundary tracing,

  c) Boundary fitting by active shape models.

  In the case of a) and b) we successfully implemented these using relevant TINA functionality; case c) remains work in progress. The prototypes have been well received within Voxar, and will be demonstrated at RSNA this November, with full productisation expected to follow soon after. This work is covered in deliverable D3.5 (sections 2-5).
- **Targeting objective 4:** Voxars experience of using TINA to develop the prototypes gave insights into its strengths and weaknesses from a commercial software engineering perspective. We distilled this into a ’critique’ of the libraries, recommending renovations to the libraries which would make them easier to use in a commercial setting - for prototyping and for direct use in a product. Progress has already been made in implementing these recommendations, but more remains to be done outwith the scope of the current project. Deliverable D3.5 (section 6) cover this in detail. These issues have also been discussed in earlier sections of this document.

3.3.3 Summary of achievements

- Development of the Web Analysis Server
- Visual Studio build environment for tina-libs
- Facilitated assessment of novel algorithms which may impact on Voxar3D development
- Stimulating relationships with key academics in the area, and help to form longer term links via the OSMIA Foundation.
- Provide commercial input to worthwhile open source system

3.4 WP4: Community Evaluation

3.4.1 Objective

To assess the suitability of the open source releases of OSMIA for community development, two evaluation projects were run with project partners respectively taking the roles of users (VSL) and contributors (UoWO). In the first case, VSL ported OSMIA algorithms into their existing software system (NeatVision); in the second, UoWO contributed existing code into OSMIA.

In wp4.t1 the community developers (VSL and UoWO) evaluated and prepared a critique (deliverable D4.1) of the alpha TINA distribution. This allowed the community developers to become familiar with TINA and helped to prioritize the actions to be taken in developing the support infrastructure in WP2. In wp4.t2 the community developers attempted to interact remotely with the system at regular (controlled) intervals and fed back results to WP2. VSL and UoWO then determined how to integrate OSMIA with their own technologies, wp4.t3 and wp4.t4, a key step in the project.

The major components of the community developers' input to the OSMIA project were to contribute existing code into OSMIA and to make the OSMIA code base available to their own software environments (wp4.t5-wp4.t9). The process of contributing code to the OSMIA repository was tested using a medical image analysis project relating to analysis of cardiac motion via optical flow (UoWO). The process of using of code downloaded from the OSMIA repository was tested using ejection fraction measurements (DCU) based on Magnetic Resonance data sets.

3.4.2 Overview of work completed

WP4 Task 1: TINA evaluation.

Objective: To evaluate the existing (alpha) TINA distribution. This involved downloading, installing and evaluating the alpha TINA distribution, and preparing a brief summary of the key components which needed to be added.

The alpha TINA distribution package and installation instructions were adequate for experienced users. The community developers were able to install and run TINA without any problems on Sun workstations running the Solaris operating system, including the installation of TINA high level applications such as medical and machine vision applications. The installation and compilation of TINA libraries under Linux (Red Hat) was also successful despite numerous compiler warnings. A number of minor functional errors became apparent on further evaluation of the TINA environment, (for example the "tinatool" template application worked, but could not be closed down). In the code base, some conflicting function declarations and function definitions were noted. While there was a significant amount of TINA documentation, much was of little use to novice users - combined with a lack of source code documentation this gave TINA a significant learning curve.

The key issue identified at this stage was the lack of suitable documentation regarding TINA functions, data structures and algorithms. From a software developer's viewpoint, a more important issue was the the tight integration of the computational engine and the user interface tools. This made third party use (such as that required by NeatVision and Voxar) of the TINA libraries very difficult.

WP4 Task 2: Feedback.

Objective: Whilst the repository system was being established, VSL attempted remote interaction with the system at regular (controlled) intervals, performed a series of trial download-configure-install cycles and fed back their results to ISBE to optimize the performance of the system.

VSL began to exercise the development cycle by implementing a number of TINA-based medical imaging applications within the NeatVision environment. This task comprised the integration into NeatVision of:

- the TINA sequence tool;
- the TINA Aorta Tracking tool;
- the TINA NMR Segmentation tool.

