COMPSCI 725

Security Oral Presentation
Jacob Hendrickx

G. Frantzeskou, S. MacDonell, E. Stamatatos, S. Gritzalis,
"**Examining the significance of high-level programming features in source code author classification**",
in Journal of Systems and Software 81:3, pp. 447-460, 2008.
DOI: 10.1016/j.jss.2007.03.004

Summary

Scientific form:

- Aim
  - Find/examine high-level, language independent, programming features for automated source code author identification
- Methods
  - The SCAP (source code author profile) approach
  - Metrics and languages examined
- Data analysis
  - Contribution of chosen metrics on high-level programming features of chosen languages
- Summary & Conclusions

Appreciative Comment

Thought provoking metrics:

Layout: Form and shape of the code; white space

Style: Naming preferences

Structure: Lines/keyword distribution

Linguistic: Sense of language used

Article focused on:

- Comments (large influence on both languages)
- Layout (large influence on Java, smaller for Lisp)
- Identifiers (symbol (small) & package (large))

Critical Comment

”On the other hand, systems that deal with plagiarism detection could use the findings of our work in order to locate the features of a piece of code that could be plagiarised.”

- There is only brief discussion of practical use and this has some fallacies
- They have a hammer, but where are the nails?

Explaination

- Cyber crime
  - Absence of source and comments
- Plagiarism
  - Lack of supportive samples
- Authorship dispute
  - Influence of intended evasion

Question?

What is the most defining feature of your code that makes it yours and not someone elses?

Who changes their style dependant on language or task? Is it because we are 'developing' developers?