

Summary of Research Papers on AI in SE

AI and SE: Status and Future Trends:

Jörg Rech, Klaus-Dieter Althoff

Coverage: contributions of Knowledge Based Systems, Ambient Intelligence, Computational Intelligence in SE

Pros: Definition of KBS, Diagram of AI in SE.

Ways of Applying Artificial Intelligence in Software Engineering:

Robert Feldt, Francisco G. de Oliveira Neto, and Richard Torkar Chalmers and the University of Gothenburg

Coverage: AI-SEAL and how to use it, Taxonomy: Point of Application, Type of AI and Level of Application

Pros: Talks about risk in using AI, categorises the types of AI, categorises the risks in PA.

Cons: Does not define Level of Automation (LA).

Conclusion: Faceted taxonomy is the safest way to approach AI in SE. By doing so, we can pin an optimal area in which the AI can be applied and with minimal risk of diverting the final product from the expected output.

Artificial Intelligence and Systems Engineering:

Ian Sommerville,

*Computing Department, Lancaster University, LANCASTER
LA1 4YR, UK.*

Coverage: Why there exists a hostility between Software Engineers and Artificial Intelligence.

Pros: Talks about the failure of REFINE, and basically the conflict between AI and software engineers

Software Engineering Using Artificial Intelligence Techniques: Current State and Open Problems

*Hany H Ammar, Walid Abdelmoez, and Mohamed Salah Hamdi,
The Lane Department of Computer Science and Electrical
Engineering, West Virginia University, USA,
Computer Science Department, Faculty of Computers and
Information, Cairo University, Egypt Arab Academy for Science,
Technology and Maritime Transport, Egypt
Ahmed Bin Mohamed Military College, Qatar*

Coverage:

1. Application of artificial intelligence approaches to the software engineering processes. Impact on reducing time and improving the quality of software in general.

2. Requirement Analysis:

1. Processing Natural Language Requirements (NLR) into specifications and design automatically.
2. Tool: Class-Model Builder (CM Builder), a NL based CASE tool that builds class diagrams specified in UML from NL requirements documents.

3. Knowledge Based Systems (KBS):

1. Store design families in KBS.
2. The system searches the KBS and proposes a design schema.

4. Intelligence Computing for Requirements Engineering:

1. Tool: SPECIFIER, a CASE based system that takes as input an informal specification of an operation.

5. Coding and Testing:

1. An expert system capable of interacting with the human programmer exactly the same way as a human assistant would.
2. A system that can generate functions, data structures, or the entire program directly from the specifications.
3. Tool: ATGen, a software test data generator.

6. Open problems:

1. Disambiguating natural language requirements

2. Developing knowledge based systems and ontologies to manage the requirements and model problem domains

3. The use of computational intelligence to solve the problems of incompleteness and prioritization of requirements.

Artificial Intelligence Techniques in Software Engineering (AITSE)

Engr. Farah Naaz Raza

Proceedings of the International MultiConference of Engineers and Computer Scientists 2009 Vol I IMECS 2009, March 18 - 20, 2009, Hong Kong

Coverage:

1. A comparative study between software development and expert system development.
2. Study of how expert programmers analyze, synthesize, modify, explain, verify and document programs, and to apply that theory towards automating the programming process.
3. Automatic Programming System (APS).
4. Genetic Code: Technique which enables computers to solve problems without being explicitly programmed. It works by using genetic algorithms to automatically generate computer programs

It Takes Three to Tango: Requirement, Outcome/data, and AI Driven Development

Jan Bosch, Helena H. Olsson and Ivica Crnkovic

Chalmers University of Technology, Department of Computer Science & Engineering, Göteborg, Sweden

Malmö University, Department of Computer Science and Media Technology, Malmö, Sweden

Coverage:

1. With automated practices for data collection and analysis, queries can be processed frequently to provide software developers and managers with rapid feedback and as a result, continuous improvements can be made to the systems.
2. AI driven development approach where the company has a large data set available and uses artificial intelligence techniques such as machine learning and deep learning to create components that act based on input data and that learn from previous actions.