This involved adopting an application-based Java Native Interface (JNI) approach to interfacing with TINA. JNI gives compile- and run-time support allowing developers to call "native" code from a Java program by creating an intermediate shared library (on UNIX systems) or DLL (Microsoft Windows systems) that acts an interface layer of software. This interface library was required because calling a TINA native method from Java used additional arguments to access Java methods and properties, but the standard TINA C Library of course did not provide these arguments. The interface library handles these issues, making its own calls to the TINA Library, and sending the results back to JAVA. See D4.2 for further explanation of the issues involved in this workpackage.

While this approach allowed quick development and the interface layer was kept to a minimum, it offered very little flexibility. Another disadvantage was that the majority of the redevelopment work had to be performed on the TINA side. It became clear that this approach to integrating with TINA was difficult and time consuming and that an alternative mechanism would be required. The key to this mechanism would be the restructuring of the TINA libraries - this was also a key issue for the commercial partner, Voxar.

WP4 Task 3: NeatVision assessment.

Objective: Assessment of the NeatVision integration task. This involved detailing the key issues to be addressed to successfully integrate the NeatVision and TINA technologies.

It had become clear that the TINA libraries would have to be split to separate the user interface tools (tina-tools) and the computational engine (tina-libs) libraries to allow meaningful interaction by third party community and commercial developers. The latter, tina-libs, is of most interest to NeatVision developers as it makes the powerful TINA libraries available to them. This library re-structuring task was agreed by the OSMIA project members and constituted a major new task for the consortium.

While the application-style (JNI) mechanism outlined in wp4.2 proved to be successful, the native TINA code has to be altered as described to accept the parameters passed to native methods as standard data types. This offers little flexibility and every new application block has to be created inside the tina-libs library and the library re-compiled.

A more elegant solution is offered by the peer class concept, where for each C/C++ object (structures/classes) a corresponding Java class is created. This powerful technique offers the possibility to create and delete C/C++ objects and pass data from Java. This can be done relatively simply by creating native methods that correspond to the constructor, internal methods and destructor in the C/C++ object, and can even be automated using SWIG. This approach was adopted as the key mechanism to integrate TINA and NeatVision.

WP4 Task 4: Optical flow assessment.

Objectives: To determine the requirements for a new tinatool user to use tinatool to re-code existing 2D and 3D optical flow algorithms. In particular, to determine what problems would be encountered and how they could be resolved.

At all four OSMIA project management meetings, technical breakout sessions were held during which all outstanding technical problems were resolved.

The major stumbling block in this task was the lack of proper written documentation. For example, for stack use, the programmer's reference documentation should have included not only an explanation of all the available functions but annotated code examples. This lack made the learning curve longer than it should have been. The on-line documentation improved matters, but still does not answer the question: how does one know if there is a function to achieve a particular task and if so what is it called? A glossary of terms, indexed to the pages in the documentation were they are mentioned, would be useful!

While learning tinatool was problematic, it was not impossible (all programs were completed and tested and proved to work). A cost-benefit assessment of the time required versus reward gained for using tinatool was reasonable. The success of the exercise means that we will continue to use tinatool to compute 3D optical flow for gated MRI cardiac data over the coming year, and 1-2 conference publications are expected from this work.

WP4 Task 5: NeatVision integration.

Objective: Integration of OSMIA algorithms into NeatVision. The major component of the project tasks for the example community developers focused on contributing existing code into the OSMIA system and making the OSMIA code base available to third party community developers using technologies such as VSL's NeatVision.

The process of contributing software back into the OSMIA open source framework was tested via medical data analysis projects relating to aorta tracking (a TINA application), cardiac motion (using optical flow methods developed in D4.3) and the calculation of the Ejection Fraction (a NeatVision application using TINA functions via proxy classes) from MR cardiac image data sets. The beta TINA 5 integration into NeatVision has been fully documented in the technical reports D4.2, D4.4(i), D4.5(i) where the Java interface to tina-libs uses peer classes that are made available to NeatVision.

