

# EURO P<sup>Y</sup>THON

## Conference 2003

CHARLEROI - BELGIUM  
June 25-27 2003



AixtraWare



---

## Table Of Contents

Welcome to EuroPython Conference 2003 (EPC2003)	4
Timetable	5
Talks and Speakers Details	11
Keynotes	11
Lightning Talks	11
Python Business	12
Python Frameworks	14
Python Language	16
Python in Science Industry	18
Zope	20
Interviews	25
Interview of Nicolas Chauvat	26
Interview of Tim Couper	29
Interview of Paul Everitt	32
Interview of Martijn Faassen	35
Interview of Duncan Grisby	40
Interview of Michael Hudson	42
Interview of Marc-Andre Lemburg	44
Interview of Phil Thompson	48
Interview of Moshe Zadka	50
Floorplan	53
General Information	54
Travel Information	55
Location of the Conference Building	56
Transportation Information	57
Transport in Charleroi	57
Public transport (Bus)	57
Taxi	58
Rent a bike	58
Hitching	58
Share cars	58
Places To Meet	59

Cafe	59
Restaurant	59
Green meeting place	60
CyberCafe	60
Contact Information	61
Credits	62
Credits	63



---

## Welcome to EuroPython Conference 2003 (EPC2003)

This is the second edition of our volunteer-organized event, focused on bringing together Python and Zope developers across Europe and beyond.

EPC2003 is also an opportunity for those that are new to these technologies to learn more, about both the software and the people behind it. The conference remains the only major European event dedicated solely to the Python programming language and its applications.

This second EuroPython conference is now a fact! The entire conference team worked hard to make EPC2003 as good as the original and we hope that we have succeeded. The conference has more talks/tutorials (over 70) than last year with three parallel tracks. Not to mention the luminaries and friends from across the world of Python, with whom you can chat and share stories.

Now that the second edition of this event is underway, we can say that EuroPython is an event to stay. An event to which every interested person can look forward to for years to come. And this was the goal to achieve: creating a solid, new Python event, at a low price, built by Python folks for other Python folks, with a promising future ahead.

Though we faced long and scary days in preparation, we deeply hope that you'll enjoy this second event. And we deeply thank all of you for coming to Charleroi to help move Python forward, together.

Have fun,

The EuroPython Team

EURO PYTHON  
Conference 2003

## Timetable

### Wednesday, June 25 (morning)

Room A	Room B	Room C
<b>08:30-09:00: Coffee</b>		
<b>09:00-09:45:</b> "The State of Zope 3", <i>Guido van Rossum</i> . <b>09:45-10:30:</b> "Zope 3 TTW: How far can you get without writing code?", <i>Mr. Steve Alexander</i> .	<b>09:15-09:30:</b> "PyDVT and P4DAT - Data visualization in Python", <i>Alessandro Mirone - Nicolas Chauvat</i> . <b>09:30-10:00:</b> "Python applications in Biotechnology", <i>Torsten Will</i> . <b>10:00-10:30:</b> "Cardis 210 - Python based electrocardiograph", <i>Wojciech Kosinski</i> .	<b>09:00-09:45:</b> "Writing Python Extensions in C++ using Boost.Python", <i>Jürgen Hermann</i> . <b>09:45-10:30:</b> "Factory and Template Design Patterns", <i>Alex Martelli</i> .
<b>10:30-11:00: Coffee</b>		
<b>11:00-11:30:</b> "Extreme Programming and Zope", <i>Albertas Agejevas</i> . <b>11:30-12:00:</b> "Zope in Public Administration and Education", <i>Thomas Reulbach</i> . <b>12:00-12:30:</b> "Plone State of The Union: past, present and future.", <i>Alexander Limi</i> .	<b>11:00-11:30:</b> "The SimPy Simulation Package", <i>Klaus G. Muller</i> . <b>11:30-12:00:</b> "Python as a scripting language for a new simulation environment", <i>Konstantin Teplinskiy</i> . <b>12:00-12:30:</b> "Thuban: A Pythonic View on Geographic Data", <i>Bernhard Herzog</i> .	<b>11:00-11:30:</b> "Writing Webclients", <i>Moshe Zadka</i> . <b>11:30-12:00:</b> "Dos and Don'ts of Python", <i>Moshe Zadka</i> . <b>12:00-12:30:</b> "Challenges in deploying Python apps", <i>Andy Robinson</i> .
<b>12:30-13:30: Lunch</b>		



## Wednesday, June 25 (afternoon)

Room A	Room B	Room C
<b>13:30-14:00:</b> "Tutorial : integrating C++/Fortran libraries with Python/Numeric", <i>Ludovic Aubry</i> . <b>14:00-14:30:</b> "Aspects of Bayesian Inference and Statistical Disclosure Control in Python", <i>Duncan Smith</i> . <b>14:30-15:00:</b> "An Introduction to SalStat: Application Development and Usability", <i>Alan James Salmoni</i> .	<b>13:30-14:15:</b> "Silva, a cross-medium publishing system", <i>kit blake</i> . <b>14:15-15:00:</b> "Consolidation using Zope - moving away from a closed-source world to open-source in the area of electronic publishing.", <i>Andreas Jung</i> .	<b>13:30-14:15:</b> "MacPython present and future (the road to World Domination)", <i>Jack Jansen</i> . <b>14:15-15:00:</b> "PyObjC, a technical overview", <i>Ronald Oussoren</i> .
<b>15:00-15:30: Coffee</b>		
<b>15:30-16:00:</b> "PyTables: Dealing with very large amounts of data in Python", <i>Francesc Alted</i> . <b>16:00-16:30:</b> "Using Python to process large data set and make scientific simulations in Astrophysics and Cosmology Experiments", <i>Federico Nati</i> . <b>16:30-17:00:</b> "Python and wide-field imaging in astronomy", <i>Roeland Rengelink</i> .	<b>15:30-16:15:</b> "Plone network", <i>Alexander Limi</i> . <b>16:15-17:00:</b> "Plone Overview for Programmers", <i>Maik Röder</i> .	<b>15:30-16:15:</b> "PyPy: writing Python in Python", <i>Mr Armin Rigo</i> . <b>16:15-17:00:</b> "Custom metaclasses", <i>Alex Martelli</i> .
<b>17:15-18:00:</b> "Building a Successful Distributed Community", <i>Francis Glassborow</i> .		

### ASdu International N.V.

ASdu International is the distributor of important computer book publishers.

ASdu International n.v. offers you a solution in your quest for the most suitable book.

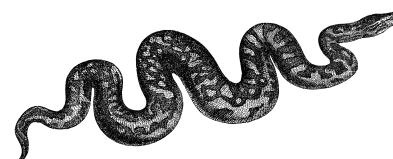


**O'REILLY™**

One of the publishers we distribute is O'Reilly. The O'Reilly books, known for the animals on their covers, occupy a treasured place on the shelves of the developers building the next generation of software. From the Internet to XML, opensource, .NET, Java and Web Services, O'Reilly puts technology on the map.

More information? Contact us !

ASdu International n.v.  
 Battelsesteenweg 455 d  
 2800 Mechelen  
 tel.: 015 285 440  
 fax: 015-285 449  
 email: [info@asdu.be](mailto:info@asdu.be)  
[www.asdu.be](http://www.asdu.be)



## Thursday, June 26 (morning)

Room A	Room B	Room C
<b>08:30-09:00: Coffee</b>		
<b>09:00-09:45:</b> "Increasing Small Business Web Service Adoption with Python", <i>M. Darrin Tisdale</i> . <b>09:45-10:30:</b> "How Python rescued a COM-Reporting Project", <i>Harald Armin Massa</i> .	<b>09:00-09:45:</b> "Unit testing Zope for fun and profit", <i>Stefan H. Holek</i> . <b>09:45-10:30:</b> "Archetypes: Next Generation CMF/Plone Types", <i>Benjamin Saller</i> .	<b>09:00-10:00:</b> "Libxml and libxslt bindings: from specs to running code", <i>Daniel Veillard</i> . <b>10:00-10:30:</b> "WebTK - Develop websites using widgets on an event-based framework", <i>Xavier Antoviaque</i> .
<b>10:30-11:00: Coffee</b>		
<b>11:00-11:45:</b> "DADO Application Server", <i>Dirk Holtwick</i> . <b>11:45-12:30:</b> "Real World Python Business Applications", <i>John Pinner</i> .	<b>11:00-11:45:</b> "icoya XML CMS", <i>Niels Mache</i> . <b>11:45-12:30:</b> "Zope Europe Association", <i>Paul Everitt, Francesco Ciriaci, Robert Boulanger, Heimo Laukkanen</i> .	<b>11:00-11:30:</b> "Presentation of itools: workflow, xml, etc..", <i>Juan David Ibáñez Palomar</i> . <b>11:30-12:00:</b> "Beyond the Config File", <i>Leonard Richardson</i> . <b>12:00-12:30:</b> "Forest, an XML Database", <i>Martijn Faassen</i> .
<b>12:30-13:30: Lunch</b>		



# ZeOmega

open minds' open solutions

Zope, Python Consulting and Hosting

**ZeOmega Infotech Pvt. Ltd.,  
Bangalore,  
INDIA.**

visit us: **[www.zeomega.com](http://www.zeomega.com)**

## Thursday, June 26 (afternoon)

Room A	Room B	Room C
<b>13:30-14:15:</b> "An Introduction to PyQt", <i>Phil Thompson</i> . <b>14:15-15:00:</b> "CORBA features for large-scale application development", <i>Dr. Duncan Grisby</i> .	<b>13:30-14:00:</b> "XML Schema - Basis for SchemaEditor", <i>Nik Klever, Ulrich Schneider</i> . <b>14:00-14:30:</b> "SchemaEditor - a partly validating, web-based XML Schemaeditor", <i>Ulrich Schneider, Nik Klever</i> . <b>14:30-15:00:</b> "The Collective - a collaborative software development effort for Plone and CMF.", <i>Alexander Limi</i> .	<b>13:30-14:15:</b> "Geeks, teams and the human factor: does Python make a difference in projects", <i>Andrew Smart</i> . <b>14:15-15:00:</b> "mxLicenseManager -- Securing Python Products", <i>Marc-André Lemburg</i> .
<b>15:00-15:30: Coffee</b>		
<b>15:30-16:15:</b> "Using Python to write Cocoa applications for MacOSX", <i>Just van Rossum</i> . <b>16:15-17:00:</b> "Reporting and Graphics for Python Applications", <i>Andy Robinson</i> .	<b>15:30-16:00:</b> "The IKAARO Content Management System", <i>Juan David Ibáñez Palomar / Christophe Perrin</i> . <b>16:00-16:15:</b> "Assembly 2003: Casefile about building Zope / Plone website and intranet", <i>Heimo Laukkanen</i> . <b>16:15-17:00:</b> "Collaborative Portal Server® (CPS)", <i>Stefane Fermier</i> .	<b>15:30-16:15:</b> "Talking to the Government Gateway with Python", <i>Mr David Chan</i> . <b>16:15-17:00:</b> "Office XML: XML on the desktop and why Python is a key component.", <i>Niels Mache</i> .
<b>17:15-18:00:</b> "State of the Python Union", <i>Guido van Rossum</i> .		



## Friday, June 27 (morning)

Room A	Room B	Room C
<b>08:30-09:00: Coffee</b>		
<b>09:00-10:30:</b> "Stackless Python 3.0 and Applications", <i>Chistian Tismer</i> .	<b>09:00-09:30:</b> "Zope storage backends", <i>Christian Zagrodnick</i> . <b>09:30-10:00:</b> "Functional testing in Zope 3", <i>Marius Gedminas</i> . <b>10:00-10:30:</b> "Certifying the Zope3 Security Model", <i>Aroldo Souza-Leite</i> .	<b>09:00-09:30:</b> "EmPy, a Python Templating System", <i>Dinu Gherman</i> . <b>09:30-10:00:</b> "Twisted Tutorial", <i>Moshe Zadka</i> . <b>10:00-10:30:</b> "Twisted Web Tutorial", <i>Moshe Zadka</i> .
<b>10:30-11:00: Coffee</b>		
<b>11:00-11:45:</b> "The Perils of Pauline: Python Tutorials and Novice Programmers", <i>Anna Ravenscroft</i> . <b>11:45-12:30:</b> "What's new in Python 2.3", <i>Alex Martelli</i> .	<b>11:00-11:45:</b> "Workflow solutions for CMF/Plone", <i>Vincenzo Di Somma</i> . <b>11:45-12:15:</b> "Zope in eGovernance: A Case Study in Cross-Border Dispute Resolution", <i>Dr Pravir Chawdhry</i> . <b>12:15-12:30:</b> "ArchGenXML", <i>Robert Boulanger</i> .	<b>11:00-11:30:</b> "High-Level Database Interaction with SQLAlchemy", <i>Brad Bollenbach</i> . <b>11:30-12:00:</b> "Woven. A web application framework.", <i>Benjamin Bruheim</i> .
<b>12:30-13:30: Lunch</b>		

Conference 2003



## Friday, June 27 (afternoon)

Room A	Room B	Room C
<p><b>13:30-13:45:</b> "Aspect-Oriented Programming in Python", <i>Nicolas Chauvat</i>.</p> <p><b>13:30-13:45:</b> "Welcome to Lightning Talks and Open Space", <i>Anna Ravenscroft</i>.</p> <p><b>13:45-14:00:</b> "PyLint : statically checking Python code and assessing quality", <i>Sylvain Thenault</i>.</p> <p><b>14:00-14:15:</b> "Python for simulation-based network research", <i>Arkaitz Bitorika</i>.</p> <p><b>14:00-14:15:</b> "reStructuredText and the docutils document processing system", <i>Michael Hudson</i>.</p> <p><b>14:15-14:30:</b> "PDF generation by HTML like markup language", <i>Dirk Holtwick</i>.</p> <p><b>14:30-14:45:</b> "Introduction to CherryPy", <i>Remi Delon</i>.</p> <p><b>14:30-14:45:</b> "Satine: a XML Data Binding technology for Python (and something more)", <i>Francesco Garelli</i>.</p>	<p><b>13:30-14:00:</b> "CMFCollectorNG - a Zope-CMF based bugtracking system", <i>Andreas Jung</i>.</p> <p><b>14:00-14:30:</b> "Kontentor CMS", <i>Joachim Werner</i>.</p> <p><b>14:30-15:00:</b> "StarOffice/OpenOffice.org XML integration for Zope/CMF", <i>Simon Eisenmann</i>.</p>	
<b>15:00-15:30: Coffee</b>		
<p><b>15:30-15:45:</b> "Python's not Java", <i>John Wilson</i>.</p> <p><b>15:30-15:45:</b> "Slides", <i>Moshe Zadka</i>.</p>	<p><b>15:30-16:15:</b> "EduZope", <i>Robert Boulanger</i>.</p> <p><b>16:15-16:45:</b> "ERP5: a Zope/CMF based Free ERP Software", <i>Jean-Paul Smets</i>.</p>	

Conference 2003



## Talks and Speakers Details

### Keynotes

**Building a Successful Distributed Community**, Francis Glassborow

day: 1, time: 17:15

This talk reflects on the ingredients that lead to success for special interest groups whose membership is widely distributed geographically. What lessons can be learnt from the experiences of 16 years. How does a group move from being a small group of C enthusiasts to an organisation with members in over forty countries. What makes ACCU Conferences so highly rated by all attendees? How have ACCU journals survived whilst their commercial rivals ceased publication? In general what turns a bunch of enthusiasts into an organisation that continues to grow and thrive after its creators have hung up their keyboards?

Francis Glassborow, a renowned trainer and mentor in the C++ world. Over 8 years as Chairman of the Association of C C++ Users, he built up a flourishing organisation with active involvement in ISO language standards, publishes several journals, and has a strong community embracing the authors and leading lights of C, C++, C#, Java and now Python.

**State of the Python Union**, Guido van Rossum

day: 2, time: 17:15

The BDFL discusses where Python is today and where it should go the coming year.

See <http://www.python.org/~guido/bio.html>

### Lightning Talks

**The Memops UML data modelling framework: automatic generation of APIs and I/O routines for multiple target languages**, Dr. Rasmus H Fogh

day: 0, time: 14:30

Memops is a data modelling facility that autogenerates complex, fully functional APIs and I/O routines directly from a single UML data model. The model uses a subset of UML with some specialised tagged values and can be adapted to any UML modelling tool that has its own scripting language. Code is generated by Python scripts working on a Python model representation. We currently generate Python APIs and XML I/O, with Java API and SQL database supports in progress and APIs for C/C++ planned. The API includes full type and consistency checking, explicitly entered constraints, derived attributes and links, callback support for GUIs, and background loading of data. Support for undo/redo is planned. The first beta release is planned for summer 2003.

Senior Research Associate, Department of Biochemistry, University of Cambridge, UK. Structural biologist turned Python programmer. Areas: Scientific applications software (NMR spectroscopy), data modelling.

**Aspect-Oriented Programming in Python**, Nicolas Chauvat

day: 3, time: 13:30

I will quickly present Logilab's aspects library that brings aspect-oriented programming to Python. Aspects are a mean to separate concerns in different functions and classes that latter get weaved together. Example of aspects are Eiffel-like contracts, Logger, Profiler, Multi-methods/dispatcher, etc. I will try to compare Logilab's module to other existing implementation if there is enough time left.

Founder of Logilab.

**Welcome to Lightning Talks and Open Space**, Anna Ravenscroft

day: 3, time: 13:30

Brief introduction for LT speakers and conference participants regarding the structure and purpose of Lightning Talks and OpenSpace.

Track Chair

**PyLint : statically checking Python code and assessing quality**, Sylvain Thenault

day: 3, time: 13:45

Pylint is a python tool that checks if a module conforms to a coding standard. Pylint can be seen as another pychecker since nearly all tests you can do with pychecker can also be done with Pylint. But Pylint offers some more features, like checking line-code's length, checking if variable names are well-formed according to your coding standard, or checking if declared interfaces are truly implemented, and much more (see the complete check list on the web site). The big advantage with Pylint is that it is highly configurable, customizable, and you can easily write a small plugin to add a personal feature.

Logilab hacker working on many projects, ranging from Narval to Plone and including pyreverse, pylint, etc.

**Python for simulation-based network research**, Arkaitz Bitorika

day: 3, time: 14:00

We developed a framework using Python to evaluate network performance through simulation. The framework provides a high-level interface to researchers: it allows for simple parameter settings, results processing, and controls the execution of multiple parallel simulations. In the talk we will discuss the advantages and the problems encountered with Python while building this framework.

Computer Science Ph.D. student at Trinity College Dublin, Ireland

**reStructuredText and the docutils document processing system**, Michael Hudson

day: 3, time: 14:00

I plan to give a very quick overview of the syntax, tools relating to and goals of reStructuredText and the docutils project.

I'm Michael Hudson, maths student, Python developer and EuroPython gopher.

**PDF generation by HTML like markup language**, Dirk Holtwick

day: 3, time: 14:15

Presentation of an easy to use PDF generator for various contexts.

CEO spirito GmbH

**Introduction to CherryPy**, Remi Delon

day: 3, time: 14:30

This talk will give an overview of the CherryPy application server: - Ideas behind it - Features of CherryPy - Writing simple applications with CherryPy

I am the author of the CherryPy application server. I've been programming in Python for 5 years and I love it because it makes me much more productive than other languages. I also run Python-Hosting.com, a web hosting service specialized in Python.

**Satine: a XML Data Binding technology for Python (and something more)**, Francesco Garelli

day: 3, time: 14:30

The Satine framework probably offers a very simple and comprehensive way to transform a generic XML document to a Python object. For example a simple SOAP document-style service can be arranged in 5 lines of code against 25 in the equivalent Java code. Satine works well with documents with XML Schema or without it. When a XML Schema is available, Satine supports the document validation. A powerful query engine can extract data from a document using a regular-expression-like syntax. The framework is evolving in a general environment to develop enterprise applications using a XML-messages style approach.

I am a graduate student at University of Padua, Italy and at University of California, Irvine. Collaborating on software architectures with prof. Taylor, I am developing a framework for developing enterprise applications through a "components and connectors" model. So far, I have developed a XML data binding technology that is going to be used in the framework, but that may be useful alone too. This work has been presented at the PyCon conference in Washington last March.

**Python's not Java**, John Wilson

day: 3, time: 15:30

As a long term Java programmer (I started with the first public beta) I describe the culture shock I encountered in attempting to become a proficient Python programmer. I describe how Skyron, my XML transformation tool, become more Pythonic as I lurked on the Python mailing list. I describe at least one epiphany....

John has been programming for over 30 years. In that time he has used a couple of dozen languages. Over the last eight years he has been concentrating on Java. Now it's time for something completely different.... Learning the Python way of doing things is challenging and fun.

**Slides**, Moshe Zadka

day: 3, time: 15:30

Slides is a simple system to generate a presentation in Python. Its chief quality is that the source, being pure Python, is infinitely extensible -- it is as easy to add a new kind of graphic element as can be.

Moshe Zadka is been a member of the Python development team, the Twisted development team and the Twisted project. He has worked in CAD/CAM integration projects, through-the-web gene browsers and the deployment and configuration of heavily used web servers and mail servers. He gave talks and tutorials at the 9th International Python Conference, the 10th International Python Conference, the 1st Debian Conference and at the Israeli Group of Linux Users.

**Python Business**

**Increasing Small Business Web Service Adoption with Python**, M. Darrin Tisdale

day: 2, time: 09:00

Python provides an excellent platform to help small businesses reap the benefits of Web Service integration. In a cooperation between AGA Food Services and the National Business to Business Centre, AGA was able to create a Web Service platform for its small business supply chain to increase the efficiency of order distribution. For this discussion, we review the critical advantage Python played in a real-life Web Service supply chain. We discuss the architecture integrating BizTalk, Cocoon, and Python. Lastly, we note the critical success factors we found for getting small businesses to adopt Web Services.

Mr. M. Darrin Tisdale is a Partner with the Tisdale Group, an international strategy consultancy. He was worked with firms such as Rolls Royce, Ford, and Airbus by using complex adaptive evolutionary analysis to align the organization's process, people, and technology. He leverages Python in solutions where dynamic, innovative flexibility is critically important for its success. Prior to starting the Tisdale Group, Mr. Tisdale was the World-wide Chief Architect for Knowledge and Technical Support at IBM. Mr. Tisdale holds a Master of Arts and Bachelor of Arts from Boston University in Technology Development Policy.

**How Python rescued a COM-Reporting Project**, Harald

Armin Massa

day: 2, time: 09:45

A projekt to create sophisticated reports from already present business-data store in an oracle database was started using MS-Access. In mid-projekt limitations concerning OOP, Multitasking and COM-Programming motivated to change. With only "hello world"-Python experience the projekt was swapped - and successful rebuild in Python. Python together with wxPython and the COM-Extensions made a lot of great features possible. Using PyCrust as interactive COM-testing tool a significant speedup in creating COM-Applications was achieved. Python's superior exception-handling helped to integrate highly effective "in the field-feedback". (Talk may be splitted in "business motivation" and "technical tutorial 'Python, COM, wxPython, Multithreading, Databases - how to replace Visual Basic (for Applications)' - please contact me

Educated in as a "Computer Salesman" (special German education combining school and practical education), working as a network technician, field supporter, computer trainer at various small computer companies; working 5 years as "CIO" at a sales company. Own company since May 1999, doing trainings (business process, economics) and computer consulting / programming. Experience in different programming languages, fallen in love with Python in Summer 2002.

**DADO Application Server**, Dirk Holtwick  
day: 2, time: 11:00

New technologies for web application development as an alternative to the use of Zope and others.

**Real World Python Business Applications**, John Pinner  
day: 2, time: 11:45

We all know that Python is very powerful, and it has been well proven in web and other application areas. However, it has not been used extensively for core backoffice business applications - the boring things like accounts and payroll which are needed to run every business. These have been the domain of COBOL, a few 4GL's, and heavyweight systems like Oracle - a bit like using an axe to spread butter. Usually, these tools are proprietary. This talk focuses on using Python to develop 'real' business systems using Python and PyQt. It relates the problems we encountered and describes a set of tools which we have developed to help us build business applications. Unlike traditional tools, these will be available under the GPL. Also unlike the old tools they are, like Python itself, multi-platform. Finally, examples of serious business systems written with Python will be shown, including an Archive Data Management System and a powerful payroll.

John Pinner has been working with Unix and then Linux for over twenty years. He has worked with several languages including Assembler, BASIC, FORTRAN, Pascal, one of the earliest 4GL's (Sculptor), C and Python. He once had a proper job but left it to start his own company, Clockwork Software Systems, in 1987. Clockwork specialises in bespoke software solutions, mainly for business applications. Nowadays their work is centred around Python and C and they have produced a development system - Thyme - to assist in business applications development. Outside software John's interests include baroque music, valve audio and classic Armstrong Siddeley motor cars.

**Geeks, teams and the human factor: does Python make a difference in projects**, Andrew Smart  
day: 2, time: 13:30

The talk consists out of two parts. The first part focuses on how IT projects are happening today, and how the troubles and problems (delays, budget) may be related to the (often ignored) human side of programmers and developers. So some answers are given to questions like "what is a team after all?", and "how can a group of geeks form a successful team?". The second part is based on the experiences of the community: does the use of the programming language Python change the way projects are done? Is there a difference in the way the people/developers communicate? What kind of obstacles and troubles do Python based projects have? In the conclusion both parts are combined to a thesis.

Being in IT since 1986 I worked as a freelancer in every project phase from development to test including project management and from DOS over Windows to Linux. Besides that he studied economic sciences in Duesseldorf, Germany. Now mainly active as a project consultant and project manager. Supporter of the EPC conferences, co-founder and member of the board of the Python Business Forum. In the shadow live of Mr. Smart he is currently developing new ideas about IT projects based on his experiences he got while being educated to work as a systematic family therapist and systematic coach. This idea development is done within a network of colleagues of each side: well known therapists and coaches, IT managers of large companies, IT consultants and, of course, developers. This ideas try to integrate the systematic view with IT and organisational knowledge based of a valid economical view, or so.

**mxLicenseManager -- Securing Python Products**, Marc-André Lemburg  
day: 2, time: 14:15

Companies developing software in Python will not always want to make their code open source and thus need a way to protect their investment. Unfortunately, Python byte code is not really a safe way to ship code to a customer and managing code /feature access is thus difficult to implement securely. eGenix has developed a commercial tool to address this problem space. We have been using the mxLicenseManager in production since December 2002 in the mxODBC Zope DA product and are looking to turn the license manager into a product for other Python based companies to use as well. This talk demonstrates some of the features and solutions found in the mxLicenseManager which can help build a business around Python based solutions which do not focus on the open source market.

Marc-Andre is the CEO and founder of eGenix.com Software GmbH, Langenfeld, Germany. He has a degree in mathematics from the University of Düsseldorf. His work with and for Python started in Winter 1993/4, in 1997, he became Python Core Developer and since February 2002, he is working for the Python Software Foundation (PSF) as board member. Marc-Andre is the author of the well-known eGenix.com mx Extensions, e.g. mxDateTime and mxODBC.

