**Paper Title**: Model-Driven Engineering

**Author(s):** Douglas C. Schmidt

**(1) Short summary of the problem / context.**

*The need for abstraction in software development and how MDE can provide it.*

**(2) Short summary of the paper's main contribution (1-2 paragraphs).**

*Firstly, paper mentions why abstraction is needed for software development. Then, it explains previous similar attempt CASE, its achievements, its failures and why it couldn’t survive in software industry. Later, he indicates the current software development limitations and how it is getting more and more complicated each year. After that, he mentions the MDE and how it can get over those difficulties. He also refers that MDE isn’t sufficient for large and complex projects. He indicates MDE needs new updates for today’s software industry features like concurrency engineering and synchronization between models. Lastly, there are 2 sidebars about how MDE works in R&D software development.*

*First sidebar mentions how MDE and MDE tools work in large and distributed systems. How they use MDE in reverse engineering with DSML, automated partial implementations and model verification. The second one points out how MDE affects QoS requirements of a project. They also refer that finding and fixing the bugs are cheaper with MDE due to the earlier detection rates.*

**(3) Who would be interested in the paper?**

*People who are working in software industry or researching software development processes. Especially, software developers who are working in specific areas/domains like aviation, finance, embedded systems, etc.*

**(4) List 1-3 strengths of the paper (1-2 sentences each)**

**S1*.*** *2 sidebars that explains the real world examples of MDE in industry projects.*

**S2.** *Mentions previous attempts and what did they learn from it.*

**S3.** *Explains the main subject as simple as possible.*

**(5) List 1-3 weaknesses of the paper (1-2 sentences each)**

**W1.** *Mentions the need for abstraction, CASE and MDE. However, none of them are described deeply.*

**W2.** *Format of the paper is not great. Sidebars split the author’s sentences. While reading the paper, I have to scroll down a lot to finish the rest of the sentence. This causes lose of attention to ideas.*

**W3.** *Software engineers could not know every development method. Therefore, longer definitions could create more attention to the paper.*

**(6) Paper presentation**

[ ] Very well written

[x] Generally well written

[ ] Readable

[ ] Needs considerable work

[ ] Unacceptably bad

Justify your choice in 2-3 sentences

*Language is simple and understandable. Introduction and the definition of the problem are straightforward. However, the flow of the paper is not great. Sidebars divide the author’s sentences into two parts. Since sidebars are nearly a page long, skipping these long parts can cause lose of attention to ideas. Also, sidebars’ languages’ aren’t as simple as the author’s one.*

**(7) Technical Depth**

[ ] Very good (comparable to best conference papers)

[ ] Good (comparable to typical conference papers)

[x] Marginal depth

[ ] Little or no depth

Justify your choice in 2-3 sentences

*The paper is not technically deep. Author mentions the abstraction in software engineering but doesn’t mentions the technical part of it. Also, CASE and MDE definitions and problems are shallow. Most people can understand the outline of the author’s ideas.*

**(8) Impact/Significance**

[x] Very significant

[ ] Significant

[ ] Marginal significance.

[ ] Little or no significance

Justify your choice in 2-3 sentences

*This paper is the most cited paper for Model Driven Engineering with over 3000 references.*

*It encourages the software R&D industry to use MDE in their projects.*