LSINF 2335

PROGRAMMING PARADIGMS: THEORY, PRACTICE AND APPLICATIONS

Theme: REFLECTION & META-PROGRAMMING

Individual Report 2013–2014

Nom_1 $Prénom_1$ – $NOMA_1$ – $email_1$
Nom_2 $Prénom_2$ – $NOMA_2$ – $email_2$

Chosen language

1. Chosen Language

- What language have you chosen (and why)?
- What kind of programming paradigm does this language belong to (functional, procedural, logic, object-oriented, multi-paradigm, …)?
- Give a brief introduction to the core syntax / semantics / concepts of that language.
- Give an illustrative *working* code example of a typical program written in that language.
- What kind of typical applications is the language targeted at?

2. Reflection and meta-programming

2.1 Reflective features

- What language features for dealing with reflection and meta-programming does the chosen language provide?
- What *kinds* of reflection and meta-programming features does that language offer?
- What is the MOP (meta-object protocol) for that language?
- What are the limitations of the reflective features provided by this language?
- Illustrate your explanations with working code fragments.

2.2 Applications of reflection

- What are the typical applications that reflection could be used for in this language?
- Can you give a *working* code example of such a typical problem that requires a reflective solution?
- Does there exist a "killer-app" for this language that has been implemented with reflection?

2.3 Comparison with other languages

- How does this language compare to Smalltalk, Java or Ruby from the point of view of the reflective features it supports, the kinds of reflection it offers, or its MOP?
- What can this language learn from those languages?
- Does it offer some specific reflective features that you do not have in either Smalltalk, Ruby or Java? (Can Smalltalk/Java learn something from reflection in this language?)

- Does it offer some powerful native (non-reflective) features that allow you to express things for which you would need reflection in other languages (like Smalltalk, Ruby or Java)?

3. Conclusion

- In conclusion, how good does this language score as a reflective language?
  - Does it provide a very rich, well-structured and well-supported set of reflective features that are supported by the programming environment as well?
  - Are there only a few ad-hoc reflective features that are not well supported by the environment?
  - What can other (reflective) languages learn from this language?
  - What can this language learn from how reflection is dealt with in other languages?

4. Bibliography

What books, articles, web pages, … did you use as input for this document? Please refer to these references at the adequate places in the text by means of an indication [Bibi,1999] and list all these references in full detail in this section using the following format.

For books :

[Author, Year] AuthorName FirstName, "Title of Book", Publisher, Year. ISBN.

For articles :

[Author, Year] AuthorName FirstName, "Title of Book or paper". Name of Journal of Proceedings, Editor, Series, Publisher, PageNumbers, Year.

For web pages :

[Topic] "URL" : Short description.

Exemples :

[Kölling&Rosenberg, 2002] Michael Kölling, John Rosenberg, "BlueJ --- The Hitch-Hikers Guide to Object Orientation", The Maersk Mc-Kinney Moller Institute for Production Technology, 2002.

[Java2 5.0 API] "http://java.sun.com/j2se/1.5.0/docs/api/" : Overview Java 2 Platform SE 5.0; API Specification.

[Kellens&al., 2006] Andy Kellens, Kim Mens, Johan Brichau & Kris Gybels. "Managing the Evolution of Aspect-Oriented Software with Model-based Pointcuts". Proceedings of the European Conference on Object-Oriented Programming (ECOOP 2006), D. Thomas, LNCS 4067, Springer-Verlag, pp. 501–525, 2006.