LSINF 2335

Programming Paradigms: Theory, Practice and Applications

Theme: Reflection & meta-programming

Individual Report 2013–2014

Capron Romain - 2140-08-00 - romain.capron@student.uclouvain.be

Marchal Antoine - 5462-08-00 - antoine.t.marchal@student.uclouvain.be

Python

UCL

**Université
catholique
de Louvain**

Contents

1 Chosen Language

• What language have you chosen (and why)?

We have chosen Python that is a ... programming language. • 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.

```python
a, b = (1, 1)
while b < 10:
    print 'a={0}, b={1} and a+b={2}'.format(a, b, (a+b))
    a, b = (b, a + b)
```

We can run this program by executing the following command :

```
$ python example.py
```

And this is its output :

```
a=1, b=1 and a+b=2
a=1, b=2 and a+b=3
a=2, b=3 and a+b=5
a=3, b=5 and a+b=8
a=5, b=8 and a+b=13
```

• What kind of typical applications is the language targeted at?

This language can be used in various kind of application domains such as :[1]

1. Web and Internet Development :

   • Frameworks such as Django and Pyramid

   • Micro-frameworks such as Flask and Bottle

   • Advanced content management systems such as Plone

2. Scientific and Numeric :

   • SciPy is a collection of packages for mathematics, science, and engineering

   • Pandas is a data analysis and modeling library

3. Education : We learned programming with Java but Python seems to be more appropriate as it has a simpler syntax for a similar behaviour.

---

[1]https://www.python.org/about/apps/

4. Software Development : Even big softwares are done in Python. For example the well known game *Sid Meier's Civilization IV* has been nearly completely implemented in Python.

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)?

and meta-programming

3   Conclusion

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