The following TINA and DCU applications have been made part of the NeatVision environment (See deliverable D4.4(i)):

- Nema and Aiff readers;
- Stereo rectification;
- Coil correction;

- Smart ROI - aorta tracking;
- Ejection Fraction calculation;
- Pairwise geometric histograms (based on collaboration with OSMIA members);
- Optical Flow (based on collaboration with OSMIA members).

As foreseen in the project plan, TINA and NeatVision were integrated to implement a number of medical imaging applications. The combination of the restructured TINA libraries (beta TINA 5) and the use of peer classes opened the TINA library functionality to the NeatVision developer community.

WP4 Task 6: Evaluation of NeatVision Integration.

Objective: Assess the success of previous work by performing collaborative data analysis with the NeatVision system.

We were able to download the software through CVS with ease and the installation and compilation of the beta TINA 5 libraries was successful under Linux (Red Hat). Some problems were present when the examples provided were executed, but were rectified in later releases. Some additional work was required to compile the libraries under Windows 2000, including modifications to several files. Also some methods in the UNIX version of math.h are not available in the win32 version of math.h and methods used to generate random numbers also differ between the two platforms. The compilation process was successful but generated a large number of warnings, some of which are important and will have to be addressed at a later stage.

While a number of TINA documentation issues are still outstanding the documentation framework is in place - however it requires some effort to be complete. NeatVision 2.1 is now fully compatible with TINA 5.0 and the NeatVision developers guide has been updated accordingly (www.neatvision.com). See deliverables D4.4(i) and D4.5(i) for details.

WP4 Task 7: Optical flow integration.

Objectives: The task was to integrate established 2D and 3D optical flow methods into tinatool. 3D optical flow had not been specified in the project proposal but its incorporation allowed the user (UoWO) to investigate the computation of motion in gated MRI cardiac data.

Three C files were produced: calc2Dflow.c (the routines for computing 2D optical flow), calc3Dflow.c (the routines for computing 3D optical flow) and flowtool.c (the routines for the tinatool GUI). TINAtool provided the environment to allow a user to explore/browse through 3D volumes of MRI data in movie mode (for a volume number, show a movie of all slices or for a slice number show a movie of all volumes). Also, a user could select individual slices in particular volumes for viewing. The 3D optical flow could be decomposed into XY and XZ 2D flow fields and superimposed on the sets of images. Thus one could view the 3D motion on slices of the MRI data in movie mode.

WP4 Task 8: Optical flow debug.

Objectives: To complete the implementation of 2D and 3D optical flow algorithms in tinatool.

The 2D algorithms were implemented over 10 years ago [Barron et al, 1994] and, on the basic of an error analysis carried out at that time, are known to be computationally efficient and accurate for good data. The 3D optical flow algorithms were more problematic to analyse. First, both differentiation and optical flow computation are extensions of the 2D versions of these algorithms. Second, 20 volumes of artificial 3D sinusoidal data (with the correct flow known) were able to compute the correct flow to within less than 1 percent error. This is a good indication that the derivatives and optical flow algorithms are correctly computed.

With respect to the actual tinatool programming, eventually all problems were resolved (through the technical breakout sessions at the project management meetings) and the optical flow program became very stable. The contact with ISBE generated through the OSMIA program was also invaluable.

WP4 Task 9: Optical flow contribution.

Objectives: To contribute C functions to the TINA 5.0 library for 2D and 3D optical flow.

2D and 3D Differentiation, performed by convolution of Simoncelli's 5-tap balanced/matched filters [1994], were coded and added to the open source TINA repository. The 2D and 3D differentiation was performed by repeatedly convolving 1D lowpass and 1D highpass filters. For example, to see the complexity involved, consider the computation of the temporal derivative, I_t, using five 3D volumes of data:

- First, one must smooth each volume in the x dimension,
- then one must smooth those x smoothed results in the y dimension and

- then one must smooth those y smoothed results in the z dimension.

Smoothing in any dimension is done by convolution in the appropriate dimension by the lowpass filter. The result after these convolutions is 5 volumes of 3D data smoothed in x, y and z. Finally, one convolves the highpass filter in the t dimension (over the 5 volumes of 3D data) to get a volume of I_t data.

