

What is software?

Speaker: Xiaowei Wang

Supervisor: John Mylopoulos, Nicola Guarino

University of Trento, Italy Laboratory for Applied Ontology (ISTC-CNR), Italy xwang@disi.unitn.it



Outline





1. Introduction



Software has been used in oftware engineering brings the aspect of human activities ngineering methodologies into the development of software.



The meaning of

"software + engineering"

• Naur and Randell,1968

The phrase '*software engineering*' was deliberately chosen as being provocative, in implying the need for software manufacture to be based on the types of theoretical foundations and practical disciplines.

• Panas,1997

Software *engineering* should be classified as a branch of engineering instead of a branch of computer science.

• Osterweil, 2008

What is *software*?

Understandings of software

- Algorithm (e.g. a bubble sorting algorithm)
- Source code (e.g. encoded in Java/C)
- Realization of source code (e.g. the code stored on a hard disk)
- Running process of algorithm (e.g. sorting process running in a computer)
- Specification document?
- Design document?
- > Requirements?

We try to provide a *unified concept* of software.

2. Literature Work

• Section 1 >> Section 2. Literature Work >> Section 3 >> Section 4 >> Section 5

Osterweil: multiple kinds of software

- Software is *non-physical* and *intangible*, containing *instructions* which manages and controls of tangible entities.
- Processes (e.g. waterfall model process)
- Laws (e.g. copyright law)
- Recipes (e.g. how to cook an apple pie)

Eden: program language expression

- Software is an *expression* "S" based on a turing-complete programming *language* "L".
- S should be a well-formed expression in L.
- S should be accepted by a compiler.
- S should support a non-trivial set of instructions (expressed in a turing-complete language).

Obelrle: ontology of software



Ruttenberg: Information entity



Section 1 >> Section 2. Literature Work >> Section 3 >> Section 4 >> Section 5

Ruttenberg: Information entity, concretized



3. Research Baseline



"Ithis and tog Biennohio bide"."





"Weily addition the state of th



Richards: Semiotic triangle



4. Research Approach



Software triangle



Software abstraction layers



Consistency in vertical



Consistency in horizontal



Software Representation (Syntax)

Software System (Referent)

Further thought on the basline



5. Conclusion

- Several *literature works* discussing the concept of software are illustrated.
- We try to provide a *unified concept of software*, by implying the idea of semiotic triangle and software abstraction layers.
- More detail analysis should be taken out, the contribution of the first step of my research could be an *ontology of software*.

The end

Thanks!

QUESTIONS?

References

- [1] Eden, A.H. and Turner, R. 2007. Problems in the ontology of computer programs. Appl. Ontol. 2, 1 (2007), 13–36.
- [2] Guarino, N. 2012. Artefactual Systems, Functional Roles and System Components. 1101–1116.
- [3] Jureta, I.J. et al. 2009. A core ontology for requirements. Applied Ontology. 4, 3 (Jan. 2009), 169–244, DOI= 10.3233/AO-2009-0069.
- [4] Naur, P. et al. 1969. Software engineering : report on a conference sponsored by the NATO Science Committee, Garmisch, Germany, 7th to 11th October 1968. (Brussels, 1969).
- [5] Oberle, D. et al. 2009. An Ontology for Software. S. Staab and D. Rudi Studer, eds. Springer Berlin Heidelberg. 383–402, DOI= 10.1007/978-3-540-92673-3_17.
- [6] Ogden, C.K. et al. 2001. The Meaning of Meaning: A Study of the Influence of Language Upon Thought and of the Science of Symbolism. Routledge.
- [7] Osterweil, L.J. 2008. What is software? Autom. Softw. Eng. 15, 3-4 (2008), 261– 273.
- [8] Parnas, D. 1997. Software Engineering: An unconsummated marriage (extended abstract). Software Engineering — ESEC/FSE'97 SE - 1. M. Jazayeri and H. Schauer, eds. Springer Berlin Heidelberg. 1–3, DOI= 10.1007/3-540-63531-9_1.