Creating ‘correct-by-construction’ code
Using high-level language and formal methods

Gene-Auto is developing a software code generator for companies involved with real-time constrained embedded systems. This generator will produce ‘correct-by-construction’ C language code from standard high level languages such as Matlab-Simulink using formal mathematical methods on a large scale. This innovative step will strengthen Europe’s competitiveness in the relevant technological domains and enable the industrial partners of the consortium to maintain their leading positions in their respective markets.

Various symposia have emphasised the need for optimised development cycles, reduced development time, design-to-cost and reusability of design and that auto-coding was the future of embedded systems coping with severe real-time constraints, such as for automotive electronics. Gene-Auto is tackling these issues directly, focusing on the market for specific embedded systems.

The main drivers for this project are:
• **Modelling and autocoding**: a way to improve early verification and validation with an incremental approach;
• **Autocoding**: a way to increase reuse of building blocks at both functional and software levels through application of the same language and design framework;
• **Qualification and/or certification of autocode**: the code generator itself can be qualified/certified and the integrity and correctness of the code generated checked using formal verification and validation techniques in a broadly automated manner; and
• **Open-source approach**

**Boosting system development**
The level of technological innovation expected at the end of the project is to have system development departments using:
• A common mathematical functional language such as Matlab-Simulink or SciLab/SciCos;
• Formal verification and validation methods on models before the software code is generated that make it possible to:
  - Verify completeness, consistency and coherency of the solution;
  - Verify real-time behaviour;
  - Modify the designs whenever needed, taking into account the results of such verifications; and
• Automatic full serial certified software code generation out of these models.
Speeding development and boosting productivity
Gene-Auto addresses a unique toolset environment for code generation of synchronous and asynchronous software. The use of this desktop-based toolset supporting the design and validation processes makes it possible to shorten development cycles and hence increase productivity, a major stake for the highly real-time constrained embedded systems market.

This development will involve a series of steps:

1. **User needs expression:**
   - Availability of the needs, as well as the presentation of the case-studies that will be used for the assessment:
   - First release jointly agreed requirements specification
   - Case-study presentation for each of the “users”

2. **Development start:**
   - End of the preparation phases (architectural design, quality manuals, etc.) and start of the actual development:
   - Architectural design of the first version of the toolset
   - State-of-the-art report on formal methods
   - User experience report verification of embedded software
   - Tool qualification plan
   - Tool software quality assurance plan

3. **Expression on user needs validation approach:**
   - Detailed models of the case studies and user needs validation approach:
   - Case-study presentations in more detail
   - Report on the validation approach

4. **First toolset version delivery and validation:**
   - First version of the toolset and the first validation efforts:
     - First version of the toolset available
     - Second release of the jointly agreed requirements specification (taking into account results on the formal methods)
     - Report on issues to be corrected in the second version of the toolset
     - Specification methods and tests report

**Embedded software code generator**
The major results of Gene-Auto are intended to be:
- Joint user requirements specification for an embedded software code generator, with supported ‘input’ formats as an essential part of the specification. A complete set of validation activities in the perspective of the toolset qualification will also be defined;
- Development of an approach and necessary tools for the application of formal techniques for verification and validation to be used with the embedded code generator. These techniques will be applied to the designed models, the code generator and the generated code; and
- An open-source code generator prototype able to produce real-time embedded qualified software.