The 2D and 3D Optical Flow was based on algorithms by Lucas and Kanade (L&K) and Horn and Schunck (H&K) in 1981. L&K is based on a local integration via Least Squares (in 5×5 2D neighbourhoods or in $5 \times 5 \times 5$ 3D neighbourhoods) of intensity derivatives into 2D and 3D optical flow. H&S is based on a global regularization of a functional that includes the motion constraint equation and a smoothness term to obtain a smooth 2D or 3D flow field. That code is now available to the community as is some data for testing purposes. In particular, a set of gated MRI cardiac data is made available.

WP4 Task 10: Assessment.

Objective: VSL to produce a report detailing their assessment of the capabilities of OSMIA which will form the deliverable D4.4 (i).

A detailed assessment of OSMIA can be found in D4.4(i). The essence of this assessment has been discussed in the previous workpackage summaries.

In general our assessment of TINA 5.0 is positive. It has been advantageous to use TINA libraries in the development of medical applications in NeatVision. Several TINA medical applications have been made part of the NeatVision environment (as mentioned previously). OSMIA has enabled the TINA libraries to be available to Java programmers, greatly expanding the potential user base. VSL are committed to continue this collaboration beyond the term of this project.

WP4 Task 11: Documentation.

Objective: Production of the relevant deliverables (see below).

- D4.1(i) DCU TINA Technical Review
- D4.2 NeatVision Integration with TINA 4
- D4.4(i) Assessment of OSMIA
- D4.5(i) NeatVision/TINA Integration Documentation / Developers guide

All documentation and reports produced and submitted within the original schedule.

3.4.3 Summary of achievements

DCU Contribution to OSMIA:

- 70% (approx.) of tina-libs exposed to Java
- Addressed warning issues related to TINA software we used
- Developed windows support for math.h, dirent.h and fnmatch.h
- Developed and implemented SWIG workspace. Documented use of same
- Developed and implemented JNI application approach. Documented use of same
- Developed and implemented Peer Class approach. Documented use of same
- Adapted PGH and Optical flow for TINA 5

OSMIA Benefits to DCU:

- NeatVision enriched
- tina-libs exposed to NeatVision community

- Access to TINA development team
- Exposure to commercial software development issues.
- Access to TINA applications and UWO optical flow
- Growth in our Cardiac imaging programme
- Highlighted a limitation of NeatVision data presentation (currently been addressed)

3.5 WP5: Dissemination and Exploitation

3.5.1 Objective

The objective of the dissemination and exploitation activities was to promote and disseminate OSMIA both electronically (such as by indexing to various search and catalogue web sites) and at academic and industrial conferences and exhibitions throughout Europe, and to support communication links within the project. Dissemination was therefore targeted at three levels: consortium, user group and universal. There were no specific tasks or milestones listed in the technical annex of the contract for work package 5, although p.24 listed the objectives and description of work to be carried out. Detailed issues have been addressed in the Dissemination Report, D5.2, and so we will only summarise the main results here.

3.5.2 Overview of work completed

At the consortium level the dissemination activities focused on face-to-face meetings for project management and the joint resolution of technical issues, as well as the establishment of the TINA/OSMIA web site and mailing lists. The face-to-face meetings held early in the project were particularly successful in that they led to a redirection of the software re-structuring to take a more practical and beneficial path than had been foreseen in the project proposal. These meetings and the mailing lists kept all project partners abreast of developments and played a significant part in the success of the project (WP2 Task 6). The original proposal foresaw regular IRC ”virtual” meetings, but in the event regular e-mail communication and the regular face-to-face technical meetings achieved the same purpose of keeping partners in close contact and solving problems in a cooperative manner.

The user group proposal did not evolve as planned. On approaching potential partners it became clear that while some were willing to participate, the volume of work required to meet the objectives of the original proposal was greater than could reasonably be expected of unfunded third parties, especially considering the large scale restructuring of the TINA libraries. We were able to obtain feedback on key issues from exisiting and new TINA users and to feed this into the decisions made by the project partners, but only now at the end of the project, with the establishment of the OSMIA Foundation, will it be possible to set up a strong user group. These issues have been discussed in section 2.1 of this document.