**Talking to the Government Gateway with Python**, Mr David Chan  
day: 2, time: 15:30

The Government Gateway is the UK Government's "Internet front-end". UK businesses send messages through it to Government departments. It is designed to use open, XML-based standards and in theory it should be extremely easy to access using existing Python modules. This talk summarises Clockwork Software's experiences in developing Python software to talk to the Gateway. It will discuss: - The Gateway's specification - What the Gateway does in practice - How we used Python to meet the challenge

David Chan is a programmer at Clockwork Software Systems, a software house in Birmingham (UK) which develops Python business software and tools.

**Office XML: XML on the desktop and why Python is a key component.**, Niels Mache  
day: 2, time: 16:15

Today's businesses live on information. Therefore XML as a universal markup language plays an important role in consolidating and concentrating information generated by different channels. The origin of most of this information is the desktop with its office applications. Why is XML not widely used as the lingua franca of information generation on the desktop? In this talk I will focus on desktop application integration using XML and Python. I will give an overview of Python applications on Windows and Linux desktops. The demonstration of icoya WordXML, a XML extension for Microsoft Word, will give an overview of the XML and COM functionality of Python on the desktop. The Python application icoya WordXML converts Word content easily into the open, format-neutral and manufacturer-independent XML format. Content can, without any special technical knowledge, be structured and saved as XML using common Word templates. Notably, the conversion takes place using a XML stylesheet (XSL) and a XML transformation (XSLT). The generated XML document can be automatically validated during the conversion using a DTD (Document Type Definition). In this manner, errors in the document structure can be recognized at the appropriate time and thus only valid XML documents are created according to the appropriate DTD. struktur AG has released WordXML under GPL licence. The WordXML sourcecode is available at Sourceforge wordxml project.

Niels Mache graduated with a M.Sc. in computer science and technical biology. He was researcher in the Human Genome Project at the Institute of Parallel and Distributed High-Performance Systems (IPVR). During his time as PhD student he developed the GENIO bioinformatics suite. He has served as R engineer at Sony Telecommunication Research and Development Europe. Niels Mache was the director of R and a founder of delix GmbH, one of the first companies worldwide providing professional Linux solutions and distributions. In June 1999, the company's Linux operations were acquired by Red Hat Inc., USA. He was the director of development for Red Hat GmbH, leading the development of Red Hat Linux in Germany. Niels Mache is the CEO and co-founder of struktur AG, the manufacturer of icoya OpenContent providing solutions for content collaboration management. Niels Mache has over 50 international publications. He got awards from Jugend Forscht, German Academy Software and MasPar Computer.

## Python Frameworks

**Libxml and libxslt bindings: from specs to running code**, Daniel Veillard  
day: 2, time: 09:00

This talk will present libxml2 and libxslt Python bindings, first as a quick overall presentation of the libraries and the set of specifications covered, then will focus on more specific aspects of the bindings, how they are generated and some specificities of the resulting bindings. Then as time permits we will go through working examples showing the use of the APIs provided by the libraries and to give a sense of how effective the code and API are.

Ex co-chair of the W3C XML Linking WG, member of the XML Core WG, main implementer of the libxml and libxslt C and Python libraries wrappers. He is also implementer

and maintainer of the rpmfind services. Veillard is a Member of the Board of the Gnome Project. PhD in distributed operating system, Grenoble 96.

**WebTK - Develop websites using widgets on an event-based framework**, Xavier Antoviaque  
day: 2, time: 10:00

Today's websites are mostly developed using poorly-designed languages on top of HTML single pages, interacting through database servers. As they grow more complex, websites become websoftware, and the old model of spaghetti code shows its limits. Standing as a possible solution to this problem, WebTK is a Python framework, using Twisted as its underlying Web server, and allowing webmasters to develop websites like any classic GUI software. It is object-oriented, providing and handling event-based widgets. It also does not need any database to work, as it uses objects persistence with the help of PyPerSyst. WebTK is, at least at the time we write this, an experimental project, whose objective is to prove the accuracy of an innovative technical design. This talk will describe its underlying concepts and present a few key examples.

President of Plebia (a French non-profit organization using the model of free software to provide hosting services) and project leader for WebTK.

**Presentation of itools: workflow, xml, etc...**, Juan David Ibáñez Palomar  
day: 2, time: 11:00

This talk will present for the first time the itools Python package, which provides several tools to Python developers, mainly addressed to the web, but useful in other contexts too. Most of the talk will be focused on the workflow module and the xml sub-package, specially the "Simple Template Language". The rest of features (internationalization, the Zope utilities, etc..) will be presented too. The itools package is free software.

J. David is a software engineer, the founder of the ITAAPY company and its CTO. He has been working with Python for more than five years now. He is better known in the Zope community, being his more famous product the Localizer internationalization tool. He has contributed to the Python language itself with the implementation of plural forms support in the gettext module, to be released with Python 2.3. Now he presents to the Python community the itools package, which provides facilities for workflow, xml manipulation, internationalisation, etc..

**Beyond the Config File**, Leonard Richardson  
day: 2, time: 11:30

[I gave this talk at PyCon 2003, where a general invitation was extended to PyCon presenters to give their talks at EuroPython. My PyCon paper and slides are available at: <http://www.crummy.com/devel/PyCon2003/>. Please let me know if you find this subject matter interesting.] Oftentimes, a very small or very new web application will keep all of its configuration settings inside the code itself (hopefully as variable settings in some obvious place). To change the configuration, a user must become a developer of the application. For example: #Whether or not to enable image uploads ENABLE\_IMAGE\_UPLOADS = 1 #The maximum size of an uploaded image, in kilobytes MAXIMUM\_IMAGE\_SIZE = 100 A larger, more flexible, or more mature application may keep its

configuration settings in a text file loaded by the application at runtime. When configuring such an application a user is changing data, rather than code, and acting as an administrator rather than a developer. Python's ConfigParser module provides built-in support for this level of flexibility, for example: [Images] enable-image-upload=1 #Whether or not to enable image uploads maximum-image-size=100 #Maximum size of an uploaded image, in kb The natural next step towards better usability is to provide an interface to the configuration file from within the application itself, using the same framework and UI standards found in the rest of the application. This allows a user to configure the application while remaining just that--a user: Enable image uploads? [X] Maximum image size? 100\_\_ kb The easiest and best way to take this next step is to use a configuration framework. The purpose of my talk is to get more of the world's configuration data out of configuration files and into configuration frameworks with Web interfaces. Configuration frameworks are easier to use, can enforce configuration semantics, and don't require shell access to the hosting machine. The framework I describe is also useful for doing a user preferences engine.

I have been an independent Python developer for several years. I am the lead developer of NewsBruiser (<http://newsbruiser.tigris.org>), a weblogging system written in PYthon,

**Forest, an XML Database**, Martijn Faassen  
day: 2, time: 12:00

Forest is an XML database in Python. It stores XML documents in a compact but highly indexed fashion. The main goals are to provide scalability (large quantities of documents and larger documents), and to provide a fast XPath query component that exploits sophisticated indexes. The underlying database backends is ZODB, but other backends (bsddb, metakit) are under consideration. Forest integrates with other Python XML systems through SAX and (read-only) DOM. XML:DB API support is also investigated.

Martijn Faassen has been active in the Python and Zope communities for some years. He co-founded Infracore, a Dutch zope and python development company. In the Zope world he is best known for Formulator, his work with Zope and XML, and Silva, the XML publishing system. Strangely enough while not connected with the Python Secret Underground at all people still associate him with it (but it doesn't exist anyway).

**An Introduction to PyQt**, Phil Thompson  
day: 2, time: 13:30

An introduction to PyQt for Python programmers covering the following topics: Overview of Qt classes (GUI, XML, SQL etc) Overview of tools (eric IDE, Qt Designer etc) PyQt for application development PyQt for rapid prototyping Python implementation of C++ features (inheritance, virtual methods, protected methods etc) Qt's signal and slot mechanism A walkthrough of several PyQt examples Supported platforms Licensing Q session

Author of PyQt, the Python bindings for the Qt GUI toolkit, and SIP, a tool for creating Python bindings for C++ libraries.

**CORBA features for large-scale application development**, Dr. Duncan Grisby  
day: 2, time: 14:15

Many of the challenges of building distributed applications only arise when the applications are large, and have to scale to many clients, large data sets, and wide distribution. Introductions to middleware platforms, whether CORBA, Web Services, or something else, inevitably describe the minimal features to get up and running, and so cannot cover these difficult issues. This talk gives an overview of some of the advanced features and patterns in CORBA for dealing with large-scale systems, without ever going into great detail about how to do a CORBA "hello world". Despite covering advanced features, this talk does not require any background knowledge of CORBA, since the problems and solutions are common to all distributed systems platforms. The only CORBA angle is that CORBA has the facilities built in to the standard.

Duncan is an experienced software architect and developer specialising in distributed and concurrent systems. He is the lead developer of omniORB, a high performance open source CORBA implementation. From 1999 until its closure in April 2002, Duncan was a researcher at AT Laboratories Cambridge, Europe's leading computing and communications research laboratory. Since leaving AT he has been working as an independent contractor on a number of large-scale distributed systems integration projects, and he is co-founder of Tideway Systems Ltd. Duncan holds MA and PhD degrees from Cambridge University, England.

**Using Python to write Cocoa applications for MacOSX**, Just van Rossum  
day: 2, time: 15:30

Cocoa is a powerful GUI framework for MacOSX, having its roots in NeXTStep. The native language of Cocoa is Objective-C, but with PyObjC it is possible to write Cocoa applications in pure Python. See also Ronal Oussoren's talk on PyObjC in the Python Language Track.

Just van Rossum is an independent graphic designer and programmer based in Haarlem, The Netherlands.

**Reporting and Graphics for Python Applications**, Andy Robinson  
day: 2, time: 16:15

This is a thorough tutorial on use of the ReportLab Toolkit, our Open Source PDF and data graphics framework. It will show you how to add rich PDF output capabilities to your Python applications, complete with tables, text, advanced layout and custom data graphics; and how to generate the same graphics in real time on the web in any web framework. The EuroPython 2003 dynamic conference brochure will be used as a teaching example

Andy Robinson is the CEO and Chief Architect of ReportLab, a Python based firm with 6 people based in London; co-author of "Python Programming on Win32" (O'Reilly 2000); and a long-time member of the Python community.

**EmPy, a Python Templating System**, Dinu Gherman  
day: 3, time: 09:00

This presentation gives an overview of EmPy, a general-purpose Python templating system, designed to be flexible and simple to use. It allows to embed Python code inside any text-like document and transform the document by executing a single module on it. EmPy stands out as a well-designed and documented Python module and has some features like stream diversions which are rarely found in modules with a similar purpose. The presentation describes version 3.0 with many examples and some application context. EmPy is continuously developed by Erik Max Francis.

Dinu Gherman is a longtime Python and ReportLab advocate and consultant, an Extreme Programming believer and a Mac OS X addict. He has translated several Python books, mostly for O'Reilly, and still hopes to write his own, really soon now. His Python homepage is here: <http://python.net/~gherman>.

**Twisted Tutorial**, Moshe Zadka  
day: 3, time: 09:30

Twisted is a framework to make writing network applications easier. I will cover how to write a new protocol for Twisted and integrate them into production-quality servers or clients, how to use the authentication and authorization support and how to integrate your implementations with the various GUI toolkits Twisted supports.

**Twisted Web Tutorial**, Moshe Zadka  
day: 3, time: 10:00

Twisted.Web is a web server written in the Twisted framework. It is powerful and flexible, but much of the power is only accessible with custom Python code. I will explain how to deploy it and how to configure it in various scenarios, and I will touch a bit on how to use it to write web applications. XXX Actually 60 minutes.

**High-Level Database Interaction with SQLAlchemy**, Brad Bollenbach  
day: 3, time: 11:00

Good programmers like to get more done with less work. Great programmers are Downright Lazy. Several projects have set out to address how to specify a mapping between the Python object world and the relational database world, so that we may save ourselves from writing SQL and thus focus more of our effort on the problem domain, and less on the boring details. SQLAlchemy is an "object-relational mapper" that allows a programmer to translate rows of a database table into Python objects, and do so in a way that makes it transparent to manipulate those objects in Python, instead of directly in SQL. Some of the benefits offered by SQLAlchemy: \* simple mapping semantics. no special XML files to create, few "special rules" to have to remember. The specification is just a Python class definition. \* takes advantage of some of the features of Python 2.2. metaclasses are used to create classes on-the-fly based on columns specified, Python properties are mapped to table columns \* a relatively flat learning curve. it's easy to get up-to-speed quickly on putting SQLAlchemy to use in your project. \* current database support includes PostgreSQL, MySQL, and the recent birth of SQLite support with support for more databases planned in the near future. This talk will demonstrate the utility of SQLAlchemy and show you how it can save you time, money and effort in the day-to-day development cycle of

database-driven applications. Abstract: In deciding how to design and develop an application, we often think of separating our applications into three distinct areas: data storage, "business objects", and the interface. SQLAlchemy is a framework for helping you develop the business objects by providing a high-level interface for database interaction. In this talk, Brad Bollenbach will introduce you to SQLAlchemy and show you how it can save you time, money and effort in the day-to-day development cycle of database-driven applications.

\* Brad Bollenbach, 24 years old, from Winnipeg, MB, Canada \* Python user since early 2000 \* One of the python.org webmasters \* Long-time contributor to #python on irc.freenode.net \* Contributor to comp.lang.python \* Pair-programmed with Albertas Agejevas to implement the RAM Cache Manager for Zope 3 \* Gave a talk to the Manitoba Unix User's Group entitled "An Introduction to Python" on April 8, 2003 \* Independent Consultant (operating as BBnet.ca) -- Central Canada's only software house whose toolbox is almost exclusively based around Python; Zope, Webware, Cheetah, Modeling, SQLAlchemy and the vast toolbox that is Debian GNU/Linux. \* Current projects include: - web-based invoicing system for tracking billable hours for a consulting team (Webware, Cheetah, SQLAlchemy, PostgreSQL) - telephony using a thin Python-wrapped C library to develop automated answering systems for a fixed-rate long distance provider in Eastern Canada. - web-based marketing application to provide members of sales team with calculated spreadsheets to help them recommend fixed-rate call plans to potential clients, based on how many minutes the person currently talks long-distance, and their per minute rate (Webware, Cheetah, Modeling, ported to both MySQL and PostgreSQL)

**Woven. A web application framework.**, Benjamin Bruheim  
day: 3, time: 11:30

Woven is a web templating system, used with the Twisted Web server. It is framework designed for creating highly interactive web applications through a versatile API inspired by the Model/View/Controller paradigm. This leads to modular code that is reusable, maintainable and, once the model is understood, easy to develop. This tutorial will teach you the principles of MVC, how to develop applications in woven and give businessobjects a web interface quickly and easily.

I am a self-taught norwegian coder, focusing on the creative aspect of programming. Working on the first year as an inhouse developer coding python fulltime, and also part of the twisted development team.

## Python Language

**Writing Python Extensions in C++ using Boost.Python**, Jürgen Hermann  
day: 1, time: 09:00

The talk will provide an overview over techniques and tools for writing Python extensions, indicating why Boost.Python is one of the better choices. It will then continue with a deeper view on Boost.Python's features and usage.

Jürgen Hermann works for WEB.DE AG. His team integrates Python into WEB.DE's technology platform and



provides internal consultancy. Before that, he worked on B2B/B2C e-commerce solutions since 1995.

**Factory and Template Design Patterns**, Alex Martelli  
day: 1, time: 09:45

Detailed presentation of Python versions of Factory and Template design patterns.

Alex Martelli (alex@strakt.com) lives in Italy and is a Senior System Developer with AB Strakt. He co-edited the "Python Cookbook" and wrote "Python in a Nutshell". Alex is a member of the Python Software Foundation and a board member of the Python Business Forum.

**Writing Webclients**, Moshe Zadka  
day: 1, time: 11:00

I will explain how to write webclients (mostly, non-interactive clients) using the modules in the Python library -- http, urllib2, urllib and httplib among others. I will cover basic screen-scraping techniques, as well as some basic HTTP.

**Dos and Don'ts of Python**, Moshe Zadka  
day: 1, time: 11:30

I will cover some useful idioms to use in Python program, and more importantly, some things to avoid when using Python. I will cover various variants of "import", the "exec" keyword and the "eval" function and remark a bit on common anti-idioms people use in string and file processing.

**Challenges in deploying Python apps**, Andy Robinson  
day: 1, time: 12:00

This talk will be a reflection on the challenges of trying to build, test and deploy frameworks and applications which depend on them, with a small team and demanding customers. The challenges go beyond the current scope of Distutils. I will outline many of the issues we face, talk about tools and approaches which help or don't help, and outline a roadmap towards better solutions which could benefit.

**MacPython present and future (the road to World Domination)**, Jack Jansen  
day: 1, time: 13:30

In this talk I will start with a look at the current state of Python on the Macintosh, especially on Mac OS X. Emphasis will be on all the functionality that came from MacPython-OS9 and that is probably new to people with a unix background. I will also explain some of the design decisions, and go into details as to why I think the Macintosh needs different solutions than other unixen. Finally I will present some future plans, both hard (functionality) and soft (marketing, packaging, evangelism).

Jack Jansen has used Python since version 0.8 or so, and is part of the core developer team. He has been responsible for MacPython ever since Guido got rid of his Macintosh, 8 years ago. He works as a researcher/software designer in the multimedia group at CWI, the national dutch research institute for mathematics and computer science.

**PyObjC, a technical overview**, Ronald Oussoren  
day: 1, time: 14:15

This talk will give a technical overview of PyObjC: What is PyObjC? Why was this approach used? What are some of the interesting technical features? How to implement subclassing across the bridge. A glimpse of the future. The talk will not be an introduction to Cocoa programming with Python.

Ronald is a programmer and system administrator that uses Python to do more programming and less system administration. He's also working on PyObjC (<http://pyobjc.sourceforge.net>) because he prefers Python over Objective-C and wanted to do some Cocoa programming on MacOS X.

**PyPy: writing Python in Python**, Mr Armin Rigo  
day: 1, time: 15:30

Most of the so-called "academic" high-level programming languages are implemented mostly in themselves. The PyPy project aims to do the same for Python: writing a very flexible multi-purpose Python interpreter in Python, which would complement the (small) range of currently available implementations, CPython and Jython. The talk discusses the technical aspect behind this, including why I think that it will make a much easier-to-understand interpreter (and not as you may think a strange beast that eats its tail). I will also detail the intended applications, which range from extracting minimal Python interpreters to high-level source verification and transformation, including (as usual) world domination.

Armin Rigo is a young math Dr working in a UK university on a computer-science project. He is the author of the specializing compiler Psyco for Python.

**Custom metaclasses**, Alex Martelli  
day: 1, time: 16:15

Detailed presentation of how and why you can customize metaclasses.

**Stackless Python 3.0 and Applications**, Christian Tismer  
day: 3, time: 09:00

The talk has three parts: "Part I: Stackless 1-2-3: Re-union of my brain halves" "Part II: TOMO - a Stackless 3D-Application" Part III: 3. The EVE Game - Stackless in Outer Space For a longer description, please see <http://starship.python.net/crew/mwh/longer-stackless.html>

Christian Tismer is an independant researcher and software developer. Currently, he is in collaboration with GraS, Berlin. He holds a degree on Mathematics and Computer science from the Free University of Berlin. Since 1978, he has done numerous things with computers. From 1985 to 1995, he worked as a specialist and researcher doing Signal Processing for the Pharmaceutic Industry. Chris later founded Applied Biometrics and from the beginning supported it with high levels of automation for report generation and database handling. These systems, first written using AWK and VB, were rewritten and improved using Python. Later he founded Professional Net Service GmbH (PNS) which developed software, nearly all using and extending Python. Today, Chris is working for IronPort (San Bruno, CA). IronPort has an intriguing email server product, partially based upon Stackless and has become the major

sponsor of his Open Source work. (<http://www.ironport.com/>) Chris regularly teaches classes on Python and data management. Chris is also the founder and Web Master for the popular Python Starship web site.

#### **The Perils of Pauline: Python Tutorials and Novice Programmers**, Anna Ravenscroft

day: 3, time: 11:00

Most Python FAQs and Tutorials are written with the idea that readers are migrating to Python from a previous programming language. This can make life difficult for the novice programmer, particularly after he or she has been told that Python is a great language for learning programming! This presentation will explore some of the available tutorials and books; examine common pitfalls and assumptions encountered, with suggestions to authors and teachers on avoiding them; and highlight a few gems that demonstrate how to do it right. [Note: this presentation is targeted at tutorial authors and teachers, not at beginners themselves.]

#### **What's new in Python 2.3**, Alex Martelli

day: 3, time: 11:45

Fast overview of the few novelties in Python 2.3.

### **Python in Science & Industry**

#### **PyDVT and P4DAT - Data visualization in Python**,

Alessandro Mirone - Nicolas Chauvat

day: 1, time: 09:15

This short talk will quickly present PyDVT and Py4DAT, Python packages dedicated to data visualization and manipulation in Python. PyDVT was developed by Alexander Gobo at European Synchrotron Research Facility in Grenoble, France. Py4DAT uses PyDVT to offer a MatLab-like environment for visualization and manipulation of experimental data at ESRF.

Alessandro Mirone works at European Synchrotron Research Facility in Grenoble, France. Nicolas Chauvat founded Logilab that specializes in python for advanced and scientific computing.

#### **Python applications in Biotechnology**, Torsten Will

day: 1, time: 09:30

I will discuss application fields of Python in a biotech / pharmaceutical company, such as \* large scale sequence analysis (example applications) \* data visualization (scenarios and example applications) \* database development (DCOracle, experiences with it) \* rapid application development (interface design tools, ...)

\* System Engineer and System Designer for an Internet Service Provider \* Bioinformatics Research Scientist for an Pharma Company - database design, database application development, science support, bioinformatics algorithms

#### **Cardis 210 - Python based electrocardiograph**,

Wojciech Kosinski

day: 1, time: 10:00

Cardis 210 is the first and only electrocardiograph with software written in Python. We are proud, that it's also one of the most advanced electrocardiographs. It allows for performing resting and exercise ECG tests with computerized analysis. It features large LCD display, integrated patient database, and usage flexibility that is not available in most other cardiographs - unlike Cardis, they are not embedded full-sized GUI applications. Python is the language of choice for creating medical applications in our Institute. We develop our products for Linux and Win32 platforms. The Python framework created by our team provides a lightweight XML database for patient data and visual components for patient management, configuration and medical data presentation. We use PyQt as GUI library. Only the components requiring top performance (especially low level graphics and digital signal processing) are written in C++. In the talk I will share our experiences in using C++ to Python bindings: SIP and boost.python. Four systems based on the described skeleton have already been developed, two of them have been deployed to the client, and the electrocardiograph is just before obtaining CE safety mark. The main thesis of my presentation is that Python is a rapid application development (RAD) tool that is suitable for creating robust medical applications. I will show how we benefit from using Python at different stages of project lifetime such as design, coding, debugging, testing and deployment. After 4 years of using Python our rule of the thumb is: if possible (except for the cases, where performance is the key factor), it's always better to implement something in Python than in C++. During the talk I will show what does it mean in practice, where in our systems lies the division between C++ and Python domains and what solutions we chose to keep maximum of Python's flexibility coupled with optimized C++ code to get true rapid application development that is also joy to work on.

Project manager and developer with 10 years of experience in programming in C++ and 4 in Python. Participated in numerous projects of development of medical and other software, also spent 1 year in the USA on game programming. Currently manages the project of Cardis 210 - Python based electrocardiograph. The project is cofinanced by Polish State Committee for Scientific Research. ITAM is a leading Polish industrial R centre with expertise in electrocardiography, pacemaker heart stimulation, monitoring systems for persons in extremal environmental conditions.

#### **The SimPy Simulation Package**, Klaus G. Muller

day: 1, time: 11:00

SimPy is an Open Source (GNU GPL) discrete event simulation package written entirely in Python. It simulates multiple processes by implementing co-routines as light-weight threads using Python generators. The talk will address the use of SimPy for simulating complex systems in industrial engineering. A number of application examples will be presented in detail. Graphical simulation output, interfaces to other applications and object oriented model development will be discussed.

I am a freelance consultant on information management and advanced information system architectures. I am one of the two "inventors" of the SimPy package (SimPy= Simulation in Python). SimPy is an Open Source effort hosted on SourceForge (SimPy.SourceForge.net). My role in the SimPy activity is lead architect and implementor.

### **Python as a scripting language for a new simulation environment**, Konstantin Teplinskiy

day: 1, time: 11:30

Process engineering and Systems Biology heavily rely on efficient simulation engines. At the MPI the simulation environment DIVA is used and developed for many years now. Since this system is a very old-fashioned piece of software, a reimplementation has been started. During this software renewal we introduced Python as a language for scripting and extending the simulator. The talk presents the main ideas of the software design and implementation, mainly: (1) Wrapping of the core simulator, written in C++ and Fortran using SWIG. (2) Dynamic loading facilities for model and numerics plugins. (3) Using Python scripting for the implementation of different optimization tasks. (4) Concepts for parallel computation of optimization tasks.

I come from Ukraine. I am studying Computer Science. I took part in projects for business solutions. Currently, I work in Max-Planck-Institute in the field of scientific computing and simulation of chemical and biological processes.

### **Thuban: A Pythonic View on Geographic Data**,

Bernhard Herzog

day: 1, time: 12:00

Thuban[1] is a Python application for interactive viewing of geographic data. It is Free Software written with wxPython and runs on Unix-like systems and Windows. Thuban is extensible with user-defined functions and tools and can be used as a framework for specialized GIS applications. Thuban is still a relatively new project and currently under very active development but it is already being used as the basis for a commercial application. The talk will give an overview of Thuban's current capabilities and demonstrate the extensibility with the simulation application GREAT-ER[2] built on top of the Thuban framework.

Bernhard Herzog is head of software development at Intevation GmbH. He has been working with Python since 1996 and has been using it as his main programming language since he started writing the vector-drawing program Sketch in the same year.

### **Tutorial : integrating C++/Fortran libraries with Python/Numeric**, Ludovic Aubry

day: 1, time: 13:30

Most scientific applications need to use highly optimized computation libraries. In most cases such libraries are written in C/C++ or Fortran. This short tutorial will provide an overview of how to use Numeric to wrap C++ and Fortran code in order to steer it from Python. Examples and comparisons will help in choosing the best tool for the job, among swig, boost, f2py, pyfort and others.

Ludovic Aubry works at Logilab where he teaches Python courses to the scientific minded and develops tools and applications that mix Python with code written in other programming languages.

### **Aspects of Bayesian Inference and Statistical Disclosure Control in Python**, Duncan Smith

day: 1, time: 14:00

This talk primarily highlights an example of how someone with little programming experience, and presented with a reasonably non-trivial programming task, can (more or less) teach themselves Python and solve the problem. Bayesian Belief Networks are graphical representations of joint probability distributions. They encode the conditional independencies of the full joint distribution. The presence of conditional independencies means that Bayesian inference can be performed without reference to the full joint probability table (which in many cases would be unfeasible). This requires a number of graph and tree based algorithms to be implemented. It also requires the ability to manipulate multidimensional probability tables. It must be possible to condition on evidence, combine tables by pointwise multiplication, and marginalise by summing over table axes. With a little ingenuity all the basic table operations can be carried out using Numeric's built-in functions. One problem in statistical disclosure control is making inferences about the counts in a multidimensional table given a number of its smaller marginal tables. It turns out that the graphical models used for inference in Bayesian Belief Networks are also relevant in this area. For certain graphical representations of the structure of the marginal tables, it is possible to calculate the exact upper and lower bounds on the counts in the full table. This requires different functions for combining the tables, but again this can be achieved using Numeric. Although Python was originally chosen (at least in part) because of the shallow learning curve, it turned out to be a highly appropriate development environment for this work. Of course, this is due (in no small part) to the Numeric extensions.

