Summary: This paper describes how an abstract programming interface and its implementation can be generated from the syntax definition of a data type. In particular we describe how a grammar (in SDF) can be used to generate a library of access functions that manipulate the parse trees of terms over this syntax. Application of this technique in the ASF+SDF Meta-Environment has resulted in the elimination of 47% of the handwritten code, thus greatly improving both maintainability of the tools and their flexibility with respect to changes in the parse tree format. Although the focus is on ATerms, the issues discussed and the techniques described are more generic and are relevant in related areas such as XML data-binding.

MSC:

68N30 Mathematical aspects of software engineering (specification, verification, metrics, requirements, etc.)
68Q65 Abstract data types; algebraic specification

Keywords:
API; ASF+SDF; ATerms; Data binding; Code generation; Parse tree manipulation

Software:

Castor; CoFI; mCRL; ASF+SDF; ATERM; Stratego

References:

[1] ()
[6] van den Brand, M.G.J.; Ringeissen, C., ASF+SDF parsing tools applied to ELAN, () · Zbl 0962.68032
[7] van den Brand, M.G.J.; Scheerder, J.; Vinju, J.J.; Visser, E., Disambiguation filters for scannerless generalized LR parsers, (), 143-158 · Zbl 1051.68874
[13] ()
[21] de Jonge, M.; Visser, J., Grammars as contracts, () - Zbl 1023.68741
[27] Visser, E., Stratego: A language for program transformation based on rewriting strategies. system description of stratego 0.5, (), 357-361 - Zbl 0981.68679

This reference list is based on information provided by the publisher or from digital mathematics libraries. Its items are heuristically matched to zbMATH identifiers and may contain data conversion errors. It attempts to reflect the references listed in the original paper as accurately as possible without claiming the completeness or perfect precision of the matching.