Widespread universal dissemination could not begin in earnest before delivery of D2.5 (Open source distribution of OSMIA) and D4.5 (Documentation for software) at the end of the project. Only now are the repositories ready for open access and contain sufficient documentation to provide guidance for non-expert users and contributors. Active dissemination during the project was initially limited but expanded as the project progressed and will continue through the activities of the OSMIA Foundation and its supporting mechanisms and documentation.

Project partners have regularly attended conferences during the project and have actively promoted OSMIA through verbal discussion and the use of promotional posters and leaflets at every opportunity. An oral presentation on the project and a demonstration of the OSMIA web analysis server (see D5.2) was given at the recent British Association Festival of Science meeting. A number of web-index sites holding links to medical image analysis and computer vision resources have been approached and most have added links to the OSMIA/TINA web site.

This work package was also responsible for deciding that a BSD style license would be adopted for the TINA software core libraries. This allows the core part of the system to be used in an unrestricted way (both commercially and non-commercially). The TINA project mechanism allows the developer to write applications which use these core libraries to be covered by alternative licenses. Thus the user can protect sensitive algorithms with more restrictive license stratergies.

We have looked into the requirements to have TINA added to standard Linux distributions, but further work will be required to allow us to do so. For example, Debian Linux requires that specific configuration files are added to the libraries to allow automatic compilation from a package manager and provides software to help do this and to test whether the Debian Policy has been followed. However, the published guidelines also make it clear that some of the issues from work package 3 must be addressed, for example the large number of compiler warnings generated by TINA would be strongly discouraged.

Clearly a major goal of the project was to establish in OSMIA an ongoing resource with a life beyond the end of the project. The project partners have established a framework for the OSMIA Foundation, which will be a self-sustaining organisation that will host and maintain the OSMIA repositories and which will provide advisory, educational and consultancy opportunities to extend the life of OSMIA beyond this period of EU funding.

3.5.3 Summary of achievements

- Ensured good dissemination between project partners
- Presentation of project OSMIA and the TINA and NeatVision systems at technical conferences and exhibitions
- Establishing the framework for the OSMIA Foundation to continue to oversee the aims of the project after its completion.
- Establish an open source BSD license for TINA

4 Outputs

4.1 Deliverables

Deliverable	**Title**	**Month**	**Nature**	**Level**	**Status**
D1.1	Project review	6	R	PP	Delivered
D1.2	Project review	18	R	PP	Delivered
D2.1	Alpha open source distribution of OSMIA	2	D	PP	Delivered
D2.2	Release of repository systems	6	D	PP	Delivered
D2.3	Release of developer website	12	D	PP	Delivered
D2.4	Release of improved codebase	17	D	PP	Delivered
D2.5	Open source distribution of OSMIA	18	D	PU	Pending
D3.1	Working specification for interface	1	R	PP	Delivered
D3.2	Interface system for OSMIA	13	D	PU	Delivered
D3.3	Interface system for Plug N View 3D	13	D	PP	Delivered
D3.4	Public release of interface specification	13	R	PU	Delivered
D3.5	Algorithm evaluation report	18	R	PP	Delivered
D4.1	Assesment of TINA technology	2	R	PP	Delivered
D4.2	OSMIA integrated with Neatvision	15	D	PU	Delivered
D4.3	Optical flow techniques in OSMIA	15	D	PU	Delivered
D4.4	Assessment of OSMIA	18	R	PU	Delivered
D4.5	Documentation for software	18	R	PU	Delivered
D5.1	Dissemination plan	2	R	PP	Delivered
D5.2	Dissemination report	18	R	PU	Delivered

4.2 Effort expended

Overall effort expenditure:

Partner	ISBE	Voxar	VSL	UoWO	**Total**
wp1	1	0	0	0	1
wp2	10	0	0	0	10
wp3	9	9	0	0	18
wp4	0	0	15	15	30
wp5	2	2	2	2	8
wp6	1	1	1	1	4
Total:	23	12	18	18	71