A statistician and Research Associate at Manchester University. As a (more or less) non-programmer needing to do some reasonably non-trivial programming I was pointed in the direction of Python; no regrets.

### **An Introduction to SalStat: Application Development and Usability**, Alan James Salmoni

day: 1, time: 14:30

This talk will outline an introduction to SalStat, a small cross-platform and open-source application written in Python and wxPython. It was designed for scientific statistical analysis (such as those performed in the field of psychology), and features an interface that was meant to be simple to use. This talk will cover the reasons for the development of SalStat and will cover some of the design processes that occurred during its development, including some basic usability testing.

The author of 2 Python applications, SalStat and TrackBrowser, Alan is completing his Ph.D. in the human-computer interaction group at Cardiff University, and loves Python because he can write large applications easily.

### **PyTables: Dealing with very large amounts of data in Python**, Francesc Alted

day: 1, time: 15:30

Processing large amounts of data is a must for people working in such fields of scientific applications as Meteorology, Astronomy, Astrophysics, High Energy Physics or Numerical simulation, to name only a few. Existing relational or object-oriented databases usually are good solutions for applications in which multiple distributed clients need to access and update a large centrally managed database (e.g., a financial trading

system). However, they are not optimally designed for efficient read-only database queries to pieces, or even single attributes, of objects, a requirement for processing data in many scientific fields such as the ones mentioned above. The proposed talk will describe PyTables [<http://pytables.sourceforge.net>], a Python library that addresses this need, enabling the end user to manipulate easily scientific data tables and regular homogeneous (such as Numeric [<http://www.pfdubois.com/numeric/>] arrays) Python data objects in a persistent, hierarchical structure. The foundation of the underlying hierarchical data organization is the excellent HDF5 [<http://hdf.ncsa.uiuc.edu/HDF5/>] C library. [Note: If you need more info about that, please, tell me]

I'm currently working as a freelance programmer, and I spend most of my time doing that. I'm also Physics teacher at the University Jaume I in Castello, Spain. I've worked as a system manager at IFIC (Institut de Física Corpuscular de València), scientific computing manager at the Computer Center of the University Jaume I of Castelló, technical director at Servicom (a Spanish ISP bought by Retevisión) in València and systems designer in Servicom2000 (an Internet value added company), in that order. During the past years, I've worked on a variety of different problems, including operating systems, clustering, symmetric multiprocessing, scientific high performance computing, design of ISP infrastructure and services, consulting services and business management issues. Two things have influenced in my current position. Firstly, before start working, I spent one year and half at CERN (European Center for Nuclear Research) in Geneva, as a post-graduate student. During this period, it was apparent to me that large computational facilities were needed by the high energy physicists. Four years later I've received a grant to do an stage at CERFACS (Toulouse, France) in the Parallel Algorithms Project where I've done some work optimizing and parallelizing a well-known computational chemistry package, MOPAC, and achieved some quite interesting results. This stages on supercomputing centers make me feel that computers and science are very interdependent one of another. Besides, I'm very comfortable when working on these fields. During all my career I've been always interested in performance, from scientific computing facilities (mainly in Computational Chemistry and High Energy Physics; see a report I've wrote about this), to the design of efficient e-mail, Web, or Web cache systems, as well as some small incursions on High Performance Computing (HPC) programming (PVM, MPI, use of BLAS, LAPACK or the excellent IBM ESSL library). Thus, I finally decided to have a try and build myself a powerful database for scientific purposes. I live in Castelló de la Plana (Spain) spending my free time between my wife Merche, walking and jogging on the beach near home.

**Using Python to process large data set and make scientific simulations in Astrophysics and Cosmology Experiments**, Federico Nati  
day: 1, time: 16:00

Cosmology is one of the ancient sciences, but today, in modern experimental Precision Cosmology, we have the ability to build hi-tech experiments that can look at the very beginning of the Universe. These telescopes need a lot of computer science, both to make models, signal simulations and to perform very large data-set reduction. Scientists working on these tasks need a programming language that must have these features: - Can process very large data-set - Works well with maps and images - Easy to learn and use - Perform mathematical tasks, like

FFT, Linear Algebra, Random algorithms... - Can easily manage 2D and 3D plots In particular, in the Astrophysics community there is a commercial, proprietary software designed to have these characteristics and which also has a large amount of users: RS Interactive Data Language. As an experimental cosmology, I need to write this kind of programs, but I found that Python is the best solution. Many astronomers made the same choice, and thanks to them we have very good Python-based tools to modernize astronomical software. This talk compares Python to the main common tools used in Astronomy, focuses on scientific and astronomical modules and briefly presents some example from my work.

I'm an experimental cosmologist; I use Python to perform data analysis and simulations. I don't develop Numeric or other well known Python modules. I refused to use rs IDL, which is the most used Software in the Astronomy community. I found that Python can be a very good solution: I adopted and use it a lot. I also work to teach it in my lab and my University.

**Python and wide-field imaging in astronomy**, Roeland Rengelink

day: 1, time: 16:30

The development of large CCDs and multi-CCD cameras will result in a huge increase in the amount of data produced by future astronomical wide-field imaging instruments. One of these instruments is OmegaCAM, whose 32 CCDs will provide 16k x 16k pixels of data per exposure. This instrument will produce approximately 1-2 TB of data per week for 10 years. These data require novel approaches to automated administration, processing, validation, analysis and distribution of data. The aim is to make all data (500+ TB over the lifetime of the instrument) available to all astronomers for remote interactive analysis. Python is a key integration tool/implementation language for the applications that support wide-field imaging instruments like OmegaCAM. Although we are currently in the R phase, I will highlight some issues that are being addressed by the OmegaCAM development team, including: o Data reduction pipelines in astronomy, from csh to Python o An infra structure for a multi-TB research archive o Preaching to the converted: Python for large applications o A novel approach to object-relational binding between Python and SQL

I am an astronomer who obtained his MSc from the University of Amsterdam and a PhD from the University of Leiden. I started working with Python in 2000, when developing data reduction pipelines for the ESI Imaging Survey. I'm currently a lead developer for OmegaCAM, a new large astronomical camera, and astro-wise, a project to distribute processed science data from OmegaCAM among the astronomical community.

## Zope

**The State of Zope 3**, Guido van Rossum  
day: 1, time: 09:00

In this talk I will discuss the state of Zope 3, channeling Jim Fulton (who can't make it to the conference).

### **Zope 3 TTW: How far can you get without writing code?**, Mr. Steve Alexander

day: 1, time: 09:45

This talk looks at how far you can get using Zope 3 if you absolutely refuse to write any Python code. Starting from defining a schema through-the-web, we go through skinning the schema, indexing and cataloguing the instances using configurable views and adapters, setting up relationships between objects, changing the schema by adding and removing fields, and all sorts of stuff like that. Without writing any Python code. Then at the end, just for kicks, or in case we want to do more advanced things using Python code, we ask the Zope 3 tools we used to construct the schema to show us the Python code for the schema and class.

Core developer of Zope 3, involved with Zope for several years, presented at EP 2002, and various other Zope events. Written articles on Zope 3.

### **Extreme Programming and Zope**, Albertas Agejevas

day: 1, time: 11:00

Extreme programming is an agile process for quickly developing high quality software. It is especially suitable if the requirements are not clear or are changing. Some of the practices of XP, such as test driven development and pair programming, are used in the world of Zope, but their meaning is not fully understood by all Zope developers. In my presentation, I will describe the practices XP is comprised of and how they work together. I will address both business/planning and technical aspects of the process. Later, I will demonstrate an example session of test driven development, a practice which, in my opinion, should be consistently used in Zope 3 development. Also, I will present our experiences of using XP, the problems we came across and how they were solved.

I work as a programmer for Codeworks since 2001. We have been using Zope all along, but started actively contributing to Zope 3 development only since the EuroPython 2002 sprint. We've worked on RDBMS connectivity, caching infrastructure, local authentication, virtual hosting, etc. We have chosen XP as the core of our process and are successfully using it in internal and customer projects. Also, I have written my Master's thesis on XP.

### **Zope in Public Administration and Education**, Thomas Reulbach

day: 1, time: 11:30

The talk is about the status and perspectives of Zope in the public field. It shows the evolution of the last two years and tries to help in forming a special interest group. The talk provides use cases and some practical experience with introducing Zope in a heterogenous public environment.

---

### **Plone State of The Union: past, present and future.**,

Alexander Limi

day: 1, time: 12:00

Plone State of The Union: past, present and future. Alexander Limi summarizes the past year of happenings in the Plone Community, talks about the current state of affairs, and outlines the future directions of Plone. Also

suitable as a gentle introduction to the Plone framework.

Alexander Limi is co-founder of the Plone project, and presented at EP2002. He works as a usability analyst and interface designer for the company Plone Solutions in cool Scandinavia. At the moment he is active in building the Plone Network to help Plone expand its reach internationally.

### **Silva, a cross-medium publishing system**, kit blake

day: 1, time: 13:30

A year ago Infrae introduced Silva at EuroPython 2002. Since then the application has undergone tremendous development. It's been improved with a host of new features, been deployed in a Zope ZEO cluster, and auxiliary components for print and multimedia have been built. SilvaMetadata is an independent component that allows users to define their own metadata sets (e.g. LOM) and apply sets to specific content types. Docma (DOCumentMAker) is a batch-processing server, written in Python, which converts Silva XML to print formats, including a 'round trip' to and from MSWord. The Open Archive Initiative Metadata Harvesting Protocol now has a Zope implementation. Using a newly released Python module and Zope product, metadata from OAI repositories can be harvested and displayed. Silva and these components will be shown in the talk.

Co-founder of Infrae, which specializes in Zope and Python applications.

### **Consolidation using Zope - moving away from a closed-source world to open-source in the area of electronic publishing.**, Andreas Jung

day: 1, time: 14:15

We at Haufe Mediengruppe are a publisher in Germany and we are the leader in the field of CD/DVD based office products in Germany. The talk will give an overview on our ongoing process to migrate our infrastructure from technologies like Coldfusion, ASP, Oracle to Zope, Python, Postgres etc. I will give some detailed examples about finished migrations. The goal of the talk to show up the benefits, risks and "lessons learned" from working with Zope as a core technology in a middle-sized company.

- born in 1967 - master degree in Computer Science and Electrical Engineering - worked for Zope Corp. - currently working for Haufe Mediengruppe senior software engineer and Zope guru

### **Plone network**, Alexander Limi

day: 1, time: 15:30

In this talk, we present a brief overview of the international business network of Plone solution providers, stated goals and plans for the future. If you run a company that uses Plone to deliver solutions, you should not miss this talk.

### **Plone Overview for Programmers**, Maik Röder

day: 1, time: 16:15

The goal of this talk is to present a bird's-eye view of the Plone programming landscape using UML Models. Both Plone 1.1 itself, and the evolution from Plone 1.0 to Plone 1.1 will be illustrated so that developers gain a clear insight into the current state of Plone and where it is heading.

Maik Röder has been active in the Zope community since 1999, and is one of the organizers of the first Zope conference at LinuxTag 2000 in Stuttgart. He is currently working as a Plone developer at Ingeniweb in Paris, France.

**Unit testing Zope for fun and profit**, Stefan H. Holey  
day: 2, time: 09:00

- Why automated tests improve your life (-span) - How to go about unit testing Zope in the first place - Unit tests v regression tests v functional tests - Why ZopeTestCase is designed the way it is - How to write tests with ZTC - How to write custom fixtures for ZTC - Zope testing strategies

I am Director of Software Development at Uptime Systemlösungen, Vienna. I have been a speaker at the Zope BBQ 2002, where I talked about the use of Zope in a large B2C e-commerce application. I am also the author of the ZopeTestCase package.

**Archetypes: Next Generation CMF/Plone Types**, Benjamin Saller  
day: 2, time: 09:45

An introduction and review of archetypes features, design and future directions.

Primary Author of Archetypes.  
<http://sf.net/projects/archetypes>

**icoya XML CMS**, Niels Mache  
day: 2, time: 11:00

This talk is about Struktur AG's XML-based content management system.

**Zope Europe Association**, Paul Everitt, Francesco Ciriaci, Robert Boulanger, Heimo Laukkanen  
day: 2, time: 11:45

Zope Europe Association. How to make Zope grow in multitudes of 10 in Europe.

--

**XML Schema - Basis for SchemaEditor**, Nik Klever, Ulrich Schneider  
day: 2, time: 13:30

This talk is a short introduction into XML Schema. XML Schema is more powerful for all kinds of XML data handling aspects than Document Type Declarations (DTDs). This talk introduces the XML aspects of the Zope Product SchemaEditor.

After studying Mathematics and Geophysics at the Universities of Berlin and Munich, Nik Klever awarded his doctorate from the University of Berlin with a thesis about numerical mathematical computing. In 1985 he joined the Federal Institute for Geosciences and Natural Resources in Hannover to set up the computing and data communication part in the project "Foundation of a National Seismological Data Center" of the Federal Foreign Office. In 1991 he became the head of the computing center of the Bayreuth Institute for Terrestrial Ecosystem Research at the University of Bayreuth. From 1997 to 2000 he held several lectures in internet and

multimedia programming at the Universities of Applied Sciences Amberg-Weiden and Hof. In 2001 he has been appointed to professorship on the subject "New Media" at the University of Applied Sciences Augsburg. Simultaneously he has been delegated to the Center of Education for the Bavarian Universities of Applied Sciences (DiZ). His main work is focused on all aspects around XML and all kind of open information on the web, especially open source like Python and Zope.

**SchemaEditor - a partly validating, web-based XML Schemaeditor**, Ulrich Schneider, Nik Klever  
day: 2, time: 14:00

This Talk describes the Zope Product SchemaEditor, a partly validating, web-based XML Schemaeditor. This Product is mainly based on the Zope Products Parsed XML, XMLWidgets and Formulator. SchemaEditor is intended to be a first version of a general UI to build XML Document Instances driven by XML Schemas.

Ulrich Schneider studies computer sciences at the University of Applied Sciences Augsburg. He is currently working at his thesis about SchemaEditor- a partly validating, web-based XML Schemaeditor.

**The Collective - a collaborative software development effort for Plone and CMF**, Alexander Limi  
day: 2, time: 14:30

Alexander Limi will talk about the Collective - a collaborative effort in software development that produces Products for the CMF and Plone frameworks. Also covered are some of the more important Products and their uses.

**The IKAARO Content Management System**, Juan David Ibáñez Palomar / Christophe Perrin  
day: 2, time: 15:30

This talk will present the IKAARO Content Management System. IKAARO is a Zope based solution and a new player in the CMS arena. Based on its 5 key concepts (users, roles, groups, resources and tasks) IKAARO provides a simple, flexible, original and powerful solution to improve the way users work together, build information and control its flow. Some of the strongest points of IKAARO are its multilingual support, which goes beyond internationalized interfaces to internationalized content, its usability with a really simple user interface that beats most (or all) of its competitors, and its extension possibilities thanks to the simplicity and flexibility of the underlying technology.

J. David is a software engineer, the founder of the ITAAPY company and its CTO. He has been working with Python for more than five years now and with Zope for four years. He has contributed to the Python community several tools, like the flux module used for workflow. He also has contributed to the Python language itself with the implementation of plural forms support in the gettext module, to be released with Python 2.3. He is better known in the Zope community, being his most famous product the Localizer internationalization tool. Christophe Perrin is a experienced consultant, who has worked with Ernst Young and Deloitte Consulting - and now is a c-founder and partner of Intervalles, Paris based consulting company. His expertise include among other following: Content management, Emarketing, Reengineering around web enabled processes, Value

Driven Project management (Intranet, Internet, Information Systems)

**Assembly 2003: Casefile about building Zope / Plone website and intranet**, Heimo Laukkanen

day: 2, time: 16:00

Assembly is an annual event held in Helsinki, collecting thousands of active demo- and gaming-scene members into Helsinki, Finland. Assembly organizers chose this year to build their website and intranet with Zope and Plone. This presentation is about how that project was managed, what lessons we learned and how you can build a high performance Zope site by following our guides.

bio

**Collaborative Portal Server® (CPS)**, Stefane Fermier

day: 2, time: 16:15

CPS

**Zope storage backends**, Christian Zagrodnick

day: 3, time: 09:00

While there are several storage backends for Zope, two seem to be particularly useful -- Directory Storage and Ape. Directory Storage, developed by Toby Dickenson, offers a turn key solution for storing one file per revision per object, replication, snapshots and more. But still objects are just stored as python pickles meaning nothing to general tools like editors or revision control systems. Directory Storage does not support versions. Ape (short for Adaptable Persistence), developed by Shane Hathaway, separates serialisation from actual storage. Ape comes with serializers for most default Zope objects and storage gateways for file system and relational database storage. The file system representation mirrors the tree structure of Zope and uses text format as much as possible making revision control via CVS possible. The relational database storage gateway creates several tables for storing data and meta data containing mainly text as well. With either gateway pickles are only used for the remaining part of an object which no serializer did serialize. Ape does neither support undo nor versions. Generally all those storages are slower than ZODB's native storage but they offer very useful features.

Founder and co-owner of gocept, a company focused on Zope application development.

**Functional testing in Zope 3**, Marius Gedminas

day: 3, time: 09:30

Zope 3 has an extensive suite of unit tests, but until recently it had no automated functional tests. I'd like to present a tutorial on writing functional tests for the Zope 3 functional testing architecture. It could be interesting to the developers of Zope 3 itself, or third party Zope 3 products. I would also like to briefly outline how the Zope 3 functional testing framework was built on top of PyUnit, the canonical Python unit testing framework. This could be interesting to developers of other Python projects that want to do automated functional testing are looking for a solution.

I'm a Python/Zope programmer at Codeworks. Python became my favourite programming language shortly after I've learned it. We've been using Zope 2 for several projects at Codeworks. We recently started to use Zope 3 for new projects. I've participated in a couple of Zope 3 sprints, and I've been involved in several areas of Zope 3 development (SQL database support, caching infrastructure, virtual hosting support, functional testing framework).

**Certifying the Zope3 Security Model**, Aroldo

Souza-Leite

day: 3, time: 10:00

Zope3 is being submitted to the main industrial certification institution in Germany, the "Technischer Ueberwachungsverein" (TUV). TUV is traditionally recognized by Germans in general as being quasi-governmental and independent. TUV will test and certify Zope3 as far as its security model is concerned. The certificate will be recognized in Europe, North America and Japan. Is this an alternative to monopolistic security certification policies?

Diplom-Mathematiker (Univ. of Bonn), living in Bonn, Germany, working with Zope for 10 months now.

**Workflow solutions for CMF/Plone**, Vincenzo Di Somma

day: 3, time: 11:00

How to use and combine workflow engines (DCWorkflow and Reflow) with CMF/Plone to create easily flexible and modular web applications integrated in the portal.

Vincenzo Di Somma is a software engineer who expertised himself in web and workflow applications. We has been working with Zope application server for more than 3 years, and he is a senior member of the Zope3 development team. He is in the free software community since 1996

**Zope in eGovernance: A Case Study in Cross-Border Dispute Resolution**, Dr Pravir Chawdhry

day: 3, time: 11:45

This talk concerns the application of Zope in the context of the European Extra-Judicial Network for Cross-Border Dispute Resolution (EEJ-Net). The talk will present the development of a multi-organisational G2C service in EEJ-Net for cross-border alternative dispute resolution (ADR) involving 400+ ADR organisations in 17 European countries. The EEJ-Net project showcases the application of Zope for a multi-lingual transactional portal involving localisation, activity-based workflow and databases. The talk is intended for the relatively non-technical audience to illustrate how an industry-strength G2C service can be built using Zope at a low cost and over a short period of time to offer a secure and trustworthy application for potentially 400 million European citizens.

Research Scientist, Cybersecurity sector, Joint Research Centre

**ArchGenXML**, Robert Boulanger

day: 3, time: 12:15

A utility which generates out of UML Class Diagrams, Plone Applications without Coding. Future Versions will support also Workflow Generation out of UML State

## Diagrams

Started 1986 on Mainframes, switched to Windows development in 1989. Founded the BlueDynamics Label together with Phil Auersperg in 1999 and switched over to OpenSource Development. Since 2000 involved in the Boa Constructor Project. Founded BlueDynamics as Company in 2001 since this time CEO of BD.

### **CMFCollectorNG - a Zope-CMF based bugtracking system**, Andreas Jung

day: 3, time: 13:30

The talk will give an overview on the architecture and features of CMFCollectorNG. There will also be a live demonstration of the product. I will talk also about ongoing works and planned features for the future.

### **Kontentor CMS**, Joachim Werner

day: 3, time: 14:00

KONTENTOR is a ready-to-use open source content management system for Zope. It is multi-language, easy to configure and customize, and suitable for small web sites as well as large organizational portals. The talk will focus on demonstrating the practical use of KONTENTOR (customizing templates and forms, maintaining content, managing users etc.). If time permits there will also be some discussion on how KONTENTOR's features might be ported to Zope 3 and how the cooperation with other Zope-based CMS projects could look like.

Joachim Werner is the CEO of iuveno AG, an IT consulting and services business located in Nürnberg, Germany. He has been using Zope from the early days on and has written one of the first German-language articles about Zope in 2001. Joachim has a degree in business administration with majors in Marketing and Organizational Science. Prior to founding iuveno AG he worked as an IT consultant for the Bertelsmann media group. iuveno AG offers IT services for businesses, educational, governmental and non-governmental organizations with a strong focus on the Zope web application framework, including web hosting, consulting and training, and custom software development.

### **StarOffice/OpenOffice.org XML integration for Zope/CMF**, Simon Eisenmann

day: 3, time: 14:30

The talk will explain functions of CMFOODocument. XSL transformations within Zope in general and a little bit of "making use of OpenOffice.org xml file format". More information about CMFOODocument can be found at: [http://www.icoya.com/produkt/module/index\\_html/tocarticle\\_view#](http://www.icoya.com/produkt/module/index_html/tocarticle_view#) [http://www.icoya.de/support/download\\_area/zope/CMFOODocument](http://www.icoya.de/support/download_area/zope/CMFOODocument) <http://www.zope.org/Members/longsleep/CMFOODocument> CMFOODocument provides the following features: · OpenOffice.org documents are rendered into HTML, WML or any other tagged format using configurable XSL stylesheets. · Embedded binary content of OpenOffice.org documents is directly accessible via the web when being published. · Automatic scaling of embedded images for online presentation. Embedded images in OpenOffice.org documents are fully automatic scaled from print resolution to 96dpi screen resolution by high-quality bicubic filtering. · Metadata is automatically inserted into the icoya content collaboration management Dublin Core. · Full-text indexing for Unicode search including phonetic search (13 languages). · Extendable Dublin core metadata search. ·

Easy access of OpenOffice.org documents through the web. Documents can be directly edited via icoya WebDAV or web browser interface (icoya EditorIntegration). · Document specific cascading stylesheets (CSS) can be applied to modify existing OpenOffice.org document formatting (fonts, colors, etc.). · icoya high-performance cache enormously speeds up document access in a high-load, multi-user environment. icoya embedded 3-level cache persistently caches XML and non-XML data like XSLT results and scaled images. · Powerful XML processing. Capable of processing XML documents sized several hundred megabytes. · Build-in transactional database with native XML-processing. · Database server clustering support. Database servers can be easily clustered across multiple processing nodes utilizing Zope Enterprise Objects (ZEO). The OpenOffice integration and supplementary software is freely available under open source license on the icoya portal [http://www.icoya.org/support/download\\_area/zope/CMFOODocument](http://www.icoya.org/support/download_area/zope/CMFOODocument)

Simon Eisenmann studied computer science in Nürnberg, Germany. Mr. Eisenmann is developing Zope applications for struktur AG since 2001 and leads the development of the icoya Content Collaboration Management (CCM) products since 2002.

### **EduZope**, Robert Boulanger

day: 3, time: 15:30

A Zope/Plone based Educational project in cooperation with OpenSource AG, Alex Limi, INFRAE and Reflab.

### **ERP5: a Zope/CMF based Free ERP Software**,

Jean-Paul Smets

day: 3, time: 16:15

The talk will provide a brief and technical introduction to ERP5. ERP5 is a Free ERP Software published under GPL license which is now in production in the apparel industry. Using Zope/CMF as the foundation for an ERP has required some major technical contributions to the core Zope architecture in order to reach acceptable performance and features. This talk will be the occasion to present and discuss our contributions. We shall also present the ERP5 object model and show how and why we could leverage python and object databases to implement it much faster than we would have with a pure relational or static typing approach.

Dr. Jean-Paul Smets is CEO of Nexedi SARL, the main contributor to the ERP5 project. M. Smets graduated from Ecole Normale Supérieure. He has been assigned to engineering positions in the industry and as civil servant at the French Ministry of Economy, Finance and Industry. He is vice-president of the French Speaking Linux Association in charge of the EuroLinux Alliance.

