Enormous diversity in:

- Standard coding rule sets do exist, e.g.:
  - Javacard: sponsored by a private company
  - High-Integrity C++ (HICPP): sponsored by a private company
  - MISRA-C

Static enforceability

- Analysis techniques required
- Program features involved

Constrain admissible constructs (e.g. forbidding error-prone features or coding styles) to help producing safer code.

About a Program Knowledge Base

Rule HICPP 3.3.15:

“Ensure base classes common to more than one derived class are virtual.”

Natural language is inherently ambiguous: Which inheritance links must be tagged as “virtual”?

A framework to formalise coding rules is necessary to statically check that programs conform to a given set. We are developing such a framework in the environment of the GGCC project.

Rule Formalisation

Based on first-order logic and written in a domain-specific language which is translated into Prolog and that:

- Formalises standard coding rule sets in a declarative style
- Makes it easier for the final user to define additional coding rules
- Provides a collection of predefined predicates about program facts (such as class/1, base_of/2, or in_call_graph/2)
- Quantification over certain domains
- Constructive negation

We focus on structural rules, which deal with relationships between static entities in the code (classes, member functions, etc.), e.g.:

Rule HICPP 3.3.15 translated into Prolog

\[
\text{violate_hicpp_3_3_15}(A, B, C, D) :-
\text{class}(A), \text{class}(B), \text{class}(C), \text{class}(D), B \subseteq C,
\text{direct_base_of}(A, B), \text{direct_base_of}(A, C),
\text{base_of}(B, D), \text{base_of}(C, D),
\langle \downarrow \text{virtual_base_of}(A, C).
\]

We do not code the rule itself, but its negation. Any program that satisfies the negated rule thus violates the coding rule. Predicates coding rule violations are queried against facts describing a program. A successful resolution flags a rule violation, providing a witness.

Work partially supported by PROFIT grants FIT-340005-2007-7 and FIT-350400-2006-44 from the Spanish Ministry of Industry, Comunidad Autónoma de Madrid grant S-0505/TIC/0407 (PROMESAS), Ministry of Education and Science grant TIN2005-09207-C03-01 (MERIT/COMVERS) and EU IST FET grant IST-15905 (MOBIUS).