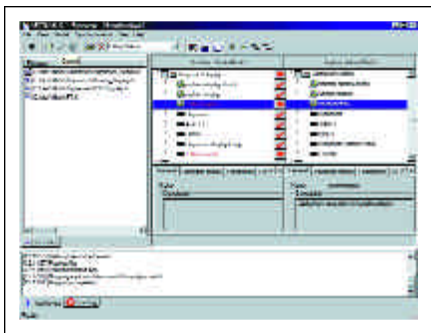


Code generation is making its mark

To be able model a real time system before implementation is a facility that many design teams should consider as a means of implementing systems and identifying flaws. The concepts of system modelling are now being supplemented by automatic code/class generation technology. Products such as Artisan Real-Time Studio Modeller/Professional, Rational Rose Modeller/Real-Time, i-Logix Rhapsody and Objectime Developer, all support the design process by allowing the developer to model and interactively develop the application before committing the design.

Automatic code generation is a facility available in some of these tools. Sometimes used originally on mainframes and mini-computers, it is now available as part of modelling and object oriented design tools like Artisan's Real-Time Studio Professional, and mathematical modelling tools like Matlab. Other tools like Beach Solutions' EASI focus on the generation of the low-level C libraries, to ensure that the hardware interaction is formally documented and controlled. It is possible to use combinations of these tools, to create systems code, but a rigid test and verification regime is needed.



Artisan's C++ Reverser

Language choices

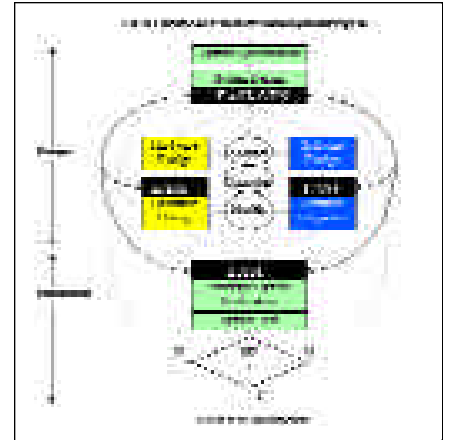
There are obvious choices for code generation systems, C, C++ (or EC++) or Java. A great number of systems are still being developed in C, developers accept the limitations and unsafe features and hopefully protect themselves from them, using paren-

thesis to enforce precedence etc.. Large scale systems obviously benefit from object oriented techniques and C++ is the language of choice, or so it seems. The big question then is, what is the nature of the code to be generated, playing safe and generate C routines? That is the choice given to users of Artisan's Real Time Studio, it now enables engineers using C to benefit from modelling in UML, including Round-trip engineering and animation. At the same time, Real Time Studio offers three options and implement a class in C++:

- Round-trip Engineering of interfaces - allows an engineer to program in his/her IDE, but keep class structure and interfaces up to date with the class model enabling clear documentation and easy visualisation.
- Maintain operation bodies - allows the user to benefit from the rich-text editor. This is model-aware, containing object-references, not just text. Accessed attributes and operations can simply be dropped into the editor, ensuring correct spelling and maintaining consistency as attribute names change.
- State models for more complex class behaviour, including model-aware rich-text for guards and actions. Complete operation bodies for state model behaviour are generated, complete with test harnesses. The resulting state model simulation can be animated and has a standard API for integration with external systems.

Low level control

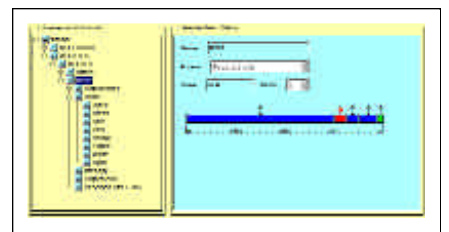
The Beach Solutions' EASI software allows the user to generate low-level hardware drivers, after entering a complete description of the environment. The user is prompted for information describing system address map, peripheral cells, registers, bits, groups of bits, their polarity levels and any programming sequences to be observed. As the user enters information it is checked for duplication, address conflicts, and correct bus width access'. Appropriate warning messages are relayed to the user indicating any data entry errors made. All information entered is saved in a database from



EASI-GEN + EASI-P enhance the lifecycle

which the EASI-GEN GUI builds the EASI-Library. Functions in the EASI system are generated at two levels:

- Level 1 - These functions are generated to access every bit-group within the peripheral cells of a system. Each function is generated with a unique name which includes a prefix comment to all functions within the library. These functions are available for inclusion as Macros or subroutines (C functions).
- Level 2 - Are composed entirely of Level 1 functions. Peripheral devices can be supported with a library of EASI Level 2 functions, producing a reduced simplified API. IP vendors can provide an API with their hardware which supports only legal programming options.



EASI-GEN input capture

Ease of use

Systems like this need to be easy to install and use. A complex final stage of code generation will be off-putting and users might leave the tool behind once they have used it to understand the problem.