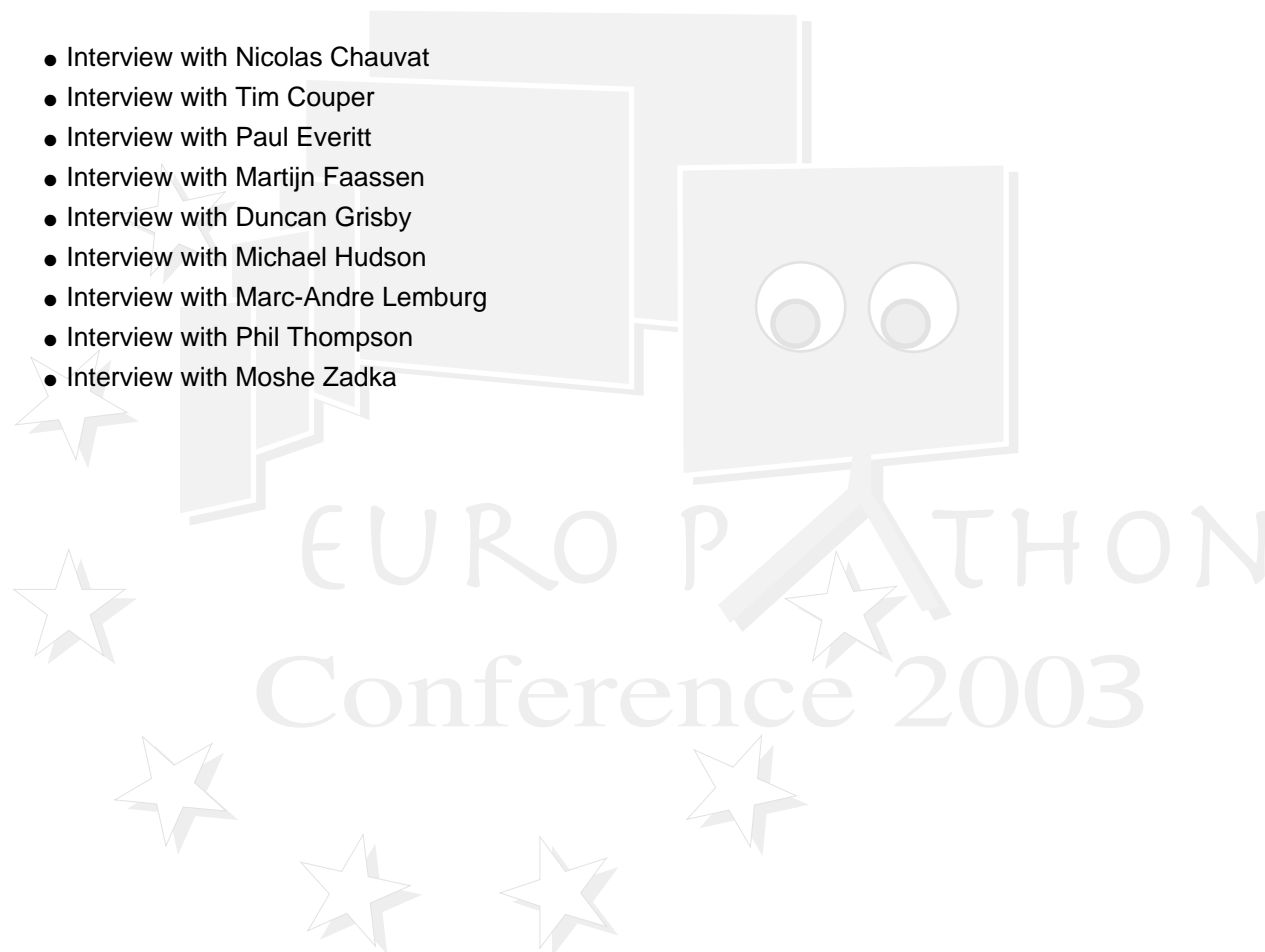


---

## Interviews

This section contains all the pre-interviews related to EuroPython2003. We hope these will slake your thirst, or stir up your appetite for EuroPython 2003.

- Interview with Nicolas Chauvat
- Interview with Tim Couper
- Interview with Paul Everitt
- Interview with Martijn Faassen
- Interview with Duncan Grisby
- Interview with Michael Hudson
- Interview with Marc-Andre Lemburg
- Interview with Phil Thompson
- Interview with Moshe Zadka



## Interview of Nicolas Chauvat

Tom Deprez

**EuroPython: Nicolas, can you introduce yourself to our readers?**

**Nicolas Chauvat:** I am one of those kids that started fooling around with computers before they turned twelve. It was a lot of fun and a great opportunity to learn tons of things, including english. Today I regret not having the internet at that time, since that would have made learning much faster.

After completing scientific studies I spent some time in a couple AI research labs and R companies. I eventually founded Logilab with a few friends in 2000. I am also the secretary of the Python Business Forum, that Logilab co-founded with AB Strakt and ReportLab last year.

**EuroPython: Logilab was founded almost 4 years ago. Can you look back satisfied that you made the right decision at that time? Which things did you encounter that you didn't expect? Both positive and negative.**

**NC:** Starting your own company is a lot of work, but it's like computers: a great opportunity to learn tons of things.

I'm very happy with the decision I made at that time. We keep growing, have a great working environment with talented people around and we are all making a living doing things we like. What's better than that ? (I know people at AB Strakt whom will agree with me here...)

**EuroPython: As last year, you are the chairman of the Python Science Industry track. What can we expect this year and did you learn something from last year?**

**NC:** Python is very strong in the scientific community and keeps gaining momentum. Last year we had several good talks and proved that a lot of disciplines are using Python for programming, ranging from astrophysics to biology.

The adoption of Python is not really surprising when you know that researchers in those fields often need to develop their own tools to process the data they gather in order to model complex phenomenons or prove theories. The ease with which Python can wrap existing C/C++ and Fortran code is also a big win. When you add simple GUIs building and database connections, very few can resist.

This year the scientific track will last for a full day and present twelve different talks, ranging from simulation to large computation and including the processing of geographic and astrophysics data. Slides are already available online for people to read, comment and think about. Authors will welcome feedback.

I wish this yearly event to be the kind of forum that any scientist needing to do some data processing by himself will turn to in order to share experiences and ideas.

**EuroPython: Can you tell us more about Logilab?**

**NC:** Logilab is a european company located in Paris, focused on advanced computing and knowledge management. We specialize in Python and provide services ranging from custom development to high-level training mixing Python, C++ and scientific computing. We also have been developing a line of products that is about to reach the market and includes a natural language processing server and a generic knowledge management and querying server. Our customers so far have been mostly industries and research entities, but we started

reaching more public administrations.

Logilab is also convinced that "Logiciel Libre" is the way to go for many reasons. We are promoting free (as in speech) software and publishing a lot of it under the GPL licence on logilab.org. Of course, that does not prevent us from providing our customers with proprietary developments, nor from selling some of our software under proprietary conditions.

**EuroPython: Why do you promote free software?**

**NC:** we promote "Logiciel Libre" because we think that IT firms should be suppliers of services, not the owners of the IT system of their clients. Nowadays, most companies would not be in business without a solid IT infrastructure. If your company uses non-free software for part of something vital, it means that a problem with that software threatens the existence of the company. You would think twice about driving abroad for the holidays if you knew that only a single garage in Europe is able to fix your car if it breaks. And you would accept not being allowed to change tires yourself? Or not allowed to look under the hood yourself nor ask someone else to do it for you... even if the garage that sold you the car closed shop?

That's even more true in the case of public administrations that are responsible for ensuring the perenity of the information that concerns the state and the citizens. Any company can die and disappear. Even one large enough to provide power to part of a continent. Free software is about putting back the user/client in control of something he needs to be able to rely on and build on. We are writing and \*selling\* free software. If our clients are unhappy with us or if we disappear, they can ask any other service company to resume development or maintenance. They do not belong to us...

Oh, and I'll take the opportunity to mention that I am opposed to software patents (but not necessarily to patents). Patents make sense to encourage invention by granting inventors a monopoly limited in time. That monopoly is supposed to allow them to get ahead of the competition and benefit from their research investments. The project of law currently discussed in Brussels is about restricting everyone's freedom to use a computer and forbidding the reuse of "ideas", not forbidding for a couple years the copy of some invention. Moreover, computer programs are already protected by the legislation that concerns authors' rights. Please read more on the topic at <http://www.eurolinux.org/>

**EuroPython: Can you elaborate on your soon to be launched line of products?**

**NC:** We have been developing software components that provide natural language processing functionalities. You can enter a sentence and have the computer make some sense out of it, then come up with an answer. I'm sure you'd want that kind a web guide for the [europepython.org](http://europepython.org) web site.

That's closely related to the Narval project we created the company for. Narval is an agent framework. It's available under the GPL licence on our website. Unfortunately, we had to put development on hold in order to finish the other projects first.

The knowledge management software I was mentioning makes it easy to maintain a knowledge base. I'm using knowledge in a broad sense. For example, we're using it

to manage our business contacts, computer hardware, etc. You enter a schema and both the web interface and the storage are handled for you. You can then browse through your data, edit it and tie entities with relationships, even if they are aggregated from different sources. A big plus is that you can query the information you stored using a high level language, much closer to your model than SQL. I'd say it's an example of what the semantic web portals are supposed to provide. Last but not least, you can query your knowledge base using natural language, so that typing "who works for Logilab?" will get you a list of persons.

**EuroPython: What do you use Python for?**

**NC:** We do everything in Python. If the result is too slow, we move part of the code to C/C++.

**EuroPython: This means that you develop everything first in Python and only later decide to move some code parts to C/C++? You never try to find the bottlenecks in advance?**

**NC:** You can \*not\* find bottlenecks in advance. "Early optimisation is the root of all evils", etc.

**EuroPython: What are your criteria to decide that a part needs to be optimized? Do you follow a certain canvas?**

**NC:** I suppose we're doing it the standard way... if it is too slow, we profile it, pick what's taking the most time and try to make it run faster. When that's done, if it's still too slow, we loop.

**EuroPython: When did you hear for the first time about Python?**

**NC:** I first heard about Python in 1995, but only used it a year later, when I was involved in a web agency and wanted a replacement for C-based CGI's to internationalize our web site. I had tried Perl a week before and never ever used it again :-)

**EuroPython: On which Python projects are you working at the moment?**

**NC:** Logilab published two new tools this week : pylint and aspects.

Pylint checks python code and syntax, looking for errors. I am a code quality maniac and try to get our whole development team to reach the highest possible quality. That means a lot of automated testing, the use of code metrics, setting up a strongly structured development environment, etc. A private company can not survive and publish a lot of free software without producing it very efficiently. Some people already know about our pyreverse tool. Both will probably end up being integrated into the same library. We also need to get closer to pychecker's authors, since pylint duplicates its functionalities.

Aspects is a module to implement Aspect-Oriented Programming in Python. We also use it for Eiffel-style contracts.

Recently, we started merging some of our knowledge management software with Plone's Archetypes. My opinion is that Plone is the future of Zope and Archetypes the future of Plone ;-). We look forward to the next version of Plone that may include Archetypes, but we decided that the best parts of that KM software will stay proprietary for some time, though. Our next step will be the implementation of semantic web features with Plone, so stay tuned.

**EuroPython: Why didn't you contacted the PyChecker authors before creating Pylint? This would perhaps avoided the work of creating a tool from the ground up?**

**NC:** Well, we already had most of what we needed, thanks to the standard library and pyreverse. And in some cases, creating a tool from the ground up and merging later with similar projects is more efficient than trying to force cooperation from the very beginning. Pychecker works and has no reason to change radically just because some punk comes in with new ideas. If you prove your ideas are worth a change and cooperation would be fruitful, that's a different matter.

**EuroPython: What is Aspect-Oriented Programming?**

**NC:** Some would say "separating orthogonal concerns". I'd say "dynamic call-path rewriting". See <http://www.aosd.net>

**EuroPython: So you use Eiffel, next to Python?**

**NC:** No, we don't use Eiffel nor any of the twenty-or-so other languages we've looked at or played with. But that does not prevent us from trying to import great things that others have implemented. Like contracts in Eiffel.

**EuroPython: Can you tell something more about contracts in Eiffel. What's nice about contracts?**

**NC:** Some people complain that "python is not strongly typed", which is a bad way putting it. Python does not bother you with method signatures or type checking during the compilation stage. The direct benefit is that it does not get in your way by forcing you to think about such checks too early. In a sense, signature checking is like optimisation, you should not try to do it too early. But when the APIs stabilize and the time comes to make the code more robust, simply checking for the types of the arguments appears like very weak. What about intervals for integers, numbers of elements for lists, etc.

Contracts are meant to address three issues : - provide really strong checking - document interfaces' requirements and promises - get rid of defensive programming techniques

If your function takes an argument named num, contracts will let you make explicit that num is an integer above 0 and below 12 for example. When contracts are activated, if the function is called with 13 as an argument, an exception ContractFailed will be raised.

Requirements on the arguments coming in and promises on the results going out are considered part of the documentation.

Since the contract requires the arguments to be correct, the body of the function can focus on processing the data and does not have to start by checking that it is correct. This helps getting rid of "defensive programming" habits.

Contracts can be enabled or disabled at will. During development, you can enable contracts and have calls made with bad arguments raise an exception, then fix the problem. When everything works, you may disable contracts to get more speed.

If someone tells you that Python is missing so-called strong-typing, just answer that it provides better than that: contracts.

**EuroPython: How do you use Eiffel contracts in Python? Can you give our readers a quick start?**

**NC:** Let's try a example of code. pre, post and inv stand for pre-conditions, post-conditions and invariants.

```

from logilab.aspects.weaver import weaver from
logilab.aspects.lib.contracts import ContractAspect,
ContractFailedError class Account : """Account can not hold
a negative amount. inv: self.amount > 0 """ def
__init__(self) : self.amount = 0 def credit(self,amount) :
"""can not credit a negative amount pre: amount > 0 """
self.amount += amount def debit(self,amount) : """can not
debit a negative amount pre: amount > 0 """ self.amount -=
amount if __name__ == '__main__' : # enable contracts
on Account class weaver.weave_methods(Account,
ContractAspect) account = Account() account.credit(100)
try: account.debit(-200) except ContractFailedError : print
"Can't debit a negative amount" try: account.debit(200)
except ContractFailedError : print "Can't debit more than
what's available"

```

For more information, download the package from [logilab.org](http://logilab.org), read the documentation and look at the examples.

**EuroPython: What did you decide that Plone is the future of Zope? There are many other CMS in Zope. Are you following the Zope 3 development?**

**NC:** I don't think I can or should "decide" anything about Zope or Plone. My personal opinion is that Plone provides what Zope/CMF was actually after. And since having "one obvious way to do it" would be nice... I don't have time to follow Zope 3 development closely but wish the other Zope based CMSes and Plone could end up being a large set of components for Zope 3. Free software is also about reusability, isn't it?

**EuroPython: What do you like best about Python?**

**NC:** Two things.

First is that Python does not enforce a design style. It's multiparadigm by nature. Want to write a ten lines script? Use Python. Need objects? Use Python. Want to mix objects and procedures? Use Python. Want part of your application in C and the rest in a high-level language that provides first class types like dates, lists and mappings? Use Python. Want more power, like introspection, very dynamic behavior and on-the-fly modification of the code? Use Python. Want contracts, aspects, logic programming? Use Python libraries from Logilab ;-)

Second is that if something you need is not in \*the\* standard library, it is probably in \*a\* standard library. Numeric, PIL, Reportlab, PyXML, twisted et al. make a real difference when compared to other languages. That makes me call Python an agile development framework rather than a programming language.

Oh, and I was about to forgot about the syntax, which is so clean that I have had a very hard time using anything else but Python to program for the past seven years.

**EuroPython: What do you not like about Python?**

**NC:** I'll like Python even better when the Python-in-Python interpreter will be available for everyone to toy with home-grown dialects and language changes or optimisations. Python needs to be stable and PyPy will be a great tool for experimentation. CPython is harder to tweak.

I also wish we could include some of the Mozart/Oz interpreter and language features, namely concurrency and distribution, but I doubt Python will ever want to go down that road. A recent thread on the mozart list mentioned that the concept of future/promise could benefit Python. It first started on another list when discussing the implementation of logic variables with

stackless. I just wish I had more time to spend on deeply understanding both Python's and Mozart's interpreters... and be able to explore the issue myself.

**EuroPython: What is the next 'big' issue Python should have, according to you?**

**NC:** I think that the standard libraries (Numeric, PIL, etc.) are the places where the real improvements should be expected. Python is already quite mature as it is. Moreover, Python 2.3 added what I missed most, the datetime module and the set type. That's why I think the next big move concerning Python itself will be Python-in-Python. Go sprinters!

**EuroPython: Are you interested in joining the sprint efforts of PyPy?**

**NC:** Unfortunately, I currently do not have the time, nor the energy.

As I said, we are already involved in and contributing to many different free software projects. Even if we're using agile methodologies like eXtreme Programming and a very structured development environment that allow us to produce software efficiently, we can not take part in all the projects we would like to see succeed.

**EuroPython: Finally, what do you expect from EuroPython?**

**NC:** A lot of goods talks, ideas and code to come back home with and think about or use. I'm confident this year's conference will fulfill my expectations, as did EuroPython 2002. See you there :-)

<http://www.europython.org>

People who would like to find out more about the mentioned topics, have a look at:

<http://www.logilab.org/>

<http://www.python-in-business.org/>

<http://aosd.net/>

<http://codespeak.net/pypy/>

<http://pychecker.sourceforge.net/>

<http://www.mozart-oz.org/>

<http://www.plone.org/>

<http://www.python.org/>

<http://www.zope.org/>

[http://docs.eiffel.com/general/guided\\_tour/](http://docs.eiffel.com/general/guided_tour/)

## Interview of Tim Couper

Tom Deprez

**EuroPython: Hi Tim, tell us something about yourself.**

Tim Couper: I'm from England ( .. still part of Europe ..), studied mathematics at Edinburgh (BSc) Oxford (PhD).

Since those days, I've been working in IT for over 20 years (I even pre-date Microsoft), and have run a number of companies over this time - software, hardware, systems telecomms integrators, and have been on the boards of disparate companies, from those providing compiler technology to satellite connectivity. This has given me lots of time to develop expertise in most of the major languages and o/s; then about 3 years ago, I discovered Python "by accident", and found it to be absolutely BRILLIANT for an extremely wide range of applications.

Within the Python/Open Source space, I'm currently the co-ordinator of the UK Python community conferences, on the Advisory Board of LinuxWorld UK, and am also the "Conference Compere" of the EuroPython Conference.

**EuroPython: So you are the owner of several companies?**

TC: I have owned, operated and sold several companies in my time; now I run TAC Software, a software consulting business, which focusses on providing solutions for companies with complex or large data-related issues.

**EuroPython: You must have a lot of experience and learned a lot over the years. Can you give some advice to just-starting companies or people planning to start?**

TC: That's a big subject! Be realistic about you can achieve, and about the products and services that you want to deliver. Plan pessimistically about cash flow and sales; realistically about expenses and costs. Find investors, if necessary, with whom you are comfortable, communicate clearly and openly your aspirations and expectations of them. And then be prepared to work hard, not only at technology, but at accounts, administration, sales, and all the other aspects of company life. And have fun while doing it: always laugh at least once a day, particularly at yourself.

**EuroPython: How big can TAC Software be seen? Do you mostly work with fixed personel or do hire on a freelance basis?**

TC: Mostly with freelance specialists on a case by case basis.

**EuroPython: Is TAC Software Python focussed? Or do you also use other languages, depending on the customer requests?**

TC: My preference would be to do Python/Jython work, because it is easiest to work in, can deliver great value in a short space of time, and the code produced is very maintainable. However, the nature of my role as a "technical authority" requires multi-o/s and multi-language skills.

**EuroPython: Marc-Andre already gave a hint on how you can win a customer into the use of Python? Can you share with us your ideas and experiences?**

TC: In many companies, suggesting Python as part of a solution can only be done from a basis of personal credibility. That is, someone in the client company is

prepared to bet on you or your company to deliver something of greater value than if they chose an alternative route. Finding many of such customers depends largely on the cultural value given to risk-taking, so I suspect there are many more in the USA and Asia than Switzerland or Germany, with the UK somewhere in between.

With my current client, as I shared at the last EuroPython, it was finding a couple of small problems whose rapid solution by Python gave it, and me, increased credibility. "Hanging out" with the technical teams is a good place to find needs. Listening to project managers and their problems informally, then equally informally providing suggestions, and solutions, makes a big difference. One of the key factors in its success was being able to have others with specialist expertise to come in and work on the project. Having people of the calibre of Duncan Grisby Marc-Andre Lemburg (both past and current speakers at EuroPython) able to give consulting time, makes a great difference in the eyes of the client. Maybe I should do another talk on this subject next year.

**EuroPython: How did you discover Python. By accident?**

TC: I noticed the book "Python in 24 hours" on a friend's bookshelf and borrowed it to read for interest's sake, and in doing so realised just what a useful, fully OO language it was. It was clear from the book that Python offered much of what I'd been looking for for a number of years, but found it difficult to believe that if it was SO good, and yet straightforward, why hadn't I heard of it before.

I subsequently went to the Python 9 conference in the USA hoping to find out what the REAL limitations of the language were, by meeting experienced developers. But I was amazed by the mixture of people and skillsets at the conference, and its use in such diverse applications as, for example, particle physics for large multi-dimensional array processing, transnational real-time games processing, neural network development, as well as its extensive business use - small, medium and large. And astounded by the claims of robustness across all the major o/s, which experience has shown to be true. And then to discover Jython too, providing the same elegance and functionality to the closed Java world.

I'm sure this discovery of unbelievable breadth of scope is something that many will experience when they "stumble across" Python.

**EuroPython: When did you hear about EuroPython?**

TC: I was involved in its foundation, having attended the pre-EuroPython event in Bordeaux in 2001.

**EuroPython: Together, with Marc-Andre Lemburg, you are the chairmen of the Python Business track again? What can we expect this year and did you learn something from last year?**

TC: Last year's presentations were really excellent, and provided a forum for sharing ideas and learning.

**EuroPython: If I recall correctly, last year you were accompanied by your daughter at the conference. Did she enjoy it? Is she a Python user as well?**

TC: Yes, Fiona is studying for a PhD in Edinburgh in Speech and Language Processing. Her department works in a mix of Python, C++ and Perl, but is beginning to

rationalise on Python, in which she is now quite proficient. We met with Guido at the recent UK Python conference who declared us to be a first in the python community: a first "father and daughter" pair. In the interests of completeness, I also have a son, Alex, studying Computer Science at Manchester University, who will be using Python for his final year project work.

**EuroPython: What does the lady of the house say to all this Python? :-)**

TC: My wife, Audrey has lived with me, computer technology and companies for these years. Our family isn't completely software-oriented: we have another daughter who is a primary school teacher and another son working in computer networks. So we're only half-Python.

**EuroPython: Will they join us this year at EuroPython?**

TC: Unfortunately not. Maybe next year.

**EuroPython: What is Siena-Tech?**

TC: This is a company that I and Marc-Andre Lemburg have created to promote the rather exciting distributed web-services/multi-tier architecture that we've deployed globally within a huge transnational pharmaceutical company.

**EuroPython: Can you tell us something about the Siena Web Services Architecture?**

TC: It's a Python/C enterprise, distributed services system that enables services be written essentially as Python methods. It provides also a client-side portion which is presented as a COM object on MS, or as a .so on \*nix, which makes the calling of the services as simple as .NET. So the development complexity of the client and server side software is reduced to application developer level. It also supports multiple interacting instances of Siena with auto-synchronisation of services. This enables server-side failover and can make use of client-side processing power (by having a completely synchronised Siena instance on the desktop, or on a departmental server).

**EuroPython: What's your relationship with Python?**

TC: Purely platonic. I'm representing python.org at the LinuxWorld UK (Birmingham NEC Sep 3-4 2003), where python.org is having a substantial stand, with speaker area, on the main exhibition floor (alongside Oracle, Linux, IBM and Microsoft). I feel that this is a unique opportunity to raise the profile of Python to the several thousand visitors that attend the conference and have a number of volunteers from the UK python community to work on the stand, but we still need more! And it'll be an opportunity for individuals to promote their own services. (Anyone interested, please contact me [tim@2wave.net](mailto:tim@2wave.net))

**EuroPython: On which Python projects are you working at the moment?**

TC: I'm completing the role as lead developer for the largest R data project (4 sites, 2 countries) that this pharmaceutical has ever done. Python has played a key role in being the language used to migrate the terabytes of data from all over the world, provide MQ broker processes and other client and server-side elements. As well as, of course, the vital Siena system, which is providing a middle-tier access to the Oracle database, and synchronising services to local instances of Siena in all parts of the world.

**EuroPython: What do you like best about Python?**

TC: As a language - its natural, "obvious" format, well-engineered, robust, and it's applicable to a wide variety of problem domains. And it's free.

As a community, a nicer bunch of people you couldn't find!

And the track record of its careful and thoughtful development ensures that software developed in it will be useable for years to come.

**EuroPython: What do you not like about Python?**

TC: Really only its lack of "market penetration". It deserves better. But that's up to us ...

**EuroPython: What is the next 'big' issue Python should have, according to you?**

TC: Don't have any burning technical issuesq. It is perhaps a weakness of its Open Source roots that appears to restrict Python awareness to dissemination largely only by word of mouth.

**EuroPython: Any ideas on how to improve this? Do you see some attitude changes towards Python, in the last few years?**

TC: There aren't any easy answers, but there are many who share this feeling within the Python community. We've seen valuable ongoing attempts to raise the profile through the Python Business Forum, for example, and in the focus of the Python track that Marc and I are leading. The effectiveness of these attempts has yet to be measured in the wider computing community.

Also the raising of Linux's profile (via IBM, Sun ..) and, European and other national government's interest in Open Source has given some validity to Open Source in general in Europe. So purchasing departments in major corporations may consider Open Source Linux whereas they wouldn't have 5 years ago; and with that change in world-view - from purchasing proprietary o/s to purchasing Linux - comes a greater freedom of thought for their development staff, and an increased need for cross-platform products, tools and services.

What makes this difficult is the reality that there are well-funded companies with expensive technologies whose existence would be threatened if the profile of Open Source alternatives became widely known, so they spend large amounts of PR money, e.g. advocating the advantages of J2EE, or in the case of another proprietary vendor, .NET. Or perhaps the SOAP fiasco is a better example. In all cases, the prolific PR and communications machines generate huge amounts of "noise" in the computer industry and, so, like in a busy street market, the majority of decision makers miss not only the "quieter" - more useable, better supported tools, but also miss the point.

**EuroPython: Finally, what do you expect from EuroPython?**

TC: A stimulating time, with pleasant Pythoneers from many nations. It would be my hope that this will be a starting point for making ever-stronger the national and cross-national relationships, and business, as a result of the new ideas found and acquaintances formed.

<http://www.europython.org>

People who would like to find out more about the mentioned topics, have a look at:

<http://www.siena-tech.com/>

<http://www.europython.org/>

---

<http://www.python.org/>



## Interview of Paul Everitt

Tom Deprez

**EuroPython: I guess most people will know you from the previous edition, but for the new, I think an introduction would be in it's place. Can you introduce yourself Paul?**

Paul: I'm Paul Everitt, director of the Zope Europe Association, co-founder of Zope Corporation. I've been involved in Python a long time (I have the t-shirt from the first Python conference at NIST).

I program but I'm not a good programmer, so in Python and Zope, I'm always the dumbest guy in the room. But in a way that's good, as there are things developers don't like to do for which I'm interested. Thus I've tried instead to participate through organization, communication, community-building, etc.

**EuroPython: Lot has changed in your life since last year, can you tell us about these changes?**

Paul: It's been a lot of fun. Last year at this time I was still at Zope Corp, living in Virginia with my family. Now we're living in Rennes, to the west of Paris. I'm no longer in management at a medium-sized company. Instead, I travel around, working with small companies and focusing on the important things in the "commons" part of Zope.

**EuroPython: What do you exactly mean with the "commons" part?**

Paul: The part of the Zope world that is shared by all, both business and community. Open source is about scratching your own itch. But to make progress on growing Zope, there are common tasks (organizing, communication) that must be done. Also known as "the dirty work". :^)

**EuroPython: When did you hear about EuroPython?**

Paul: Last year on the EuroZope mailing list.

**EuroPython: What was your first thought concerning such an initiative?**

Paul: I thought it was perfect timing. The original IPC Python conferences were coming unravelling. This conference proved the idea that a low-cost conference could deliver most of what Python folks wanted.

**EuroPython: Last year, you organised the Zope Track together with Tom Deprez, this year together with Heimo Laukkanen. What can we expect from the track this year and what did you learn from last year?**

Paul: Though my name is on the Zope Track this year, it's really Heimo that did all the work. Heimo is awesome. He has put together a really, really interesting track, with several topics that I'm very eager to learn more about. And it is diverse, covering technology and business.

