# 10 Overview

The following sections give a description of an interactive software environment for UNIX systems, called *DABTool*, in which the algorithmic techniques discussed in the first part of this report have actually been implemented. The reader should refer to Part I for all terms and concepts not defined here. We also adopt the same notation as in Part I, see Section 4. In particular, when discussing a generic instance of the ensemble planning problem and a corresponding solution, $G = (V, E)$ denotes the area graph, $R$ the requirements, $M$ the maximum ensemble size, and $\mathcal{B}$ a corresponding ensemble assignment.

The current version of the software has been developed and tested under SuSE Linux 6.x with kernel 2.2, libc6, X11R6, GNU C++, LEDA 3.8 and Tcl/Tk 8.0. However, the system should be portable to any POSIX-compliant UNIX system with recent versions of the X11 windowing system, GNU C++ compiler and the LEDA library. The test configuration and database interface modules require Tcl/Tk (including the Tix extension) and a recent ADABAS version, but these are optional add-ons which are not required for the remaining parts of the software to work properly. In fact, previous versions of the software have been ported to SUNOS 4.x and 5.x systems with minor effort.

We emphasize that the system described here has mainly been designed for research purposes and as such is not (yet) a program for the end user. The main purpose of the program is to carry out systematic testing of different solution approaches some of which are still in development. However, it already provides the basic facilities needed to actually perform calculations on real-world DAB network planning problems.

The DABTool program is in fact a heavily modified version of the LEDA graph editor tailored for the DAB ensemble planning problem (LEDA = *Library of Efficient Data Structures and Algorithms* is a C++ library featuring high-level graph classes and operations, which is being developed at the Max Planck Institute of Saarbrücken, Germany; see http://www.mpi-sb.mpg.de/LEDA). It supports automatic generation of graphs with up to 200 nodes and arbitrary edge densities. (The number 200 is more a practical limit for interactive usage, since even for, say, only 100 nodes the display becomes a bit crowded. Internally, and when the program is run in macro/batch mode, the system can handle much larger graphs if enough main memory is available.) Both general and specific (bipartite, grid, planar, disk) types of graphs can be generated. A collection of useful operations for testing graph properties, manipulation of the graph and automatic embedding of the graph is provided as well.

The program also supports automatic generation of up to 100 services with random integer sizes in the range $[0, 100]$ and generation of different types of random requirements. (Again, these limits are only a pragmatic restriction of input parameters; internally, the system can handle service lists of arbitrary length and with arbitrary sizes.)

Currently, the following algorithmic tools are provided:

- Computation of four kinds of approximate LLB's ("Largest Lower Bounds"): The *sum* and *FF LLB's* provide a lower and upper bound for the clique packing number, respectively, in which the size $p_M(W)$ of a clique $W$ is approximated using $\lceil \mu_{R_W}/M \rceil$