A Calculus for Uniform Feature Composition

SVEN APEL University of Passau and DELESLEY HUTCHINS MZA Associates Corporation

The goal of *feature-oriented programming* (FOP) is to modularize software systems in terms of features. A *feature* refines the content of a base program. Both base programs and features may contain various kinds of software artifacts, for example, source code in different languages, models, build scripts, and documentation. We and others have noticed that when composing features, different kinds of software artifacts can be refined in a uniform way, regardless of what they represent. We present *g*DEEP, a core calculus for feature composition, which captures the language independence of FOP; it can be used to compose features containing many different kinds of artifact in a type-safe way. The calculus allows us to gain insight into the principles of FOP and to define general algorithms for feature composition and validation. We provide the formal syntax, operational semantics, and type system of *g*DEEP and outline how languages like Java, Haskell, Bali, and XML can be plugged in.

Categories and Subject Descriptors: D.3.1 [**Programming Languages**]: Formal Definitions and Theory; D.3.3 [**Programming Languages**]: Language Constructs and Features

General Terms: Design, Languages, Theory

Additional Key Words and Phrases: Feature-oriented programming, feature composition, type systems, principle of uniformity

ACM Reference Format:

Apel, S. and Hutchins, D. 2010. A calculus for uniform feature composition. ACM Trans. Program. Lang. Syst. 32, 5, Article 19 (May 2010), 33 pages. DOI = 10.1145/1745212.1745216. http://doi.org/10.1145/1745212.1745216

 $DOI = 10.1145/1745312.1745316 \ http://doi.acm.org/10.1145/1745312.1745316$

DOI 10.1145/1745312.1745316 http://doi.acm.org/10.1145/1745312.1745316

ACM Transactions on Programming Languages and Systems, Vol. 32, No. 5, Article 19, Publication date: May 2010.

This work was funded in part by the German Research Foundation (DFG), project number AP 206/2-1.

Authors' addresses: S. Apel, Department of Informatics and Mathematics, University of Passau, Innstr. 33, 94032 Passau, Germany; email: apel@uni-passau.de; D. Hutchins, MZA Associates Corporation, 2021 Girard Blvd. SE, Suite 150, Albuquerque, NM 87106-3140; email: delesley@gmail.com.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org. © 2010 ACM 0164-0925/2010/05-ART19 \$10.00