Last year's conference seemed like such a perfect event, with the conversations and real pleasure people had talking to each other, that I really hope this year can be the same. I've learned a lot more, through organizing OSCOM (<http://www.oscom.org/>) events, about conferences since Charleroi last year. The biggest lesson is to not over-organize. Leave time and space for people to hang out and network.

**EuroPython: What made you decide in helping on the OSCOM conference and what was your involvement?**

Paul: I was invited to speak at the first OSCOM in March of last year. Wow, Zurich is a gorgeous place. The conference had this neat vibe to it, and the part I enjoyed the most was hanging out with the "competitors". During their talks I found myself thinking, "Hmm, they do such-and-such better than Zope does, we should learn from them."

Afterwards I volunteered to help organize OSCOM 2 and thus wound up on the board of directors. My biggest contribution is to write articles with Gregor that get posted on the front of Slashdot one week before the conferences. :^)

This year I taught a half-day tutorial on "Building CMS Clients With Mozilla".

**EuroPython: You are the founder of Zope Europe Association (ZEA). Was there a big need for such an organisation? How did you initiate this?**

Paul: The ZEA's mission is to grow Zope business in Europe. The story goes like this. Do the Zope companies in Europe need Zope to grow? Yup. Are they individually responsible for Zope's growth. Nope, few are big enough to fully fund a marketing person or campaign. But can we wait around and hope for the best? Nope, that won't work either.

Thus, I'm like a shared marketing expense, trying to do the publicity and communications for the common Zope market. And we know how much most developers enjoy working on marketing. :^)

I'm not a competitor in the Zope market: Zope Europe is a non-profit and I'm not part of ZC (though I still have my stock).

At ZC, we talked a lot about the growth of Zope in Europe. It wasn't really feasible, nor perhaps wise, to open an office in Europe. They proposed instead a foundation involving several companies, where I'd move to Europe and be the director from outside of ZC. I was also thinking the same thing, so we quickly worked out a deal. Thus I'm now an American abroad.

**EuroPython: Did you know of the EuroZope Foundation and did you or they contact each other?**

The original plan was to try and work out of the existing EuroZope eV. However, its purpose, charter, legal form, etc. were all geared towards something quite different, more like a user group than an business with an employee. After the Rotterdam sprint we found out from a German lawyer that it wasn't realistic.

**EuroPython: So basically only companies can become member of the ZEA? What do people have to do to become a member?**

Paul: I'm a big believer in trying to focus your mission and not be all things to all people. Conversely, this means being clear about what you "won't" do, which is where lots of people fail and thus dilute the message into corporatespeak.

Zope Europe is about growing the business of Zope. It isn't a user group. However, there is a huge need for user groups in each country, and it's part of my job to support these efforts. Increasingly I think there's a role for a common, top-level user group that speaks for the Zope community. If anybody's interested in such a thing, lemme know.

**EuroPython: How big is ZEA at the moment? I.e. how many employees does the ZEA count at the moment?**



Paul: It's the perfect size: small. :^) There are five sponsors (ZC, BlueDynamics in Austria, Euralba in France, ZopeMag in Germany, and Refflab in Italy) plus two members (Ingeniweb and Pilot Systems). These are the companies that have really supported the mission of growing Zope business in Europe. Once I finish the first round of deliverables, and thus have proof of the value, I plan to ask more companies to participate.

ZEA has one employee, me. I don't expect ZEA to get big. Rather, it is Zope Europe's job to make the participating companies big.

**EuroPython: How does Zope Corporation look at this association?**

ZC is the original ZEA sponsor, and along with BlueDynamics, they are the ones that got things in motion last summer. ZC wants Zope to grow, but they shouldn't be burdened with 100% of that responsibility. It is important for the community to shoulder some of the effort for market growth. Thus, Zope Europe isn't some permanent charity that ZC has to provide a blank check for.

All in all, it's a healthy situation and evidence that Zope is growing into adulthood.

**EuroPython: If you look back, are you at the moment in the position you thought would have been?**

Paul: On the big picture, wow, no way. The world of Zope is an amazing privilege to be a part of. Each time I see a Zope book in a bookstore, or see Zope mentioned in a mainstream magazine, or sit beside a stranger at a dinner in France and find out they've heard of Zope, I still get amazed. 5 years ago, Rob, Jim, Hadar, and I put together a company that started this ball rolling. It's a gratifying feeling, but also a deep sense of debt and responsibility to the world of Zope.

On the little picture, no, I think I'm 3-6 months behind where I thought I'd be on getting Zope Europe stuff done. Matriculation in France was more work than I thought, plus I signed up for a horrific travel schedule. For those that went to the Zope track last year, I'm 50/50 on needing to wear the "I Suck" shirt. :^)

**EuroPython: What's your relationship with Zope?**

Paul: The Zope/Business/Europe trifecta is where I spend most of my time these days. I also try to spend time on the community. Probably my biggest responsibility is to attract attention, then point that attention at important and worthy companies and projects in the world of Zope.

For instance, Giuseppe Masili is an organizer of Zope Italia, a Zope user group in Italy. There are probably people in Italy that don't know about what he's doing. But maybe they'll read this and contact him.

Thus, in Zope I try to connect things together. There's so much cool stuff and excitement going on that, obviously, I have a very fun job.

**EuroPython: In which projects are you involved at the moment?**

Paul: Organizing my mammoth pile of travel expenses. :^) I just finished a period with lots of travel, plus helping organize two conferences (OSCOM 3 and EPC2003) that were a month apart.

I'm now getting my sights set on the real Zope Europe stuff I need to do: producing quality marketing material for decision makers, increasing my speaking/interview

schedule, and formally starting the discussion about a business network. If anybody has suggestions about what should be on ZEA's radar, come grab me in Charleroi and let's talk.

I'm also in deep with the Plone community, again, not in a code sense. Plone is to Zope what Zope is to Python. The Plone folks really have gathered up a diverse, active community, one which is in some ways bigger than the Zope community.

Finally, I spend a lot of time working with OSCOM. I'm trying my hand at a small software project under the OSCOM umbrella. Since I complain a lot about developers, perhaps I'll gain a little sympathy and perspective. :^)

**EuroPython: What do you like best about Zope?**

Paul: The people. I don't think an outsider should judge an open source project based on the software, as software changes. Instead, look at the people, how well they work with each other, how strong they feel about the project, the quality of their ideas, etc.

The world of Zope is, like the world of Python from whence it came, a wonderful place to hang out. Someone from the PostNuke community sent me a wild anecdote about how mean people can be in other communities.

**EuroPython: What do you not like about Zope?**

Paul: I think Zope needs to try harder to play nice with others. Instead of inventing our own answers to some things, sort of a NIH, we should make it a higher priority to leverage existing work.

Through OSCOM I spend a lot of quality time listening to the non-Zope universe. You can learn a lot by listening, about strategic things like barriers to adoption.

Some folks have heard me talk about "10x", the idea that Zope should grow market share tenfold in the next year. It seems that, of that 10x, the 9x ahead of us think differently than the 1x we already have. We have to work harder to care about 9x, if 10x is a high priority.

**EuroPython: Do you think Zope 3 will keep the Zope users of today? Or do you think some might search for another tool because of the changes in Zope 3?**

Paul: Very good question. Whenever you start over on a mature system, you introduce a disruption. I think that Zope 3 is shaping up, though, to make component developers very very happy. What's critical is that something like Plone come along, atop Zope, that covers the "site developer" audience.

If you're a Python guy and Zope 2 didn't float your boat, give the Zope 3 developer tutorial a try. The tutorial shows that you can write all your Python code without introducing any Zope stuff into it. The Zope stuff is asserted outside your code, in configuration and interface files.

Another important, but less obvious point, about Zope 3: it is developed by the community. Two years ago, ZC was the only ones with CVS access. Now, the majority of people and checkins are outside ZC. This will have both obvious and subtle impacts, and marks the growing maturity of the world of Zope.

Thus, Jim Fulton has pulled a double miracle on Zope 3: he conjured up a better component architecture, plus he birthed a new culture in the process. Jim should be quite proud of both.

At a minimum, Zope 3 will fix the mistake we made in Zope 2, where we aimed at too wide an audience.

**EuroPython: Have you ever looked at Twisted, Siena Web Services Architecture, CherryPy, ...? How do you compare all these web application tools?**

Paul: Unfortunately no, though since I'm not heavy into development, I wouldn't really appreciate what I was seeing. :^) However, I do talk to all of the people, and have a lot of appreciation for their efforts. Especially Twisted. It seems like Zope has done a dance with Twisted but still hasn't asked it out on a date.

I'd like to see the various projects enjoy each other's success more. Some stuff has been written about Zope that, IMO, crossed the line between healthy criticism and mean-spirited rhetoric. The latter doesn't help us learn from each other and doesn't really fit the Python culture. Plus, ZC has done a lot for Python, with some aspects being quite measurable.

Actually, through OSCOM, I spend more time these days working with, and chatting with, content management system projects that aren't Zope or Python.

**EuroPython: Have you at OSCOM seen some applications from which Zope could learn a lot?**

Paul: Very much so. For instance, other projects are in their second generation of XML support in the core. It doesn't appear that Zope 3 will have XML in the core, except for ZCML. I also think that the pipeline approach to template processing (Cocoon and AxKit, as well as Sean McGrath's Propylon) is a nice model.

In reverse, I think that WebDAV is glue for content management, and I'm working with the other OSCOM "Band of Brothers" to improve DAV support. Clearly Zope is far, far ahead on this front.

**EuroPython: Does that gives you a feeling of uncertainty about Zope's future? Will you be able to achieve your goal if it is missing some features which competitors do have?**

Paul: I sort of see it in reverse: there's something that needs to get done and we need to find a way to do it. Fortunately there are people interested in working on such a thing.

**EuroPython: Why the emphasis on OSCOM? Isn't that helping the competitors?**

Paul: Good question. I don't think code alone is the primary factor for Zope's future growth. In my opinion, there are factors at the macro-level: alliances, interoperability, and telling a story that the mainstream is ready to hear.

I think that open source content management must work together to educate the market about the key selling points. None of us can tackle that job alone. We need to validate open source as a mainstream option in content management.

Finally, there are two practical points. First, I think we owe it to our customers to make ourselves replaceable. Open source should actively fight, not passively promote, vendor lockin. Second, it's clear that OSCOM is a hot story that mainstream journalists want to write about, and goodness knows we aren't doing enough regular publicity.

**EuroPython: Finally, what do you expect from EuroPython?**

Paul: Chimay, the official beer of Python. Also, I want to leave with feedback from Zope companies about what Zope Europe should do in 2003.

Interested readers can find more info at:

<http://www.zope-europe.org/>

<http://www.europython.org/>

<http://www.zope.org/>

<http://www.python.org/>

<http://www.twistedmatrix.com/>

<http://www.siena-tech.com/>

<http://www.cherrypy.org/>

<http://www.bluedynamics.com/>

<http://www.oscom.org/>

<http://cocoon.apache.org/2.0/index.html>

<http://axkit.org/>

<http://www.propylon.com/>

<http://www.europython.org>

## Interview of Martijn Faassen

Tom Deprez

**EuroPython: Martijn, please introduce yourself to our readers.**

Martijn Faassen: I'm from the Netherlands, and co-run (with Kit Blake) a small software development company here, called Infrae (<http://www.infrae.com>). My interests are fairly diverse, in a geeky way. I like computer programming, history, philosophy, strange humor and games, for instance. Since I've been fairly active in the Python and Zope communities for some years quite a few people in those communities know about me.

**EuroPython: You are the chairman of the Python Frameworks track. What can we expect this year and did you learned something from last year?**

MF: Last year I ran both the Python Language and Python Applications track, besides helping out to organize the conference itself. This was too much for me to juggle, so this year I've taken it easier. Though I still have helped some in the general organization, my main task is to run the Python Frameworks track. There are many frameworks from small to large in Python to do a wide range of things. One important reason we have so many even with the comparatively small resources of the Python community is because Python makes it so easy to develop frameworks.

The track has talks ranging from discussions of GUI toolkits to XML processing frameworks. And of course also a number of talks about the famous Twisted. Two of these latter talks are part of the special Moshe Zadka track we seem have at the conference. If they are canceled that would be as he will have collapsed with exhaustion by day 3. The track being about frameworks means many of these talks will feature some tutorial style content -- frameworks are meant to make your life easier, and the track's here to help you learn how.

**EuroPython: Can you tell us more about Infrae? Where is it situated? The philosophy, ...?**

MF: We're a small company based in the middle of Rotterdam, the Netherlands. For the last half year we've been hovering between 4 to 6 people working here in the office, not including Torvald, a big norwegian forest cat (Kit's cat). Before that we were smaller. We also have a wide network of freelancers within the Python and Zope worlds that we draw on occasionally.

Our philosophy is evolving, but it definitely involves us being active in the Python and Zope communities. We like the community, and it's a very useful network to draw on for business reasons as well. We produce open source software, in a customer-driven fashion. If a customer wants something and pays for it, we create or extend a system to do it. Of course we try to keep the new development general enough to be useful for others as well. So far that has worked quite well for us, probably partially because many of our customers are from the academic sector and may be more in tune with open source. The whole open source business strategy is a permanent experiment, though -- we'll see how it develops over time.

Since we like developing new things and would like to improve the foundation, we've also put quite some time in projects like Zope 3 and Forest, so we're not completely customer-driven.

**EuroPython: What's your relationship with Python and Zope?**

MF: My main contributions to the Python community are me helping to organize EuroPython and my very wacky postings about the PSU that I deny any knowledge of to the `comp.lang.python` newsgroup. I haven't been contributing a lot codewise, though there are a few little modules here and there now.

I've contributed significant amounts of code to the Zope community over the years though. Well known in the community is Formulator. I also maintain a few other Zope products, such as ParsedXML. And in the last year and a half I've been the lead developer for Silva, Infrae's structured content management system based on Zope (<http://www.infrae.com/products/silva>).

I make my living using Python and Zope in the context of Infrae. Infrae is an open source company and we like the community aspects a lot.

**EuroPython: Can you tell more about Silva, the product behind Infrae?**

MF: Silva is a content management system, even though Kit would prefer to use another term as 'content management system' is so vague. So we say Silva is an 'authoring and publication system'. Silva is focused on managing large quantities of structured content. This content can be presented within a web site, but can also be exported as XML and transformed (for instance to MS Word). Since we are aggressive about keeping the content entered into Silva clean and structured it is future-proof and media-independent.

Silva can be used by just a single user, but it shines in large organization, as its tree based structure is designed to scale. Silva can be hooked up to user authentication systems like LDAP and Silva's user interface makes it easy even large quantities of users. Silva also has a multi-version workflow, an extensible metadata system, and much more.

By the way, while Silva is very important for Infrae we're definitely also interested in doing other Python or Zope related development (hint hint). We might find a way to integrate it into Silva though. :)

**EuroPython: What made you decide to build your own Zope framework system, except starting from existing ones, like CMF, CPS, Kontentor, Plone?**

MF: We started building Silva innocently and unambitiously back in late 2001. While CMF had been around, it had been in vaporware stage (portal toolkit, etc) for so long I was wary of going with it then. Of course it matured a lot since then and gained tremendous momentum, so it became more attractive. Then again, going your own way has its advantages too, as you can determine your priorities (and fulfill those of your customers) yourself.

I don't think we'd heard about either Plone or CPS yet; I'm not sure whether either was around at the time. An early version of Kontentor was actually around I think. Joachim Werner (main developer of Kontentor) told us that it is actually inspired by some of the same ideas Silva has, as both derive conceptually from a very early implementation of XMLWidgets I did in 1999/2000 or so. But still our ambitions weren't so high, and we had our own ideas...

Over time of course Silva quickly developed and became more ambitious, driven by customer requests. We generalized a number of the components more and more, and the user interface has received endless polishing over time.

**EuroPython: Which customers are using Silva? Can you point us to some websites as examples of use?**

MF: The biggest 'sponsor' of initial Silva development was the Erasmus University here in Rotterdam. They use Silva for parts of the website, but also and especially use Docma to export that same information to study guides that are printed on paper. This is an example of a Silva site at Erasmus:

<http://web.eur.nl/master>

Another large contributor to Silva development is ETH in Zurich, Switzerland (Swiss Federal Institute of Technology). They run quite a number of Silva sites at various departments, among others:

<http://www.mat.ethz.ch/>

<http://www.vc.ethz.ch/>

The Materials site has a page on Silva there as well:

<http://www.mat.ethz.ch/silva>

And I just found out from Kit they have a Silva course, which looks pretty neat!

<http://www.mat.ethz.ch/silva/course>

Another customer that uses Silva is the Bijvoet Center of Biomolecular Research at the University of Utrecht.

<http://www.bijvoet-center.nl/glycotrain>

Besides these and other customers there are also a quantity of Silva sites that we may not even know about (though we know about some as the developers post on our mailing lists). Because it is so easy to plug in completely different layout templates in Silva, it is hard to recognize a Silva site even if you run into one!

**EuroPython: What is Formulator?**

MF: Developing web applications means you have to write a lot of web forms. Being a conscientious programmer, I kept worrying about validation of the user input into the form, and kept finding myself very similar validation functionality over and over again (is this field required? not too long? etc). So being also a lazy programmer I started writing a Zope system to make the validation and also generation of web forms easier. I also tried very hard to make it fit inside of Zope's through the web user interface.

Apparently I did it well enough for people to decide to put Formulator in their own Zope toolchest. I regularly get mail from all over the world from people who use Formulator. We at Infrae use Formulator in many places ourselves, and currently the requirements of Silva tend to drive Formulator development the most.

In Zope 3 the Formulator concepts evolved further and this turned into Zope 3's Schema and Forms. The core of this code was written by me, though now it passed through many hands and underwent many changes. Zope 3's schemas and forms are used throughout the Zope 3 codebase.

**EuroPython: When did you hear for the first time about Python and Zope?**

MF: I discovered Python somewhere in '98, though I heard about it first some years before that, I think. At the time I was programming a lot in C++ and Delphi, and I was looking for a scripting language to supplement those. I messed around with Perl for a week or two and while I could recognize its power I couldn't get it into my head.

So one night I ended up on [www.python.org](http://www.python.org), and downloaded and printed Guido's tutorial. I read it before going to sleep, and the next day I could program in Python. Inexpertly, but I soon fixed that, and I fell in love with the language. Quickly forgot about Perl, and didn't do much anymore with C++ or Delphi either.

I went to work for the University of Utrecht only a few weeks after that and had the great fortune that I could introduce and use Python there right away. Since then I've been making my living programming in Python!

Then when Zope was open sourced somewhere around the end of 1998 I started using it for work soon after. In 2001 I and Kit Blake founded Infrae, a Python and Zope development company.

**EuroPython: On which projects are you working at the moment?**

MF: Silva and associated projects take most of my time at Infrae right now. My task is to guide the other developers on the project in their activities when it comes to the code. Recently the Silva user interface got a big overhaul both codewise and in its look and feel. We've also added an extensive metadata system and improved the performance of various parts of the user interface and public pages significantly. Silva 0.9.2 is released in beta form now and it offers a vast amount of new features compared to the 0.9.1 release -- please check it out! We also have a system called Docma which allows for integration of Silva with Word; it can both export Silva content to Word as well as import Word content into Silva.

My biggest new project is Forest, the XML database I will be talking about at the conference. It can store and index arbitrary XML data and has an implementation of the XPath query language. My dream is to use it as a storage backend for Silva's XML data at some point in the future, when we have ported Silva over to Zope 3.

Another new project I'm working on is an OAI-PMH client for Python, along with Zope integration and Silva integration. OAI-PMH stands for "Open Archives Initiative Protocol for Metadata Harvesting", an enormous mouthful I love to roll off my tongue like it is star trek technology doubletalk. It's an XML over HTTP protocol for the harvesting of metadata about resources, such as publications.

A university for instance has a lot of people publishing scientific papers. If they have a server serving out OAI-PMH they can expose metadata about these papers to external harvesters (title, author, etc). Other people can then build applications using the harvested information. We've built a basic harvester (a Python module), and we're also building applications on top of it -- Zope integration using the ZCatalog as well as Silva integration.

I'm not working on these projects alone -- Silva especially has had many different people contributing to it, both within Infrae and through the open source process. Working in a team is a lot of fun, as everybody has their own strengths. I for instance suck at user interfaces, but Kit and Jan-Wijbrand are great at them.

I'm also very interested in Zope 3 and have contributed some things to it; I kicked off the development of Zope 3 schema and have recently been active in the integration of XML facilities into Zope. And in december 2002 Infrae organized the Zope 3 'Sprintathon' here in Rotterdam, which was generally thought to be very successful.

**EuroPython: Can you tell us more about Forest? How far are you in the development of it? Will it be a stand-alone product?**

MF: Forest is a stand-alone system, but it does currently require ZODB4 (which hasn't been released yet but is developed along with Zope 3). It should be easy enough to port it over to ZODB3 and it shouldn't be too hard to port it over to layer over other data storage technologies like Berkeley DB or Metakit.

I intend to provide tight integration with Zope 3 when the time comes, however.

Forest is far enough along to be able to store large XML datasets and it support a large part of XPath. It's not far enough along yet to be ready for production use, and I intend to do a lot of performance work on it still as well. It \*is\* far enough along for other people to drop in and join in with the fun, however. If you are interested in XML, query language implementation and databases then do drop me a mail. :)

For more about Forest please come to my talk!

**EuroPython: Only one question on this subject, you talk about an XML database, however, you use the ZODB or other data storage technologies as backend. Can you then still talk about an XML Database?**

MF: Forest uses an underlying data storage technology such as the ZODB and its BTree facilities. It however creates its own indexes and datastructures optimized for efficient queries of XML data in particular. This is similar to the way the ZODB and use Berkeley DB as an underlying storage, or to the way the Zope Catalog adds indexing facilities and querying facilities to the ZODB. Another example is MySQL, which can use Berkeley DB as well to offer a transactional storage. So I guess "XML Database" is still the best word I have for this. :)

**EuroPython: Is Silva already preparing the port to Zope3? What do you think of Zope3? Will people still be able to create the same projects as they do now or is the new version more heading to Content Management Systems?**

MF: Zope 3 is definitely general enough for any web application, not just content management systems. It pushes the Zope 2 concept of combining building blocks into applications much further still with its component architecture. As a Python programmer Zope 3 is very exciting. The Zope 2 gods require a large amount of concessions and sacrifices in Python code. Zope 3 has been designed to create a minimal burden to the Python programmer instead -- I can just write Python and \*then\* add all the stuff separately to make it work in the framework, such as a user interface. It also should become easier to combine different components into one application.

Infrae has not started porting Silva to Zope 3 yet, but we are actively planning for it. Since Silva content can all be exported as clean XML, it will be relatively easy for us to port over content from Zope 2 Silva into Zope 3 Silva. Silva is also 'prepared conceptually' to bridge the gap more easily -- we split views from content for instance, and use services throughout. That said porting to Zope 3 will

still be a significant effort. Customer contributions are welcome. :)

**EuroPython: How do you handle the view-content split in Silva?**

MF: It isn't too dissimilar from the way the CMF Skins mechanism does it, though we have different advantages of disadvantages. Silva since the fall of last year actually uses the same technology as CMF does to store Page Templates and Python Scripts on the filesystem instead of in the ZODB, which makes for much easier management of the code in a version control system like CVS. This functionality has been extracted from the CMF core by Kapil Thangavelu into a product called 'FileSystemSite', which we use.

Silva's content objects are implemented as classes in Python, in the normal way you write a Python product for Zope. Silva's views are in a separate 'view' hierarchy, which is full of page templates and python scripts, along with some XML serializations of Formulator Forms. An individual view is actually a Zope folder in the view hierarchy (but this can be treated as a directory on the filesystem thanks to FileSystemSite). As a view is a Zope folder it acquires the content of Folders higher up, faking a simple kind of single inheritance in views.

Each content type normally has three types of view associated with it -- an adding view showing an add form, an edit view showing an edit form for the author (including tabs and the like) and a public view that determines how the content is rendered to the end user of the site (though this not include layout or navigation, which we add later outside the view system). Views are associated to content objects in a view registry. The Silva UI is composed mostly out of edit views with some add views, and the public website is composed from a few layout templates placed in content space along with the public view of the objects.

This model/view architecture allows us to easily vary the two independently from each other. We rewrote large parts of the view code in the recent months, but the content code did not change that much, for instance.

The CMF skin mechanism is quite similar. The difference is that CMF Skins are a stack while Silva views are a tree. A disadvantage of CMF skins in my mind is that a single skin can contain views for any content object, which can make things rather messy if this isn't organized very strictly. Also there doesn't seem to be as easy a way to make separate public views, but I may be mistaken there. An advantage of the CMF skins compared to Silva views is that the extension mechanism of CMF skins is more flexible. It is easier to insert a skin in a stack than it is to graft a different sub-tree onto the main view tree in Silva, though we do manage to do it. :)

**EuroPython: What do you like best about Python?**

MF: It doesn't constrain me as a programmer. I can generally express what I need without having to jump through hoops to make it work. It's concise but not at the expense of readability. It has just enough syntax; not too little and not too much. Python is a very "balanced" language, its main strength is that it does not have a main strength at the expense of other things.

**EuroPython: What do you not like about Python?**

MF: Aside from a few very minor nits here and there the main thing I would like to see improved is the relatively slow performance of the current batch of interpreters. While for the majority of tasks Python is fast enough, and while one can always escape to extensions in C or some

other speedier language, it would be nice if my Python code was blazingly fast as well. This is why I'm very excited about projects like Psyco and Pypy, which aim to improve performance. But Python's plenty nice enough right now for me to wait patiently for these to mature.

One area of Python that I'm particularly interested in is the XML infrastructure so naturally I have suggestions on how to improve it too. :)

**EuroPython: What is the next 'big' issue Python should have, according to you?**

MF: Increased performance would be my best candidate for the next 'big' thing for the language implementation itself. I don't have any suggestions for feature additions to the language itself; I'd prefer it if that remains relatively stable. On the other hand, it's always fun to experiment with new approaches...

One thing that will continue to happen is the creation of intriguing new libraries and frameworks in Python. Just witness the rise of Twisted in recent times, for instance. It's hard to predict what piece of Python software will be next making it big, though clearly I hope Zope 3 will be. Overall there seems to be a focus on the construction of larger application frameworks in Python recently. I suspect one enabling big change for that is the growth of unit testing in the last three years or so within the community. A good test harness allows frameworks to become more solid and more ambitious.

**EuroPython: You've mentioned Zope and Twisted. How do you compare these two? Do you know of the Siena Web Services Architecture, mentioned in a previous interview here at EuroPython?**

MF: I haven't had the opportunity to play with Twisted yet, though we do use it in our Docma server application. Zope has a focus on content and meta-content building blocks, all wrapped up in a user interface. Twisted focuses more on the lower level network architecture issues. Twisted is more for Python programmers while Zope can also be used to good effect by other people such as designers and people who program incidentally as part of another task.

It is my hope that eventually someone will step up and integrate Twisted and Zope 3, so that we can gain the best of both worlds. There have been some smaller steps taken in this direction already, and the good intent is there, now it only needs to happen.

