

Preface

These proceedings record the First International Workshop on Persistence and Java, which was held in Drymen, Scotland in September 1996. The focus of this workshop was the relationship between the Java languages and long-term data storage, such as databases and orthogonal persistence. There are many approaches being taken, some pragmatic and some guided by design principles. If future application programmers building large and long-lived systems are to be well supported, it is essential that the lessons of existing research into language and database combinations are utilized, and that the research community develops further results needed for Java.

Attendance at the workshop was limited to those actively working on persistence mechanisms for Java, and required the submission of a position paper. From approximately twenty submissions, seventeen were chosen. The papers cover the spectrum of persistence solutions that are currently being pursued in industry and academia. As we were initiating a new workshop series and had to react quickly, the arrangements were fairly informal. Malcolm Atkinson and Mick Jordan took responsibility for soliciting and selecting the material, inviting participants and organizing the event.

The initial idea for the workshop came from Malcolm Atkinson, who leads the Persistence and Distribution Research group at Glasgow University. The idea was one of the first fruits of the collaborative research program that was initiated between Sun Microsystems Laboratories (SunLabs) and the Glasgow group in the fall of 1995. SunLabs sponsored the workshop to cover the attendees' local costs.

Malcolm had previously been jointly responsible for starting the series of workshops on Persistent Object Systems, which meets bi-annually, with the last (POS7) meeting being in May of 1996. Given that forum, one might ask why we deemed a special Java-specific workshop to be necessary. The answer lies in the phenomenal growth of interest and acceptance of the Java language and platform, which is unprecedented in the history of computing. By the fall of 1995, this phenomenon was well established and we were beginning to see proposals for Java persistence mechanisms. It seemed important for Java persistence developers to meet at the earliest opportunity in order to compare strategies and develop consensus.

Our goal for the workshop was to gather together as many groups actively working on persistence as possible, a forum that would provide the maximum opportunity for discussion and effective interaction. For the most part, we were successful in this endeavor. The workshop location provided an excellent environment for interaction, and the participants clearly benefited from the intimate atmosphere, which contributed to some lively discussions. Our one concern, as with POS7, was the limited amount of industrial participation. However, given the frantic rush to develop products for Java, this is perhaps understandable.

At the conclusion of the workshop, we polled the attendees on whether we should hold a second workshop, and if so, when. The unanimous response was "Yes, next year." PJW2 will be held in Half Moon Bay in the San Francisco Bay Area, California, August 13-15, 1997. We hope to further the original PJW1 goals, and papers are sought on topics such as:

- implementation techniques for persistent Java systems
- analyses of performance or usability of persistent Java systems

- analyses of projects using particular forms of persistence for Java
- the influence persistence should have on future Java systems
- new models of persistence for Java
- comparison of models of persistence for Java
- tools supporting application programming using persistent versions of Java.

We plan to have a focused session on the ways in which the Java virtual machine might be adapted to better support persistent implementations.

Further details may be found at <http://www.sunlabs.com/research/forest/pjw2>.

The papers are arranged in the proceedings in the order that they were presented at the workshop.

—Malcolm Atkinson & Mick Jordan