The $\mathcal{N_TS}$ project: from conception to birth*

Philip Taylor
RHBNC, University of London, United Kingdom
p.taylor@exch1.rhbnc.ac.uk

Jiří Zlatuška
Masaryk University, Brno, Czech Republic
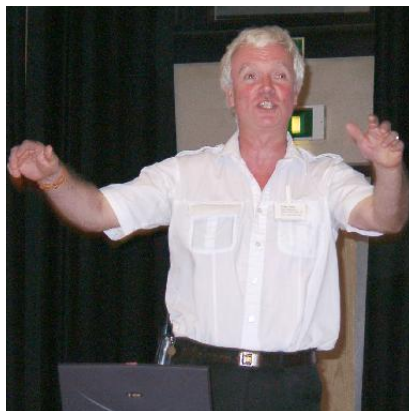zlatuska@muni.cz

Introduction

It is an enormous pleasure and privilege to be able to present this paper on $\mathcal{N_TS}$ at a TUG conference in Oxford. For almost ten years, $\mathcal{N_TS}$ has slowly been evolving from a concept to a reality, and I am delighted to be able to report that $\mathcal{N_TS}$ is virtually complete. The fact that we have reached this point is due almost entirely to the efforts of one man: my co-presenter, Karel Skoupý. Karel has work tirelessly on this project, and without his efforts I have no hesitation in saying that we would not be presenting $\mathcal{N_TS}$ as a success story today.

Let me start by presenting an overview of today's talk and presentation; We will attempt to cover seven separate areas, including (of course) the mandatory questions and answers at the end. The seven areas to be covered are:

- A brief history of $\mathcal{N_TS}$
- TEX, ε-TEX & $\mathcal{N_TS}$ compared
- The choice of Java as the language of implementation
- An overview of the classes, object and methods of $\mathcal{N_TS}$
- A summary of the *status quo*
- A demonstration of $\mathcal{N_TS}$, and comparison with TEX
- Questions & answers

and you will soon realize that my expertise lies very much in the earlier areas; the *implementation details* of $\mathcal{N_TS}$ are very much Karel's area, and I will defer to him whenever any explanation of a detailed implementation issue is called for.

Phil Taylor