I hadn't heard about the Siena Web Services Architecture until I read about it in the interview with Marc-Andre. :)

**EuroPython: Do you have a general view on how Twisted and Zope3 could be integrated in each other?**

MF: Just a general view only. Zope 3 like Zope 2 currently contains its own networking code to handle things like HTTP and FTP. Twisted thrives on implementing networking protocols like this. If Zope 3 integrated well with Twisted we'd get support for a \*lot\* of protocols for free, and the Twisted people would maintain them. The Zope 3 people could then focus on the aspects of Zope 3 that do make it special, such as the component architecture and exposing it through its user interface.

Then again, Zope 3's HTTP server works just fine, so there is no urgent need in the Zope 3 community to start this integration activity. We may feel the need more when Zope 3 gets big customer deployments; Twisted may help it scale better.

**EuroPython: You like games, computer games? Have you looked at PyGame? Do you have an interest in creating one day your own computer game?**

MF: I like games a lot, both computer games as well as non-computer games. I have been running an RPG campaign for friends since the mid-nineties now, but we only find time once every two or three months on average to get together to play. I used to play more computer games, but I have found less time than I might wish to play them after starting Infrae. Infrae's lots of fun too, though. :)

One computer game I like a lot is Europa Universalis II. It's a strategy game that simulates world history from about 1400 - 1800, and is quite detailed.

I've been tracking PyGame since the project started, and played with it once every while. And yes, one of my ambitions since I first encountered computers when very young was to create my own games. Over time I've created a few half-finished attempts, and it is how I learned to program; I imagine I'm not the only programmer who followed this route. And it's still one of my hobby-ambitions to create some computer game one day. But then I have many such projects, so you may have to wait a few decades. :)

**EuroPython: Python and history simulations... You might have a look at Civil (<http://civil.sourceforge.net/>) and start from there. As a start you could add a European theatre :)**

MF: I am aware of Civil, though I haven't tried it yet. Civil seems to be more about individual battles, while Europa Universalis is of a really grand strategic scope indeed. Battles are entirely abstracted, but it does feature intricate alliance diplomacy, new countries forming (the Dutch rebellion from the Spanish is in there for instance), and the reformation. I like open ended simulations. I also enjoy artificial life simulations of evolution for that reason (and I've written a few unpublished experimental ones in Python).

I have been vaguely considering writing a game of that type in Python someday. But that's just a thought, which is many steps away from a project. Again, a few decades perhaps... :)

**EuroPython: Can you reveal the secret around 'Python Secret Underground'? It might be difficult -since you aren't part of it- of course, but you might at least have heard something about it. :)**

MF: I have no idea what the Python Secret Underground even could be. I never heard of it. However if they existed they might be an ominous all powerful organization, as they would have access to Guido's time machine. This would give them the power to roam across time and space and branch of alternate timelines with abandon. There would be a rumor that in one perverse experiment they branched off a timeline where Python did not exist, probably just to see what would happen, at least that would've been my reason to do that, they might've been more serious.

In that timeline, Zope ended up being written in some other language by Jim Fulton who shortly thereafter went entirely mad. He was locked up in a lonely monastery in Belgium where they brewed Chimay. On calm dark nights his cries "the braces, oh no, the braces!" could be heard from kilometers away. Incidentally Guido ended up there as well, as one day in some newsgroup he proposed a language that would do block structure by indentation

only. Kind people reported him to the police and they kindly put him in the same monastery. I myself happily discovered Lisp instead of Python, and therefore grew more and more eccentric over time. I ended up founding an entity called the Flee Software Foundation, and giving speeches at conferences about how we are all doomed and we should flee software before it is too late. There also is a song I sing about it (don't ask).

People just didn't know any better in that timeline. The sad end was when they equipped all their missile launching systems with their timeline's version of Zope. Unfortunately the missiles implicitly acquired the wrong targets and humanity was annihilated. Even worse, we were all using other languages and there was no EuroPython.

This is all entirely rumor though. I disclaim any knowledge of such matters, as the PSU does not exist.

**EuroPython: Finally, what do you expect from EuroPython?**

MF: Besides learning about all the interesting projects going on I hope to meet new people as well as reacquaint myself with those I met before. It wouldn't hurt if we could drum up some more business for Infracore as well!

If you read this far, thanks for reading my incessant babbling! It's all Tom's fault, he kept asking more and more questions! Actually thanks Tom for all your effort. :)

Finally, I won't be at any Python Secret Underground meetings at EuroPython, as there is no such thing as the PSU and there is certainly no plan to hold any PSU meetings on the Open Spaces track. Meaningless non-code phrase follows, please ignore: Today, Torvald does not like the banana. Fnord. Repeat: Today, Torvald does not like the banana. Fnord.

<http://www.europython.org>

People who would like to find out more about the mentioned topics, have a look at:

<http://www.infracore.com/>

<http://www.zope.org/Members/faassen/>

<http://www.openarchives.org/>

<http://www.eur.nl/>

<http://www.europython.org/>

<http://www.twistedmatrix.com/>

<http://www.python.org/>

<http://www.zope.org/>

<http://www.siena-tech.com/>

<http://civil.sourceforge.net/>

<http://pygame.org/>

<http://www.europa-universalis.com/>

## Interview of Duncan Grisby

Martijn Faassen

**EuroPython: Duncan, please introduce yourself to our readers.**

Duncan Grisby: I'm best known in the Python world for my work with CORBA. I'm the author of omniORBpy, one of the CORBA implementations for Python; I also maintain omniORB, the C++ part.

Professionally, I currently have two main things that occupy my time. I own a company called Apasphere which offers consulting, teaching, and commercial support for omniORB. I am also the co-founder of a startup company called Tideway Systems which is working on software to help large companies manage their IT infrastructure by understanding dependencies between components. At Tideway, we're making heavy use of both Python and CORBA.

Outside of work, I'm a keen photographer, my favourite subject being black and white figure studies. I also enjoy kickboxing, and I'm the assistant instructor at the local kickboxing club.

**EuroPython: What is interesting about distributed systems?**

DG: The main thing is that they are very hard. As soon as more than one computer is involved in an application, there are many more things that can cause it to go wrong, including network partitioning, data loss, and concurrency. Getting it right, and building robust and efficient distributed applications is a challenge. I like challenges.

**EuroPython: What first attracted you to CORBA?**

DG: In 1995, I started a PhD in Computer Science at Cambridge University, sponsored by Olivetti and Oracle Research Ltd (ORL). At that time, a group at ORL had just started developing what became VNC. Although VNC is great on a local area network, it isn't so good in a wider area, where no matter how much bandwidth you have, latency gets you. I started thinking about building a whole new windowing system that would let you move your desktop from one place to another, without suffering the effects of latency so much.

At that time, CORBA was just coming to maturity as a middleware platform that had a classical object model, the possibility of high performance, the ability to interwork between programming languages. Even better, one of the researchers at ORL, Sai-Lai Lo, was working on an ultra fast and compact CORBA 2 ORB in C++, called omniORB. CORBA, and omniORB, was therefore the obvious choice of platform upon which to base my PhD work.

I still like and use CORBA because it is still by far the best solution to a wide range of distributed systems problems.

**EuroPython: What came of the windowing system you were thinking about?**

DG: I never intended to make it a complete system, just enough to experiment with, and prove the things I wanted to prove. I did develop it enough to build some simple applications, but to make it a complete window system would have required a huge amount of work. I thought about releasing it as an open source project, but I decided that it would cause me to get sucked into dealing with it into the future, which I didn't want to do. Apart from

anything else, a lot of the code was far from robust, since it was never intended to be used for real, so the first step would probably have been to throw everything away and start again doing it properly. Mainly, I think the world's existing window systems are too entrenched, so trying to make a new one is almost certainly doomed to failure.

Anyone who is really interested can read my PhD dissertation, linked from <http://www.grisby.org/research/>

**EuroPython: Are you aware of Fresco (used to be called Berlin)? They use CORBA in the core of their windowing system. What is your opinion on it?**

DG: Yes, I know about Fresco. Berlin, as it was then, was just starting its second complete development team and technical overhaul at the time I started my PhD. I had quite a lot of discussions with the people working on that when I was working on my PhD, but they had a rather different view of how to structure their system than I did. Since that time, as far as I know, there has been another gradual but complete change of developers, and another change in direction, resurrecting the original Fresco system, which started out as a component framework on top of X.

The other connection to Berlin / Fresco I have is that they are using omniORB as their CORBA implementation, and I have implemented a couple of small omniORB features to help them out.

I think Fresco has two main problems. First, they make rather too many CORBA calls, meaning performance is a difficult issue for them. Second, they have the same issue I mentioned earlier, that no matter how great their software ends up being, they will have a very hard time competing with existing window systems, X in particular.

**EuroPython: What motivated you to start developing OmniORB?**

DG: Well, I didn't start omniORB -- Sai-Lai did. In fact, Sai-Lai started omniORB 2. omniORB 1 predated the GIOP standard, and only interworked with Orbix. Anyway, omniORB was written because at the time, ORL had some small custom-designed hardware devices and wanted an ORB to run on them. It was only once it was written that everyone realised it was better than the other ORBs available, and we started using it on mainstream platforms.

I started working on omniORB when I encountered some things I needed for my PhD, which weren't yet implemented. When I finished my PhD, I was offered a job at ORL. By the time I started, ORL had become AT Laboratories Cambridge. Because of my previous work on omniORB, the natural thing for me to do was to join Sai-Lai working on omniORB. A while after that, I became the project leader, and I have continued in that role since AT closed the Cambridge lab last year.

**EuroPython: How did you first get involved with Python?**

DG: Each month, one of the projects at AT gave a review presentation to tell everyone about their progress and show some demos. In one of the CORBA project reviews, we built a distributed music player that streamed CD audio over the network via CORBA. It was controlled with various interesting devices we had access to, but we needed a GUI to finish it off. I'd been wanting to try out



Python for a while, so I decided to make the GUI with that, using a CORBA implementation for Python called Fnorb.

Within about a week, I had learnt Python and built quite a complex GUI using PyGTK, which controlled the rest of our CORBA demo via Fnorb (adding threading support to PyGTK along the way). After that, I was hooked on Python, but I was disappointed with some aspects of Fnorb so I started writing omniORBpy.

**EuroPython: In what areas do you think Python could be improved?**

DG: Nothing significant, really. I'm happy as it is.

**EuroPython: There is a lot of buzz concerning SOAP and web services recently. Could you compare and contrast with CORBA?**

DG: The software world is largely driven by hype. At any one time, there are usually one or two "hot" technologies that are hyped enormously, and are generally proclaimed to be the solutions to all problems. As part of this, the standard practice is to rubbish all other technologies that are remotely similar to the one being hyped, in order to convince the world that the technology being hyped is indeed the solution to all problems.

Web services and SOAP are currently occupying the hype spot. (Before that, the hot technology was Java, which was going to render all other languages obsolete. Remember that?). There is some overlap between SOAP and CORBA (but not as much as some people say), so CORBA is getting the rubbishing treatment.

The main technical difference between SOAP and CORBA is that SOAP is procedural or message oriented and CORBA is object oriented. The distinction is exactly the same as the difference between procedural languages and object oriented ones. For some applications, a procedural style is the easiest to use; for others, you really want object orientation. Another significant difference is that SOAP messages are XML text, while CORBA uses a binary protocol. Text is possibly easier to deal with when debugging (although tools like Ethereal will decode CORBA's GIOP for you, and have you tried to read a SOAP message!?), but text requires much much more network bandwidth and processor time to handle.

The CORBA specification also goes much further than the web services specifications, in that it specifies an object model, standard language mappings, and a wide range of standard services that applications can use. Web services will no doubt grow most of these things, but they don't exist yet.

The thing I find really sad about the whole web services push is that when CORBA was being designed, the people writing the specifications inevitably made some mistakes. Most of the mistakes were fixed, but a few remain, largely to keep backwards compatibility. The web services people could learn an enormous amount from the mistakes made with CORBA (and the successes of course), but on the whole they are ignoring everything that has gone before, and are making all the same mistakes (and more) again.

To close what turned into a rather long answer, I'm a strong believer that people should choose the right technology for the job in hand. For some problems, the right technology is web services; for others it is CORBA; for yet others it is something else.

**EuroPython: Recently REST (REpresentational State Transfer) has been put out as a contrast against**

**remote procedure call style mechanisms like SOAP and CORBA. What is your opinion about a design philosophy like this, compared to an RPC style system?**

DG: REST fits nicely in the "something else" category I just mentioned. It is definitely a good solution for some kinds of problem. It is, however, most clearly not a very good solution for other kinds of problem. The REST articles I have read tend to fall into the trap of trying to convince the reader that REST is the solution to all distributed computing problems, which clearly it isn't. Having said that, I certainly think that some of the things REST proposes are a far more sensible than some of the most complex parts of the SOAP and related specifications.

**EuroPython: Finally, what do you expect from EuroPython?**

DG: Mainly, I hope to have a really great time like I did last year. It's always fun to meet interesting people from the Python community. The Belgian beer is good too...

<http://www.europython.org>

People who would like to find out more about the mentioned topics, have a look at:

<http://omniorb.sourceforge.net/>

<http://www.grisby.org/research/>

<http://www.grisby.org/presentations>

<http://www.realvnc.com/>

<http://www.apasphere.com/>

<http://www.tideway.com/>

<http://www.fresco.org/>

<http://www.w3.org/TR/SOAP/>

<http://www.w3.org/2002/ws/>

<http://internet.conveyor.com/RESTwiki/moin.cgi/FrontPage>

## Interview of Michael Hudson

Tom Deprez

**EuroPython: Hi Michael, can you introduce yourself to our readers?**

Michael Hudson: I'm Michael Hudson, a 24 year old mathematics PhD student at the University of Bristol in the UK.

I'm chairing the Python Language Track at EuroPython 2003.

**EuroPython: What can we expect from the Python Language Track? Can you whet our appetite?**

MH: I had a difficult time trying to guess what people would find interesting as talk subjects, so I've organized a series of talks that I'm expecting to find very interesting, and hoping everyone else does too. We have talks on extending Python, on bridging Python to other language runtimes, on integrating Python into an operating system, explaining some of the deep and powerful parts of the language, and many others.

**EuroPython: When did you hear about EuroPython?**

MH: Martijn Faassen emailed me in 2001 when he was kicking off the process that led to EuroPython 2002, although I only attended EuroPython last year and this year I'm helping to run it.

**EuroPython: Any peculiar things you still remember from EuroPython2002?**

MH: Not really... Google can find you my diary from last year's conference, which I'd have to read to refresh my memory anyway.

[editor note: For the curious readers: <http://starship.python.net/crew/mwh/europython.html>]

**EuroPython: Ouch, not a good point for EuroPython... If it would have been a great congress, you clearly would have known something.**

MH: Well, you did ask if there was anything peculiar I remembered. I mostly remember a sense of having fun and being interested.

Actually, probably the best bit of EuroPython 2002 for me was Armin's talk on psyco.

**EuroPython: What's your relationship with Python?**

MH: If I remember right, I first encountered Python back in 1997 in an article in Byte, but it wasn't until I went to university later in the year -- and got a fast internet connection right in my room -- that I really learned the language. I submitted a few patches here and there over the next few years, joined python-dev late spring 2000, was given commit rights to Python CVS in 2001 and somewhat accidentally ended up doing the bulk of the work for the 2.2.1 release in April 2002.

Just another student who uses up his free time hacking Open Source software...

[editor note: this article was by jeffrey shell, an original developer of Zope and employee at DC/ZC]

**EuroPython: How can someone accidentally do the bulk of work for a certain version?**

MH: Well, it was a maintenance version, so the bulk of the work was deciding whether a checkin to the trunk of

Python's CVS repository qualified as bug fixes, applying them to the release22-maint branch and so on. I was just monitoring the python-checkins list and porting the occasional fix when the idea was suggested that 2.2.1 should be released fairly soon. That was when a week suddenly disappeared into searching CVS logs for fixes, the trackers on SF for bugs, fixing last minute crashes, editing web pages and so on.

I helped a bit for 2.2.2 as well, but that was a more distributed effort.

**EuroPython: You attend several PyPy sprints. Can you tell us something about this project?**

MH: "Several" == "Two" == all the sprints we've had so far

PyPy is an attempt to implement Python in Python, which is an idea that appeared from time to time. The current project apparently emerged from a discussion on the German Python list, particularly between Christian Tismer and Holger Krekel -- both of whom I met at EuroPython last year.

Holger set up a mailing list and a subversion archive and so on and organised the first sprint in Hildesheim back in February. The second sprint (which I am actually at now as I type this has been organised by Laura Creighton and Jacob Hallén in G?thenburg and there's a third sprint planned for just before EuroPython.

**EuroPython: What do you expect from this project?**

MH: Firstly, another implementation of Python. We hope that this new implementation will be more flexible -- enabling tweaks to algorithms and structures used by the interpreter to be prototyped more easily, for example -- and, maybe, hopefully, faster using techniques somewhat like psyco (although it currently runs something like a thousand times slower...).

**EuroPython: With another implementation, you mean a new implementation built from the ground up?**

MH: More or less, yes. Currently we depend on the extant CPython implementation both as an implementation host and for borrowing bits of functionality that we don't yet implement. But the goal is to achieve a self-hosting system independent from any existing implementation, yes.

**EuroPython: At the end, if the PyPy project will succeed, will it mean that specialized compilers as Psyco become unnecessary? Are people like the developers of Psyco involved in PyPy?**

MH: The developer of psyco is Armin Rigo and he is heavily involved in PyPy, yes. He's giving a talk on PyPy as part of my track.

**EuroPython: Interesting, so his talk will be really fresh, since the next PyPy sprint is right before EuroPython at Louvain-la-Neuve!**

MH: Yes! Actually I'm expecting quite a few speakers from my track at Louvain-la-Neuve: Alex Martelli, Anna Ravenscroft and Christian Tismer will all be there and are all giving talks in the Python Language track. I will be really unimpressed if they are still writing their talks and not hacking PyPy.

**EuroPython: How will PyPy handle platform differences? Or isn't this an importance for PyPy?**

MH: This is not yet much of an issue. People are using Linux, Windows and Mac OS X to develop PyPy, so a certain amount of portability is likely to be maintained.

For now, we basically inherit portability from CPython. We will have to worry more when we attempt to make the implementation free-standing.

#### **EuroPython: How does the project get along?**

MH: Reasonably well, I think. Currently, pretty much all the work that has been done has been done at the two sprints, at least partly because the whole system architecture has been in a state of flux and it's easier to have the courage to break everyone else's code when you can explain to them in person what they have to do to fix.

Hopefully by the end of the current sprint, or maybe the next one, which is only a few weeks off, we will have reached a point where there are multiple, largely independent areas to work on which suits the usual mode of open source projects much better.

#### **EuroPython: So, after these initial sprints, the PyPy initiators hope to attract more people in the project? Which independent areas does PyPy have in mind? Does the PyPy project have already an official website?**

MH: That's the idea, yes. As to which areas need work, I fear this margin is too small to contain an explanation.

There is a website: <http://codespeak.net/pypy/>

#### **EuroPython: On which other Python projects are you working at the moment?**

MH: Well, I do some work on CPython itself. We're coming up to the release of version 2.3 and so are in bugfixing mode for that.

I also have written a readline-a-like, pyrepl, that occupies a certain amount of my time, but this project has reached the point where it is good enough for the purposes I originally wrote it for, so I don't do so much work on it any more.

Other than that, I use Python for a bunch of unreleased and unreleasable tasks, from analyzing and creating plugins for Escape Velocity Nova to things other people probably use their desktop calculator for.

#### **EuroPython: Why do all these different Python implementations exist, e.g. Jython, CPython? Is there a big need for them? I can imagine that it confuses some beginners in the Python language.**

MH: Well, the ultimate goal of PyPy is world domination. Eventually we want to have the best, fastest, most flexible implementation of the Python language around.

OTOH, that's certainly not what we have today, so any newcomer to the language who thinks that PyPy is the implementation she should be using is clearly getting her information from the wrong sources.

I don't see much Jython/CPython confusion out on comp.lang.python, so I'm not too worried about this aspect. We need to maintain very good compatibility with CPython, and that's a goal.

#### **EuroPython: Can you tell us more on pyrepl?**

MH: Well, I describe it as a "library for building command-line interfaces". It's similar to the GNU readline library, but with support for multiline editing and a more flexible interface (this latter largely comes for free with being implemented in Python, not C).

#### **EuroPython: Will pyrepl one day be implemented in the python core? Or is there a reason why it shouldn't? Looks a nice improvement to me.**

MH: pyrepl probably isn't sufficiently mature or portable to be part of the Python core as yet, but one day it would be nice.

#### **EuroPython: Erm, "Escape Velocity Nova"?**

MH: It's a game by Ambrosia Software:

<http://www.ambrosiasw.com/games/evn>

One of the problems of buying an Apple laptop was putting me back in range of these guys.

#### **EuroPython: What do you like best about Python?**

MH: I think, probably, the single biggest plus of Python is that it blurs the distinction between using and programming a computer.

#### **EuroPython: Can you elaborate on that?**

MH: Well, the other day on IRC I was trying to help someone with some file format problems. It turned out that all he needed to do was remove the first 128 bytes of the file, and he didn't know how to do that. If he'd been familiar with Python, this would have been trivial.

#### **EuroPython: What do you not like about Python?**

MH: Occasionally, the lack of syntactic flexibility means that you have to type more boilerplate than you really want to. But the cultural tendency against magical source-compressing techniques overall counts as a plus.

#### **EuroPython: What the next 'big' issue Python should have, according to you?**

MH: I don't know. Like my pyrepl package I mentioned earlier, Python works "well enough". If there were no major changes to Python between now and the end of eternity, this would hardly be a major disaster.

#### **EuroPython: What's your experience with clearing rhododendrons?**

MH: They burn really well.

[editor note: For the confused readers: Have a look on the personal website of Michael]

#### **EuroPython: Finally, what do you expect from EuroPython?**

MH: I expect a lot of really interesting talks, and to meet a lot of really interesting people!

<http://www.europython.org>

People who would like to find out more about the mentioned topics, have a look at:

<http://codespeak.net/pypy/>

<http://pyrepl.codespeak.net/>

<http://starship.python.net/crew/mwh/>

<http://www.ambrosiasw.com/games/evn/>

<http://www.europython.org/>

## Interview of Marc-Andre Lemburg

Tom Deprez

**EuroPython: Marc-Andre, can you introduce yourself to our readers?**

MAL: Most people probably know me because they have used one of the well-known mx Extensions for Python, of which mxDateTime and mxODBC are probably the most used of these out there.

I started working with Python in the Winter of 1993 and immediately fell in love with Guido's design decisions. Later on, when Python's development moved to SourceForge, I was invited as Python Developer and granted CVS write access. Since early 2001, I am also member of the Python Software Foundation (PSF) board and busy working on getting the legal and financial aspects of Python contributions in place.

Ever since I left university, I have been working professionally with and for Python. I first started working as consultant (which I still enjoy) and founded eGenix.com early in 2000, a German company aimed at providing high quality professional skills, products and tools on the basis of the Python programming language.

**EuroPython: Can you tell more about the Python Software Foundation (PSF) board?**

MAL: The board manages the everyday business of the PSF and decides on all major activities related to the PSF's mission, i.e. furthering the use and development of Python and related technologies/activities.

The latest big undertaking of the PSF was to help organize and financially backup the PyCon conference in the US (which happens to be a low-cost conference similar to EuroPython).

Apart from these outstanding events, we are currently looking into getting legal backup for securing Python's IP with the ultimate goal of obtaining the IP for the whole interpreter. In short: try to reduce the size of the LICENSE file.

Most of the tasks are managed offline, but we do have monthly IRC meetings to discuss important issues and to pass motions which guide future developments.

**EuroPython: How is your progress in the work you do for the PSF and are/were there caveats?**

MAL: As with all open source or voluntary work, available time is the single major caveat in working for the PSF. Things are not always moving as fast as they could, but we're still making steady and very healthy progress.

**EuroPython: On which parts of Python are you working as Python developer? Which parts interest you most?**

MAL: Since I wrote much of Python's Unicode implementation building on an initial prototype written by Fredrik Lundh a few years ago, I still maintain most of it. These days I tend not to have much time to actually do coding work, but I try to overlook the general design and make sure that it stays in line with what the original idea behind the Unicode integration.

**EuroPython: Can you explain the original idea around the Unicode integration? It might be handy for some of the developers to keep it in mind when developing.**

MAL: The original idea was to make 8-bit strings and Unicode behave well when they meet each other in the interpreter workings. We had long discussions about how to implement this properly and I think the current decision to use the ASCII encoding as basis for string/Unicode inter-operation was a very practical one. Everybody dealing with XML or similar Unicode aware technology will have come across the infamous "UnicodeError: ordinal not in range(128)" -- that's "explicit is better than implicit" in real life :-)

My suggestion to developer dealing with Unicode is to always work with Unicode in the core of an application and to explicitly apply conversions to the data at the application interface boundaries, e.g. GUI interface, file I/O, etc.

**EuroPython: Last year, you were one of the co-organizers and one of the chairmen of the Python Business track. What can we expect this year from the Business track and what did you learn from last year?**

MAL: I learned a lot in the process of co-organizing EuroPython 2002.

Trying to organize such an event in the same way as you would an open source project simply didn't work out. The difference between working on e.g. Python and conference is that there are hard deadlines, a tight budget, real money and lots of small tasks involved which nobody really likes or cares about, but which still have to be done to make such an event a success.

My feeling is that a bazar style management is not the right approach here. The cathedral works much better, since it gives more control and also assures that unpleasant work is done right. People shouldn't be afraid to get paid for this.

Anyway, it was interesting to work on a conference and I might do it again in the future.

Regarding the business track this year, I think we have a set of very interesting talks covering everything from Python awareness marketing to proven business solutions. The Python market is growing and now is a great time to participate in it.

**EuroPython: Aren't you afraid that paying people for organizing an event like EuroPython would just raise the budget too high? You could fill the gap by attracting sponsors, but at the moment most possible sponsorships seem to be already too high (i.e. not many companies support EuroPython financially).**

MAL: EuroPython is a very valuable event for Python users and companies investing into Python. If you compare the pricing structure to other similar events such as e.g. the ACCU conferences in the UK, you'll find that it would well be possible to pay the organizers for the work they put into the event.

As side effect it would also make the conference more attractive to larger companies because it adds more "value" to the event (the more you pay, the more important it is). These companies could then have a big impact on the sponsoring budget item.

**EuroPython: Isn't the Python community still too small? Is the amount of large capital companies big enough to provide visitors which can pay high entrance fees? Don't you think you would lose**

**another very interesting group of Python developers, e.g. the scientists, small companies, etc?**

MAL: The Python community is not as small as you might think. The fact that only few attendees at EuroPython come from large companies suggests that there is not enough awareness at these companies.

What Python really needs is to grow up in this respect and tap right into these structures. The potential is certainly there, but people seem to be generally afraid of losing the warm fluffy community feeling when opening themselves to IT professionals like IBM, HP, Microsoft, etc.

Other events such as Linux World have shown that this needn't be the case and we could well benefit from their marketing experience which would then open more doors and generate more corporate awareness.

That said, I do think that there's a pricing structure which would benefit all parties, e.g. by implementing ideas like the DevDay we had at the IPC conferences or tutorial days which are not part of the standard entry fee.

**EuroPython: You speak of low awareness of EuroPython to certain companies. Since EuroPython is organized in an open-source-spirit way, we hoped that everybody would put in some effort and the word would have been spread around. Is it your opinion that this didn't happen? This would indeed imply that the present method of organising doesn't work. Or is it just that these companies automatically think that low-budget conferences are not worth it?**

MAL: I don't think that a voluntary organization is a less perfect way of approaching a conference and you all did a great job in managing it and getting the word out to the open source community.

However, in my experience, getting the word out to companies not active in the open source world needs a different approach, e.g. press releases have to be sent out to news sources and press release services, direct mailings help reach those companies which you are particularly interested in, ads in popular magazines can also raise the awareness and credibility, etc. It may even be worthwhile to invite press to the conference. All this adds to the budget, of course.

**EuroPython: Anyway, I don't think the organizers of EuroPython are afraid to expect money for the organization and the time they've put in ;-).**

MAL: Perhaps you should set up an online donation system like the PSF is currently underway in implementing ?!

**EuroPython: That's a good idea ;-)**

**EuroPython: When did you heard for the first time about Python?**

MAL: I found it on an OS/2 Hobbes CR-ROM, whizzed through the tutorial and was immediately convinced that I had found what I had always been looking for (leaving BASIC, Assembler, various Pascals and FORTRAN behind).

**EuroPython: What's your relationship to Python?**

MAL: I use Python and C for just about everything I work on. In general, I use Python for the OO-style works and C for performance. The eGenix mx Extensions are an example of the way I work: performance relevant bits are coded in C and then glued together using Python. Higher level APIs are all coded in Python.

**EuroPython: Is there a general rule when you start using C? Do you first work everything out in Python and then start looking for the bottlenecks, or do you think ahead on these issues, before starting to program?**

MAL: I know what Python is slow at and generally write these bits in C for performance. While the other way around is probably more common, there have only been few instances where I took the approach to implement something in Python and then rewrite it in C. However, I do prototype C implementations in Python.

There are also cases where a C interface is simply more convenient, e.g. in the case of mxDateTime which was basically written so that I could interface date/time values back and forth between mxODBC and Python.

**EuroPython: Can you tell us something about eGenix? When did you decide to start your own company? And why did you choose Python, while other more well-known languages might be more profitable.**

MAL: eGenix is managed as virtual company. I use my contacts in the Python world to find the right people for each project and also enjoy working on them myself.

The company has been in business since early in 2000 and has been active doing mostly project business. Last year we have been moving more towards a product based company and we expect the product revenue share to grow over the next couple of years. Our new Zope product, the mxODBC Zope DA is a good success and we intend to offer more products in that area.

Why Python ? I think that Python offers a much more productive approach to business solutions and so do our customers. It is sometimes hard to convince new customers of going the Python way, but once you talk them into it (mentioning Jython for Java-buzzword compliance usually helps a lot), they find out that they get more for less by deploying this technology.

**EuroPython: How do you convince new customers? Do you use the same arguments the most of the time? Which arguments have the most success?**

MAL: Ok, here's a recipe:

The most successful approach has been to talk in terms of solutions rather than using highly detailed technology presentations.

If you get a feeling that the other knows a bit more than just the usual set of buzzwords, then you can test the waters and try to find out whether they are aware of Python. If they are, then there's a chance that bringing up the Python topic can gain you something. Python is still not well-known enough to not frighten management.

Support, finding programmers, etc. is a big issue.

If using Python as argument for your solution does not fit the situation, it's better to say you're programming C with a smart toolkit on top of it :-)

**EuroPython: How do you handle the 'support' and 'Python programmers' issue? Would a Python Job-market be a good idea?**

MAL: We already have a Python job market. I don't know whether it actually works or how many contacts are made this way.

For virtual companies, support usually means email support, plus second level phone support for resellers. Email is great for delegation and works around the world.

Phone support is expensive, so we restrict that to the absolute minimum.

**EuroPython: By "virtual company" you mean that it is rather small as fixed personnel, but that it is big with consultants? Is a project manageable with people all over the world? What's your experience with this?**

MAL: Whether a project is manageable without direct contact depends on the project itself and on how much trust the customer is willing to put into the quality of the project management and lead. I've found that software projects can very well be organized in this way -- using the same tools that we build upon in the open source world.

The advantage of the approach is obvious: you can access a much greater skill set and have people work on the project around the clock if needed.

Not employing programmers also reduces financial risks for the company itself e.g. in case of economic decline.

TheKompany is a well-known example of a company where this works very well.

**EuroPython: You offer commercial products written in Python, an open source programming language. What's your experience in this field? Aren't people in the open source world more used to paying for consultancy?**

MAL: That is an interesting question. I have found that many people in the Python and Zope world are not very keen on getting paid support (they don't mind getting free support ;-). Fortunately, the situation is a little different for paid products.

**EuroPython: That's an interesting experience as well, I didn't expect this.**

**EuroPython: Aren't you afraid by making commercial products that you lose some of the benefits of open source, such as accessibility to a big audience and a wider group can provide improvements to the code?**

MAL: No. I generally receive very little feedback for the open source parts of the eGenix mx Extensions. Looks like everybody is mostly happy with them as they are :-)

**EuroPython: How do you handle the license schema? I.e. the mixture of commercial and open source projects in a same application**

MAL: Yes. With the commercial bits protected by a licensing scheme.

**EuroPython: On which Python projects are you working at the moment?**

MAL: The Zope DA is pretty much done and works great. Next to come are new releases of egenix-mx-base and mxODBC 2.1 (which already happens to be part of the Zope DA). I'm also working on mxLicenseManager which provides companies like eGenix a better way to ship commercial products to customers. The Zope DA already uses it, for example.

There are also three other projects I'm currently involved with, one building on Mailman and the two others in the application server space. All use Python as primary working horse.

**EuroPython: You've written a Zope product. What is your experiences on Zope and why did you decide to write an ODBC driver product for Zope?**

MAL: Because I found that the state of Zope's SQL connectivity wasn't all that brilliant at the time (even Paul Everitt agreed here). Developing the Zope product was more work than expected, due to the missing documentation of so many important aspects in the Zope machinery. It turned out that reading the Zope source was the only way to get at all the needed details.

Now that we've completed that work, our customers don't have to bother with this any more and getting Zope connected to the world of ODBC compatible databases has become really easy.

**EuroPython: Have you already looked at the Zope3 development concerning SQL connectivity?**

MAL: Last I looked everything looked pretty much the same as in the Zope2 tree. eGenix will watch this development, of course.

**EuroPython: Can you elaborate more on the Mailman project?**

Only this much: it deals with mailing lists that have a huge number of subscribers and is built around an SQL database.

**EuroPython: About which application servers are we talking here?**

MAL: The eGenix Application Server (our in-house product, which is also part of the Siena Web Services Architecture) and custom Python-based application server platforms in general.

**EuroPython: How does the eGenix Application Server fit next to Zope and Twisted? Are the three systems comparable?**

MAL: Zope is comparable to the eGenix Application Server (eAS). In fact, both were developed around the same time.

eAS is programmed using a component model of loosely coupled objects, much like Zope3 will be in the years to come. However, it doesn't restrict the programmer to a specific working model, so it is basically up to the implementation whether it wants to use an object based approach or rather go with the traditional separation of data and logic.

The eGenix Application Server is also written with application writing in mind rather than mixing content management with web server operations. Adding new features is easy and that's the main benefit I'd expect from an application server.

Can't say much about Twisted, but I'll definitely have a go at it if the right project comes along.

**EuroPython: Why did the need arise to create another application server?**

Because there weren't any usable Python based application servers at the time (1995).

**EuroPython: Sorry, bad investigation on my part, the company Siena Technology was founded in 2002, so I thought it was developed around that time as well. What's the reason that Zope is more in the picture at the moment than other Python web application servers? Or is this just a wrong perception of the market?**

MAL: Zope is very popular and has a large following. However, it's not the answer to all problems. In cases where you are more focussed on applications than

presenting content or you have machine-machine interaction, different solutions are worth considering and provide more robustness, e.g. you don't want to a web service server to fall over just because someone added a service which causes all resources of the machine to be locked.

**EuroPython: Do you think both products can learn from each other? Would a combination of both give birth to a new application server closer to a perfect ideal?**

MAL: I thought about that, yes. It would be possible to run Zope as a service within the eGenix Application Server, for example.

**EuroPython: Why did you start in developing the mx-base library? When did the need arise? And did you at that moment expected it would become so vast and popular?**

MAL: Not really :-). I started out in November 1997 with mxDateTime and mxODBC. In 1998, I added mxTools, mxStack, mxTextTools and mxProxy.

All of these were originally written for what now is the eGenix Application Server. I made them open source to be sure they run on as many platforms as possible and because I think that basic building blocks of solutions are always good candidates for open source projects.

**EuroPython: A lot of people use the egenix-mx-base products, wouldn't it be good to include them in the standard Python bundle?**

MAL: I often get this question and from a licensing point of view there wouldn't be any obstacle in doing so. The reason why it is not done is that eGenix wouldn't maintain the product inside the Python development space. Our method of doing development is different from the Python one and we also care about things that PythonLabs is not all that interested in, e.g. Python 1.5.2 compatibility.

Not much of a deal, though, since Linux distributions are already starting to ship egenix-mx-base as part of the OS.

**EuroPython: In which way is your development method different from the PythonLabs?**

MAL: Very simple: I'm not Dutch :-)

**EuroPython: Interesting, do you want to imply that Dutch people have another way of development than the world? Or do you use an exotic way of development? :-)**

MAL: Ask Tim Peters, he knows the answer.

Seriously, software development is a creative process and sometimes the mindsets of different groups of people don't match. We'd rather listen to our users than having to discuss e.g. backwards compatibility or coding style with PythonLabs.

**EuroPython: What do you like best about Python?**

MAL: The elegance and swiftness of the language.

**EuroPython: What do you not like about Python?**

MAL: The deprecation process (because it's implemented too fast).

**EuroPython: What is the next 'big' feature Python should have, according to you?**

MAL: Nothing. The last few versions have introduced enough features already.

In my opinion, the next releases should put even more effort into optimizing the interpreter (Python 2.3 is already a lot faster than 2.2 and 2.1). I think there's a lot which can be done by giving the Python byte code compiler more hints as to which optimizations are possible, e.g. about constantness of certain object values, etc.

**EuroPython: Finally, what do you expect from EuroPython?**

MAL: Lot's of fun and good contacts. I really enjoyed last year and am looking forward to this year's event in sunny Charleroi. Hope to see you all :-)

<http://www.europython.org>

People who would like to find out more about the mentioned topics, have a look at:

<http://www.egenix.com/>

<http://www.europython.org/>

<http://www.gnu.org/software/mailman/mailman.html/>

<http://www.python.org/>

<http://www.python.org/psf/>

<http://www.siena-tech.com/>

<http://www.thekompany.com/home/>

<http://www.zope.org/>

## Interview of Phil Thompson

Martijn Faassen

**EuroPython: Phil, please introduce yourself to our readers?**

Phil Thompson: I run a software development business, Riverbank Computing, that provides programming services to anybody who has a need. Over the past couple of years this has involved a lot of embedded Linux work - kernel porting, device driver development - plus PyQt of course.

I've been working with UNIX in some form or other for over 20 years. In that time I've been a university lecturer, managed development/support teams of 60+ people (and been out of work from time to time).

At the moment I'm focusing on redecorating the bedroom which I agreed to do in a moment of weakness (ie. after my wife bought me a couple of beers).

**EuroPython: How did you get involved with Python?**

PT: I first looked at Python around 1990 to solve a particular problem, but I couldn't believe that a language was trying to tell me how I should layout my code.

I looked at it again about 5 years ago when I wanted an alternative to Tcl. I was using Tcl, Tk and incrTcl/Tk but was getting increasingly frustrated with how ugly my applications were to the user. The final straw was a release of incrWidgets that included a date widget but with the US format hardcoded.

However, I can't explain why I then moved to a language that also had Tk as the default GUI - but it all worked out in the end.

**EuroPython: How did you get involved with Qt?**

PT: I installed KDE v1. As I implied before, if I'm looking at a screen for hours at a time, it's important that I quite like what I'm looking at. Obviously it's all a matter of personal taste, but I like my desktop and it works for me. It followed that I wanted my own applications to fit in.

**EuroPython: What do you like about Python?**

PT: I think the first thing that attracted me (once I'd let the language win and decide how I should layout my code) was the extent of the standard modules. Reading the documentation I felt like a kid in a toy shop.

These days I'm more interested in the C API. The API, and the way the internals are structured, are so clean and consistent - a joy to work with. With the Python source code you know that a particular function should exist, you can guess what it is called, you can guess the name of the source file, and you can guess the sub-directory that should contain that file.

**EuroPython: In what areas could Python be improved?**

PT: My biggest concern has been resolved with the introduction of new style classes. The next major release of SIP (the wrapper generator used to create PyQt) will implement new style classes - it always feels like a step forward when you are able to rip out lines of code rather than add them.

I've always felt that Python was much weaker than Tcl when it comes to version control - particularly of external modules. Tcl allows multiple versions of a module to be

installed, and an application can specify which version it needs to use. About 3 years ago I promised to write a PEP on the subject, and one of these days I may actually find the time to do it.

**EuroPython: What distinguishes Qt from other GUI toolkits?**

PT: (Puts on asbestos suit.) I haven't really looked at other toolkits for a couple of years - so (I expect) my views are out of date.

From a user point of view, as I have said, I think visual appearance is important. If I was trying to get money out of a potential customer I would never, ever, ever show them a Tk (on UNIX) GUI based application. Gtk based applications are not pretty, but acceptable.

From a developer point of view, taken at it's simplest, Qt is appealing because you write fewer lines of code to get the same job done. I've been involved with two projects that have allowed a direct comparison between a Gtk implementation and a Qt implementation. VeePee required about 20% less code, QScintilla (the port of Scintilla to Qt) requires about 40% less code.

PyQt exploits all the advantages of Qt - plus you get the additional productivity of using an interpreted language. I have two favorite examples...

Hans-Peter Jansen wrote a demo (included with the PyQt examples) that uses Qt's data aware widgets to query and update any SQL database in 180 lines of Python.

Detlev Offenbach has developed eric (the IDE written entirely in PyQt) in a very short period of time, but it is fully featured including a debugger, syntax aware editor, links to Qt Designer, UnitTest, code coverage and metrics and two source code control systems.

**EuroPython: How did you get started writing PyQt?**

PT: It goes back to my decision to switch to Python from Tcl and wanting applications that wouldn't look out of place on a KDE desktop.

At the time (in 1998) there had been an attempt to create Qt wrappers using SWIG. The problem was (at the time - SWIG may have changed since) the wrappers were incomplete and required some "unnatural" SWIG-imposed conventions. I quickly realised that an alternative to SWIG was needed that tightly integrated C++ class libraries with Python without imposing any conventions on the programmer. So SIP was born as a small SWIG, although it has grown up a bit since then.

**EuroPython: I heard that the Qt developers were actually skeptical that this level of integration could be done, could you tell me more about that?**

PT: I've seen the comment on their web site, and they also said the same about PerlQt. I think the reaction says more about their lack of knowledge of Python-like languages and what you can do with them. Personally I have no idea why people (particularly commercial organisations where time and code re-use is money) write applications in C++. Lots of organisations have significant investment in older libraries, but they should be turning those into Python modules using SIP/SWIG etc.

**EuroPython: What was their reaction when you did it?**



PT: Confusion I think. Trolltech were kind enough to invite me to Oslo a couple of years ago. They have lots of very bright C++ developers. I'm not sure they understood why anybody would want to program using any other language.

**EuroPython: What's coming up for Qt and PyQt?**

PT: I can't speak for Qt - v3.2 is currently in beta and my guess would be that v4 would be out sometime later this year.

PyQt will continue to closely track Qt and Python. When a new version of Qt is released then PyQt is usually updated within a few days.

The major development that is being done is to SIP. SIP v4 will generate wrappers that use new style classes. It will also generate a version of PyQt that will work on native MacOS/X. I am also taking the opportunity to drop support for older versions of Python and Qt - although SIP v3 will still be maintained for those that need it.

**EuroPython: Your interviewer has been drooling over specifications of the Sharp Zaurus handheld recently, mainly because it runs Linux and Python. You ported Python to the Zaurus, could you tell us more about it?**

PT: As the Zaurus is just a Linux box, porting Python is just a case of setting up an appropriate development environment.

The extra work I did was minimising the memory footprint for such devices by looking at which of the standard modules could be removed (given that a PDA is more likely to be used as a client rather than a server) and how to split the remainder into a number of collections. Users could then choose to install only the subset they needed.

To do this I did an analysis of all the inter-module dependencies. It's surprising how many modules are imported by the time the first line of your Python code gets executed. The standard site.py imports distutils and pydoc which themselves drag in literally dozens more. I simply changed site.py to gracefully cope if distutils and pydoc aren't installed.

Porting SIP and PyQt to the Zaurus was only a little bit more difficult. With the embedded version of Qt the hardware manufacturer can choose, at a fairly low level, which classes to include. I really just had to make the configuration process a bit more flexible to deal with non-standard sets of classes.

**EuroPython: Finally, what do you expect from EuroPython?**

PT: I think the same as most other people - the opportunity to meet people who you have been reading for several years.

**EuroPython: Don't forget the Belgian beer; the other people I interviewed mentioned that. :)**

PT: Of course. When you are a guest in other people's countries it is only polite to immerse yourself fully in their culture and traditions.

<http://www.europython.org>

People who would like to find out more about the mentioned topics, have a look at:

<http://www.riverbankcomputing.co.uk/>

<http://www.riverbankcomputing.co.uk/pyqt/index.php>

<http://www.riverbankcomputing.co.uk/sip/index.php>

<http://www.die-offenbachs.de/detlev/eric3.html>

<http://www.opendocs.org/pyqt/>

<http://www.kde.org/>

<http://www.swig.org/>

<http://www.trolltech.com/>

<http://www.trolltech.com/products/qt/>

<http://www.riverbankcomputing.co.uk/zaurus/index.php>

<http://www.zaurus.com/>



## Interview of Moshe Zadka

Tom Deprez

**EuroPython:** I've you search the net for Moshe Zadka, you easily end up in: <http://www.moshez.org/>. So, it's easy to learn a lot about 'Moshe', but anyway, can you introduce yourself to the readers in a condensed way?

Moshe Zadka: I'm a mathematician (currently working on my MSc. in differential topology) from Israel. I have had the UNIX bug since 1995, and the free software bug since 1997.

**EuroPython:** How is your name pronounced? Do you experience that people have some difficulty with your name?

MZ: I'll start with the second part: "Moshe" is a fairly popular Jewish name, so Jews usually know how to pronounce it. English speakers tend to pronounce "she" like the 2nd person female pronoun, instead of as the beginning of "shell". As long as that mistake is avoided, I am pretty flexible with how my name is pronounced.

**EuroPython:** Have you heard about EuroPython before?

MZ: Yes. I have thought about going to the first EP, but life got in the way. I am very happy to be able to go to this one!

**EuroPython:** What's your relationship with Python?

MZ: I'm a (fairly non-active, currently) core Python developer. I did all kinds of small things in the Python CVS, including documenting a few of the more obscure modules and implementing `_contains_`.

**EuroPython:** Did you manage to combine Python with your work, or do you have to use other programming languages?

MZ: Currently I don't do any programming professionally. However, I've managed to use Python in several of my previous jobs -- as infrastructure for exploratory and algorithmic statistical research, build infrastructure and dynamic web sites programming.

**EuroPython:** You're the person with the highest talk proposals. Ten, of which seven are accepted. For being 'fairly non-active', you've do admit that it is a bit contradictory. Where is the catch? Do you like to talk or is there another driving force? Python Advocay?

MZ: Well, I'm not as "non-active" as it seems. I have stopped being active on the main Python mailing lists, but that's mostly because my energies have been devoted to Twisted -- the new up and comer. Also, yes, I do enjoy public speaking, as curious as it may be in a geek.

**EuroPython:** On your website, I've read about your own project 'Python Mail System' and your involvement in the 'Bicycle Repair Man' project. Can you tell us more about these projects?

MZ: PMS is the mailer I use daily, but I do not maintain it anymore. The biggest lesson I have learned from PMS is that in Python, it is often easier to just rewrite an application from the ground up then to use someone else's -- nowadays, I'm recommending to people who want to write a Python MUA to use the "email" module and write it themselves.

I'm no longer involved in BRM. I helped Shae Erisson a bit, back when he was the lead developer. It is a very exciting project -- the only refactoring browser for Python which is free. A refactoring browser is a tool to help do menial tasks when refactoring code such as abstracting repeated code into functions, moving functions between modules, etc.

**EuroPython:** So you actually encourage people not to use PMS? And write one of their own?

MZ: Yes. I realize it sounds weird, but I realized at some point PMS is an extension of what I wanted in e-mail software. Everyone has different needs and priorities. Many of the things in PMS started out as being generic, but I only used the genericity in one point. It really is so easy to write software with Python, often frameworks aren't needed.

**EuroPython:** Why the name 'Bicycle Repair Man'?

MZ: I wasn't involved in choosing it. It is my understanding it's based on a Monty Python skit.

**EuroPython:** Is the BRM project still active? Would you recommend people to have a look at it and join it?

MZ: I have no idea. Like I said, I am no longer involved in it.

**EuroPython:** On which Python projects are you working at the moment?

MZ: I'm most active on "Twisted", the asynchronous networking framework. I'm involved in design and documentation, but what interests me most from a research point of view is how to deploy Twisted-based applications in a flexible manner. That ties in with my biggest role in the Twisted project as the Debian maintainer. I make sure Twisted integrates with the Debian tools, so that I can give users of the Debian operating system a system which is powerful, easy and flexible.

**EuroPython:** We've heard about Twisted before :-)) (see also interview with Itamar last year). To what can it be compared?

MZ: Well, it's hard to compare it honestly to anything else. The first thing that comes to mind is "asyncore" and "medusa", but Twisted is based on a much stronger pattern (complete decoupling between protocol and transport), and has much higher-level parts (like a web server, NNTP server and IRC clients), all of which can be easily integrated. It can be compared to Zope, but it does not depart that much from standard Pythonic sense (for example, it is easy to avoid inheriting from multiple classes in Twisted) and does not force a complete methodology.

But really, perhaps the best comparison is to the Python standard library, but for networking: you don't have to use it, but if you don't, you'll waste your time and reinvent the wheel.

**EuroPython:** I'm a little bit confused. You compare Twisted with a library and an application. Is it both? Ie does it contain a general layer (the library) and a layer on top which makes Twisted as an application? Are all layers independent from each other?

MZ: In Python, there is no clear difference between applications and libraries -- which is why I prefer the word

"framework". Here is an example: if you are writing a Gtk+ program, it is possible, with Twisted, to embed a production-grade web server in it with one function call. Another way of saying that same thing is that it is easy to write a GUI program which will manage, in-process, a Twisted web server. It is hard to say which of these depictions is closer to the truth, if such a truth exists.

I gave the example of a web server intentionally: that same line to add a web server would work on the command line, as a way of writing a web server force a complete methodology.

But really, perhaps the best comparison is to the Python standard library, but for networking: you don't have to use it, but if you don't, you'll waste your time and reinvent the wheel.(this is how most GUI programs inside Twisted work).

Note that the Python standard library contains such things as Tkinter's mainloop or SocketServer's serve\_forever, both intended for taking over an application's main loop.

**EuroPython: What is Twisted for yourself? Why would you recommend people to Twisted and how would you make a person 'warm' for Twisted?**

MZ: As I said above, I really think that every Python programmer writing networking code should know about Twisted.

**EuroPython: You've mentioned Zope. EuroPython has a complete track, three days long devoted on Zope. Can this be devoted on different ways of marketing? Or is Twisted younger and needs a bigger group of followers?**

MZ: Well, Twisted is younger. Zope has been around, if you count in Principia and Bobo, for about a decade, while Twisted has only been around for three years. I'm pretty sure, though, you could fill in three days of Twisted track if you wanted -- so you'll have to ask the EP organizers to give Twisted a track next year.

**EuroPython: Have you've used Zope and Twisted in such a way that you could give us a sort of comparison (your view)? What are the Pro's/Con's of both?**

MZ: I have to first say that I hadn't had occasion to use Zope 3, so all my remarks will pertain to Zope 2. Zope 2 was a pleasure to work with, including the TTW management. I even wrote a Hero(TM) GM's helper with Zope, which automatically rolled dice and calculated statistics, and integrated with the prose for an adventure in a transparent way. However, Zope is hard to extend with pure Python. As soon as I tried to do non-trivial programming, I had to use Products and learn about arcane things like "the Folder interface". I feel that Zope is very well suited to a mainly-content site, especially when the TTW editing capabilities are important to the clients. Twisted is good in programming-heavy contexts, especially when the "site" has to integrate with other protocols.

**EuroPython: If I'm right, it is one of the intentions of Zope3 to make it easier to extend it with pure Python.**

MZ: So I have heard too.?

**EuroPython: Are you following the development of Zope3?**

MZ: No.

**EuroPython: Is someone else of Twisted following Zope3? Or the other way around is someone of the Zope developers also involved in Twisted?**

MZ: I am not sure. I think itamar follows Zope, but perhaps not too closely. As for the Zope developers, you'll have to ask them.

**EuroPython: Do you think Zope and Twisted could benefit from each other (cross-breed) or are they competitors? Is there any direction in cooperation between the developers of both products?**

MZ: There has been some work on basing Zope 3's network layer on Twisted, and there was an ugly hack for making this happen with Zope 2. It is my belief that Zope, and ZEO, could benefit from using Twisted rather than medusa. However, currently nobody is working on making the official Zope 3 network layer be Twisted, as far as I know.

**EuroPython: Any idea why there is less 'official' interest? Is it a simple priority rule? It seems to me, what I have heard during this interview, both could indeed benefit a lot from each other.?**

MZ: I am not sure. Again, this is something where I expect Zope developers interest to lead the way, and I am not involved in Zope.

**EuroPython: Can you direct us to some applications which are able to show the full power of Twisted?**

MZ: No :-)

The simple reason is that Twisted is very powerful, so it's hard to show the full power of Twisted. However, there are quite a few nice Twisted-based programs: HEP is an integrated message server, mc-foo is a client/server media player, conch is a Twisted ssh client/server and Twisted Web, which I will be talking about in the conference.

**EuroPython: What do you like best about Python?**

MZ: I don't know. I want to clarify that -- Python has a certain quality that makes it "easy to use". I don't know what it is -- if it is the namespace model, the indentation-for-structure or something else. But it is this quality that makes me choose Python for each task, or to be sorry I didn't whenever I don't.

**EuroPython: What do you not like about Python?**

MZ: I am somewhat apprehensive about the "new style classes". It might be just an old-timer's reluctance to change, but I feel they lack the simplicity that Python has.

**EuroPython: Can you elaborate more on the 'new style classes'?**

MZ: Not really. I haven't used them enough in a production environment (Twisted only recently lifted the compatibility with Python 2.1 requirement) so I don't have any real data yet.

**EuroPython: Is there something that Python doesn't have yet, of which Twisted would be able to benefit if someone implemented it?**

MZ: I'm going to be unoriginal here: a cross-platform high-quality version of psyco integrated in the standard library.

**EuroPython: Why is it that Python doesn't play along with the bigger development languages, like Java, C++, Delphi, ... ? Is Python too immature? Or is it something else?**

MZ: I beg to differ.

Python certainly "plays along" with the "bigger" development languages. Many projects are written in Python (for example, large parts of the Debian infrastructure are) and Python is used for mission critical roles everywhere. Just look at where posters to c.l.py work :-). Python may not have Java's marketing hype or C++'s "industry standard" stamp, but it is certainly alive and well.

**EuroPython: Correct. But isn't it a fact that lot's of companies from the closed-source tend to use Java, C++ or the newcommer C# more quickly than Python?**

MZ: I think that nowadays nobody is seriously considering Java as an end-user delivery language, except for Sun and Oracle. C++ has the "industry standard" stamp -- people feel it is a "safe" choice. I don't know of any major applications written in C# either.

**EuroPython: I lately hear a lot from people in my neighbourhood that in the future most development (mostly applications which need a very strong connection to the user. Very high User Interface or internet/intranet related) will be done in either Java or .Net. What do you think about such statements. Is/will Python able to deliver the same things?**

MZ: I feel such statements are usually made by people buying Sun's or MS's hype. Python is already able to deliver those things: with Twisted, writing graphical clients for network applications is a breeze, and they are easier to program since no threads and race conditions have to be taken into account.

**EuroPython: Can you direct us to some examples of such clients written in Twisted?**

MZ: tkonch is a nice one, in Twisted: it is a graphical ssh client (much like PuTTY).

**EuroPython: Browsing through your webpages, I saw something named 'Advogato'. What is it and why does it interest you?**

MZ: <http://advogato.org> is a blog devoted to free software and the free software community. Many interesting people use it as their main blog and there are occasionally interesting articles related to free software there.

**EuroPython: You are a Debian developer. Is there a special reason why you choose Debian?**

MZ: Debian is a portable, comprehensive and free operating system. This means I can trust Debian to work on all the machines I have, to have the software I need and to not limit me in how I use it. It is an experiment in integrating a lot of software, and as far as I am concerned -- a successful one.

**EuroPython: Are Debian and Python the only things you are interested for in the computer world?**

MZ: I am not much of a theory guy, when it comes to computer. Thus, I enjoy implementing ideas I have. Python is (almost) always the right technology to use, and Debian is (almost) always the correct way to deploy it. I'm not sure if it qualifies as a "yes" or a "no", though.

**EuroPython: If the readers didn't knew yet, together with Anna Ravenscroft, you volunteered to host the Lightning Talks and Open Space. To you like these kind of talks/discussions?**

MZ: I always have a simple recommendation of lightning talks: even if they are dead-boring, the boredom is over in five minutes. Most often, the speaker has to finish before

you would get bored.

**EuroPython: To whom need the interested people address to give an LT at EuroPython2003?**

MZ: They should contact either me, at [epit@moshez.org](mailto:epit@moshez.org), or Anna, at [rev\\_anna\\_r@yahoo.com](mailto:rev_anna_r@yahoo.com). It's recommended to CC: us both.

**EuroPython: If EuroPython would give an explicit track to Twisted next year. Could we then count on you for a full day of Moshe's Twisted talks? ;-)**

MZ: Probably :-)

**EuroPython: Finally, what do you expect from EuroPython? Anything you look forward to?**

MZ: I am looking forward to meeting people whom I've met before (like Alex Martelli and Jack Jansen) and to meeting people with whom I have had quite a bit of online dealings but have never met (like Michael Hudson and Anna Ravenscroft). For me, conferences are mostly about meeting people, not about giving or listening to talks, even if the amount of my talk proposals would seem to indicate otherwise.

**Interested readers can find more info at:**

<http://www.advogato.org/>

<http://www.debian.org/>

<http://www.europython.org/>

<http://www.moshez.org/>

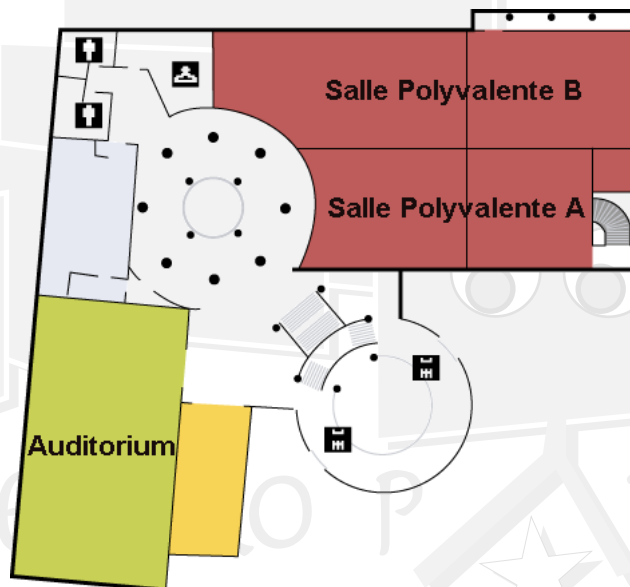
<http://www.twistedmatrix.com/>

## Floorplan

We've rented the CEME congress building at Charleroi. On this page you'll find the floorplans of the ground and first floor.

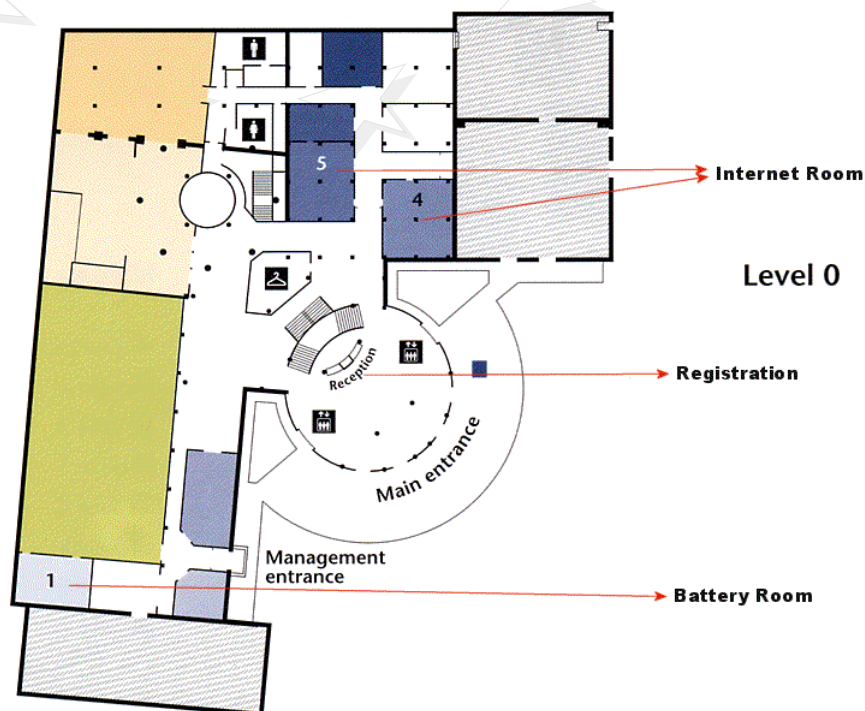
### Conference Rooms

The conference rooms are located on the first floor of the building:



### Internet access and battery charging

These rooms are located on the ground floor. We will give you all the technical details at the conference itself. Just take your laptop and an Ethernet cable (RJ45) with you.



---

## General Information

### City Information Office

It's in front of the main railway station of Charleroi. They speak English so don't be afraid. They will be pleased to help you, it's their job.

**Telephone :** 32(0)71 31.82.18

**Open Time** 9h until 17h

(If they don't answer, here is another phone number where you could get some info about the town:

**Telephone :** 32(0)71 86.61.52

### Phone Numbers

If you use your own international phone you have to compose **32 71** in the beginning. If you use a public phone in Charleroi you have to start with **071**.

### Important Phone Numbers

**European Emergency Help:** 112

**Medical Emergency:** 100

**FireDepartment:** 100

EURO PYTHON  
Conference 2003

## Travel Information



With one airport closeby and another just reachable within an hour, a Railway station with a connection to the High Speed Train and several motorways, Charleroi is easily reachable.

Once in Charleroi, public transport can bring you to almost everywhere.

For more information on the location of the conference, please have a look at the location webpage.

### **Getting to Charleroi:**

#### **By road**

E19 from Brussels, Antwerp and Amsterdam, in the North

E 42 from Lille, Paris and the Euro Tunnel, in the West

E 42 from Namur, Lie`ge and Cologne, in the East

#### **By train**

Arrival at Charleroi Station, right at the heart of the city centre. Train links from all big European cities to Charleroi. For more information, visit SNCB

Charleroi Station has also a connection with the TGV (high-speed train) network

#### **By air**

Arrival at Brussels Airport in Zaventem. Charleroi is easily accessible by road, after an hour's drive on the E 19 or by train, from the airport via Brussels.

The train trip will take aprox. 1 hour 25 minutes, the ticket (Zaventem Airport - Charleroi Sud) costs around 8,30 EUR. The times below may vary by 5 minutes, so be sure to be 10 minutes earlier at the train station.

#### **Time Schedules from Zaventem Airport to Charleroi Sud**

Zaventem Airport	Every 05' or 30' past the hour
Arrival Brussels North (Bruxelles Nord)	Every 20' or 52' past the hour
Departure to Charleroi	Every 32' or 01' past the hour
Arrival at Charleroi	Every 26' or 53' past the hour

Regional Airport "Brussels South Charleroi", Building S11B, 6041 Charleroi (Gosselies) phone 32(0)71 25.12.11

If flying from other parts of Europe on a budget, strongly consider taking one of the budget airlines ([www.ryanair.com](http://www.ryanair.com), [www.easyjet.com](http://www.easyjet.com), [www.go-fly.com](http://www.go-fly.com)) to Stansted and out to Charleroi. This would cover about 30 locations in Europe...

From the USA, consider staging through London...

## Location of the Conference Building

The conference will take place in Belgium, Charleroi to be more precise. The building is called "CEME" (Charleroi Espace Meeting Européen). You can find more information about it on their website. You may see a huge image of the surroundings by clicking [here](#).

You can reach the conference building by car or by bus:

### By bus (TEC):

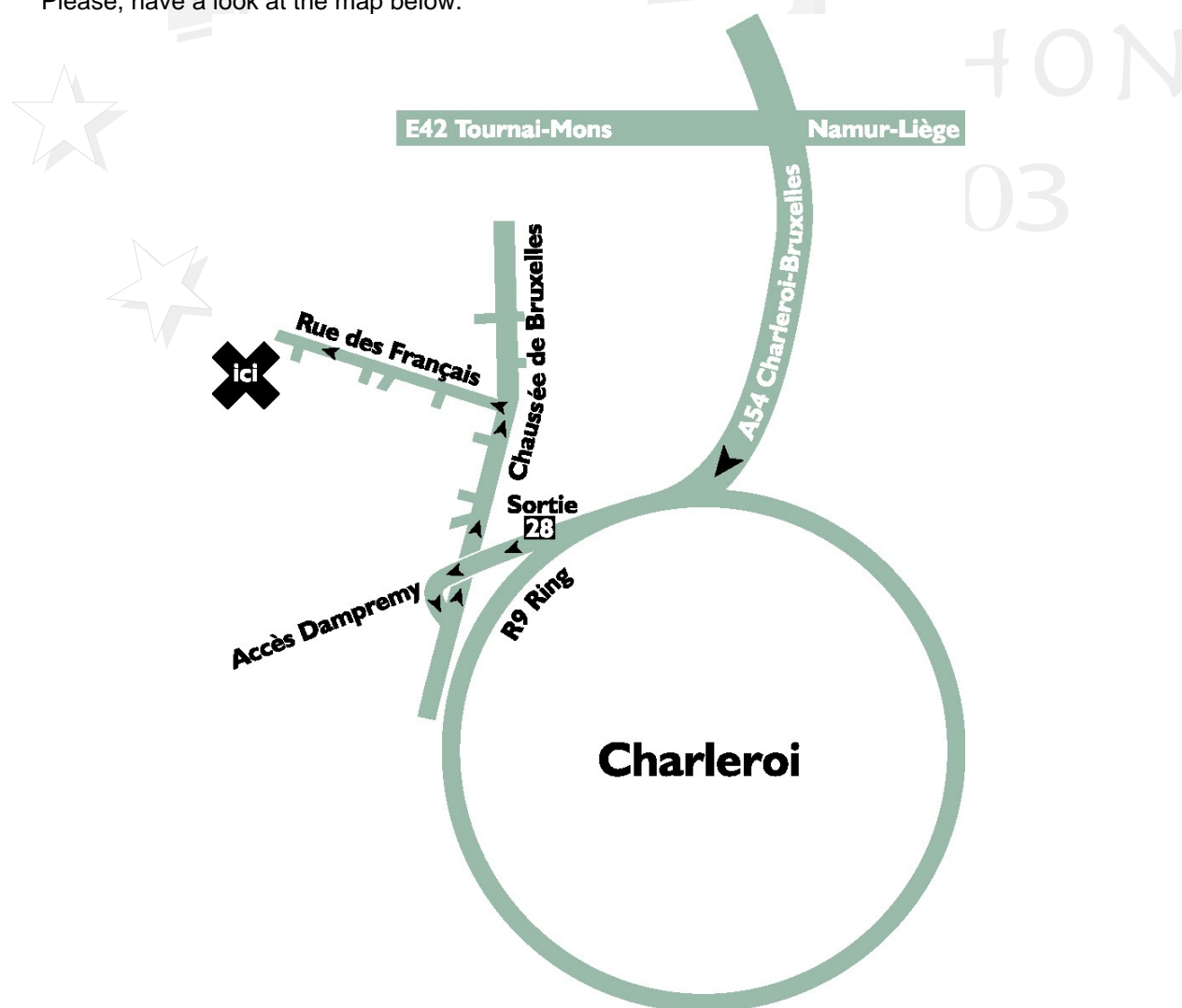
Starting at the 'South' railway station of Charleroi ('gare du Sud'), you have two possibilities (A bus ticket costs around 1 Euro):

Take any bus, indicating the direction 'Chaussée de Bruxelles'. Exit the bus at bus stop 'La Planche', traverse the road. On the other side of the road, follow the street 'rue des Français'. You will find the CEME at the end of this street (approximately 5 minutes on foot).

Take the bus with line-number 41 heading for 'Courcelles' (less frequent than the above possibility). The trip will take about 10 minutes. Exit the bus at bus stop 'Ravin' and take the street 'Gohyssart' (rue de Gohyssart) which leads to the CEME (approximately 3 minutes on foot).

### By car

Please, have a look at the map below.





## Transportation Information

### Transport in Charleroi

#### Public transport (Bus)

With public transport in Charleroi you can go almost everywhere (68 lines). Just take the time to read carefully the long timetables. If you know where your hotel is, have a look now on the website (unfortunately only in French...): <http://www.tec-charleroi.be>

They also have a phone number if you get lost among all their lines.

**Phone:** 32(0)71 23.41.15

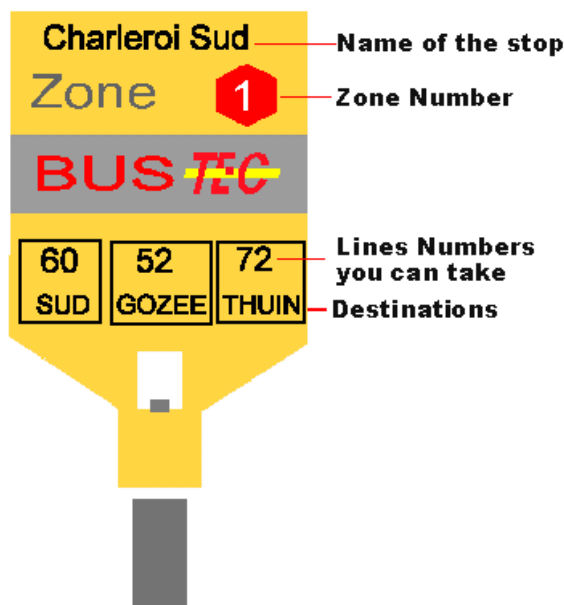
**Open:** 6h30 - 18h30

Unfortunately they don't speak english but force them a bit and you will succeed to know what you need.

**Payment:** Ready cash (most of the drivers won't accept your big notes). The price depends of how many zones you will have to travel:

1 TICKET		
Number of ZONE	Normal price	For 1 Day
1 or 2	1,10	-
3	1,65	-
4	2,20	-
5	2,75	-
6 and more	3,30	-
All zones	3,30	5,50

**Signs:** To understand the bus signs, here is what you will see most of the time:



### Interesting lines for CEME:

**LINE 41** Take the bus with linenumber 41 heading for 'Courcelles' (less frequent than the above possibility). The trip will take about 10 minutes. Exit the bus at busstop 'Ravin' and take the street 'Gohyssart' (rue de Gohyssart) which leads to the CEME (approximately 3 minutes on foot). (Here is the timetable). Be careful that this bus is going from 5h40 until 20:50 !! So if you want to go to restaurant or something you will have to find another solution.

**Second solution:** Take any bus, indicating the direction '**Chaussée de Bruxelles**'. Exit the bus at busstop '**La Planche**', traverse the road. On the other side of the road, follow the street 'rue des Français'. You will find the CEME at the end of this street (approximately 5 minutes on foot).

### Taxi

Here are the 3 main taxi companies of Charleroi. It's the same price for all of them so just make your choice (about 1 euro/km).

#### TAXIS CAROLO

Phone 1: 32(0)71 32.32.32  
Phone 2: 32(0)71 20.83.76  
Phone 3: 32(0)71 20.83.83

#### IDEALTAX

Phone 1: 32(0)71 36.52.42  
Phone 2: 32(0)71 43.05.05

#### Confort Tax

Phone 1: 32(0)71 41.31.31  
Phone 2: 32(0)71 48.82.00  
Phone 3: 32(0)71 21.99.99

### Rent a bike

Bicycle is a way to feel more free in Charleroi. Here are the places:

**La Maison des Cyclistes** Bicycle only:

Address: Rue du Grand Central 57, 6000 Charleroi

Phone: 32(0)71 53.58.50

Open Time: 12h30 - 17h

### Hitching

I am sure you will be able to find a friendly Belgian ready to give you a lift!

### Share cars

But best way is maybe to share cars among you...

## Places To Meet

### Cafe

Best place to meet and best moments you expect, I am sure: meetings around belgian beers... But where to drink them? There are about 150 cafes in Charleroi and the surroundings, so you will have 'some' choice. Here is a small list of our favourite ones:

**Brasserie de l'Eden** A big brasserie where you can drink a lot of belgian beer and also eat. Last year, this brasserie was the meeting place after each conference day. This year again they are ready to welcome all Europython participants. A free drink is offered there by them. As we go there quite often we even took a MENU (Vegetarian food available!). We told them to be ready with lot's of Chimay.

Address: Boulevard Jacques Bertrand 1  
6000 Charleroi

Phone: 32(0)71 30.64.96

Open Time: 9h - Open

**Le Luxembourg** A really good cafe/restaurant near to the Business HOTEL. Very calm and nice.

Address: Rue du Pont Neuf, 41  
6000 Charleroi

Phone: 32(0)71 31.63.59

Open Time: 10h - 02h

**Star Rock Cafe** Well known cafe in the center of Charleroi.

Address: Boulevard Joseph Tirou 122  
6000 Charleroi

Phone: 32(0)71 30.28.77

**Le Village Africain** Great african atmosphere there, really simple but nice.

Address: ...  
6000 Charleroi

Phone: 32(0)....

**The Irish Times Pub** If you miss the Irish Guinness and the Irish atmosphere (they also have Belgian beer [fortunately...]). They speak more English than french over there ;)

Address: Boulevard Joseph Tirou 86  
6000 Charleroi

Phone: 32(0)71 31.61.52

Open Time: 12h - 02h

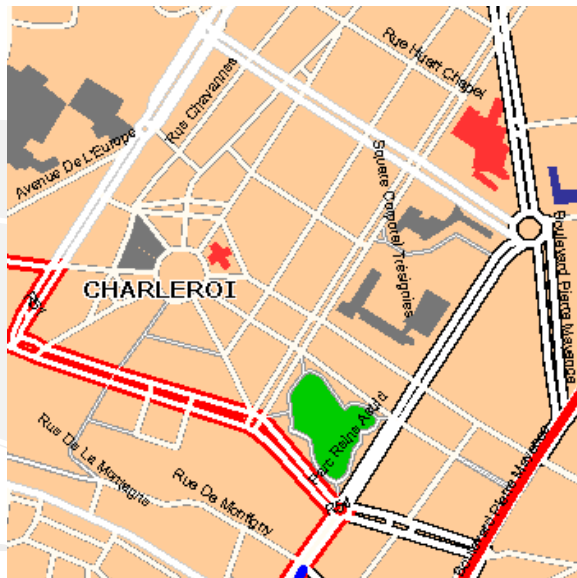
### Restaurant

There are about 70 restaurants in Charleroi. Here too you will have many choices. Of course it's impossible to list them all here so we advise you to make your choice on this website: <http://www.resto.be> Or just see when you will be there. Charleroi isn't big so you find different kind

of restaurant near to each other.

### **Green meeting place**

Charleroi isn't a so green town but there one main park: **the Reine Astrid Park**. It's near to the center of Charleroi. Here it is on the map:



### **CyberCafe**

For the most addicted among you, you will be able to access the net after Europython (don't forget that you will be able to access the net during the conference!!) We didn't tried this shop so it's at your own risk...

### **Mont Cameroun**

Sounds african internet there :)

Address: Rue de l'Aigle Noir 7  
6000 Charleroi  
Phone: 32(0)71 50.92.40  
Open Time: 09h - 22h

We sincerely hope you will enjoy your stay in Charleroi.

---

## Contact Information

If you need to contact local people at Charleroi concerning information about the conference, sponsorship, hiring a booth, etc. :

### General means:

Telephone : +32(0)71/27.03.11

Fax : +32(0)71/31.67.35

(Let them know that you are calling concerning the EuroPython congress and they will know whom to contact)

### Electronic Mail:

You can contact the organizers directly by mail. We will do our best to answer all your questions and if possible pass you to the right person for your case.

You can reach the organisers through the following email address:  
[europython@p3b.org](mailto:europython@p3b.org)

Address of the Organization Team  
EuroPython Conference Team  
P3B c/o Aragne  
Boulevard Général Michel 1E  
B-6000 Charleroi.

EURO PYTHON  
Conference 2003

## Credits

<b>Conference Chairman</b>	Tom Deprez
<b>Local Chairman</b>	Denis Frère
<b>Conference Compere</b>	Tim Couper
<b>Track Chairs</b>	<i>Without them we wouldn't have over 70 talks and tutorials! "Thou mighty prophets; --Be nice to them during the conference"</i>
<i>Lightning Talks, Open Space, Bofs</i>	Anna Ravenscroft; Moshe Zadka
<i>Python Frameworks Track</i>	Martijn Faassen
<i>Python Language Track</i>	Michael Hudson
<i>Python in Business Track</i>	Tim Couper; Marc-Andre Lemburg
<i>Python in Science Industry Track</i>	Nicolas Chauvat
<i>Zope Track</i>	Heimo Laukkanen; Paul Everitt
<b>Local Organisation Team</b>	<i>Without them, there wouldn't be a conference building, no food, no internet room, no t-shirts, ... and most important: are taking the financial risk!; --Make it easy for them at the conference</i>  Denis Frère; François Gillet; Vincent Maton; Alain Meurant; the whole Aragne, P3B and OS3B team.
<b>Website</b>	<i>Keeping everybody up-to-date. Providing interesting info,...</i> Tom Deprez; Vincent Maton; Jean-François Roche; Joachim Schmitz
<i>Registration and Database</i>	Aixtraware
<i>Hosting</i>	Amaze
<b>Graphics</b>	<i>Who made everything pleasing for the eyes</i> Vincent Maton
<b>Daily IRC meetings club</b>	<i>Who offered big help and support, gave a lot of ideas, discussed a lot of issues, provided contact points, fixed very important things; In one sentence : "they worked their butts off, be them ever grateful"</i>  Laura Creighton; Tom Deprez; Martijn Faassen; Michael Hudson; Anna Ravenscroft; Moshe Zadka
<b>Proofreading</b>	<i>Who made sure we didn't upload too many spelling mistakes</i> Paul Everitt; Anna Ravenscroft
<b>Brochure</b>	<i>Without them, we wouldn't have such a nice brochure as we are now looking at.</i> Reportlab

## Credits

**Speakers** *Without them there wouldn't be much to see at EuroPython*

Albertas Agejevas; Steve Alexander; Francesc Alted; Xavier Antoviaque; Harald Armin Massa; Ludovic Aubry; Kit Blake; Brad Bollenbach; Robert Boulanger; Benjamin Bruheim; David Chan; Pravir Chawdhry; Francesco Ciriaci; Simon Eisenmann; Paul Everitt; Martijn Faassen; Stefane Fermier; Rasmus H. Fogh; Francesco Garelli; Marius Gedminas; Dinu Gherman; Francis Glassborow; Duncan Grisby; Jürgen Hermann; Bernhard Herzog; Stefan H. Holek; Dirk Holtwick; Michael Hudson; Juan David Ibáñez Palomar; Jack Jansen; Andreas Jung; Nik Klever; Wojciech Kosinski; Heimo Laukkanen; Marc-André Lemburg; Alexander Limi; Niels Mache; Alex Martelli; Klaus G. Müller; Federico Nati; Ronald Oussoren; Christophe Perrin; John Pinner; Anna Ravenscroft; Roeland Rengelink; Thomas Reulbach; Leonard Richardson; Armin Rigo; Andy Robinson; Maik Röder; Guido van Rossum; Just van Rossum; Benjamin Saller; Alan James Salmon; Ulrich Schneider; Vincenzo Di Somma; Aroldo Souza-Leite; Andrew Smart; Jean-Paul Smets; Duncan Smith; Konstantin Teplinskiy; Phil Thompson; Darrin Tisdale; Christian Tismer; Daniel Veillard; Joachim Werner; Torsten Will; John Wilson; Moshe Zadka; Christian Zagrodnick

**Visitors** *Yes, without you, all our efforts would have been wasted and speakers would have a very silent audience... Thank you for supporting EuroPython 2003!*

And finally, a big "thank you" to all of you who attended the conference!

### ***How this brochure was made***

This brochure was made using ReportLab software - both the free open source toolkit and the commercial products.

The initial information came from a variety of filtered sources - including a Zope database. The cover, template for the inside pages and all the adverts were submitted as PDF files. The ReportLab product PageCatcher was used to 'catch' these pages, allowing us to manipulate and reuse them in the final brochure. All this was integrated into this brochure using the open source ReportLab toolkit (and the Python programming language).

These techniques meant that the brochure could be regenerated when needed, and kept constantly up-to-date despite the inevitable wording and timetable changes.

For more information about ReportLab, the reportlab open source PDF toolkit and PageCatcher (and our other commercial products), go to [www.reportlab.com](http://www.reportlab.com).