

# Recent Topic Trends at the Annual SAICSIT Symposium

Stefan Gruner

Department of Computer Science, University of Pretoria.  
[sg@cs.up.ac.za](mailto:sg@cs.up.ac.za)

This issue of SACJ contains an interesting meta-review by Paula Kotze and Alta van der Merwe on the research topics presented in this journal since its early days. I have recently done a similar (though considerably less thorough) review of the topic trends at the annual SAICSIT symposium, to which this journal, SACJ, is closely linked via the SAICSIT organisation. The results of my findings are described in the following.

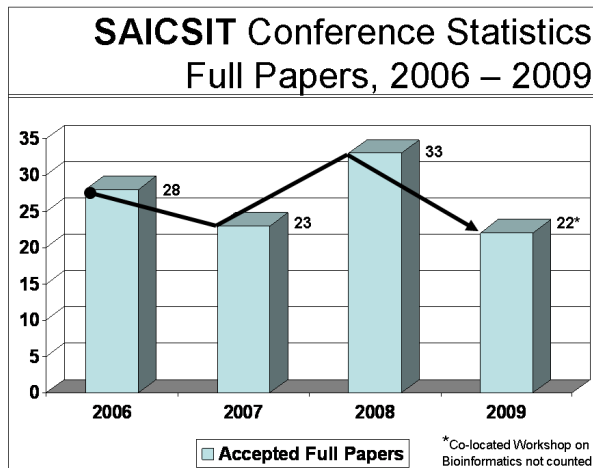


Figure 1.

Figure 1 shows the total numbers of accepted and presented *full papers* (not: short papers or posters) at the SAICSIT conferences from the years 2006 to 2009, whereby the co-located (but otherwise independent) workshop on Bioinformatics in the year 2009 has not been taken into account. Within that 4-year period, the symposium had reached its maximum popularity in the year 2008 (with 33 full papers) and dropped down to its minimum this year (with only 22 full papers).

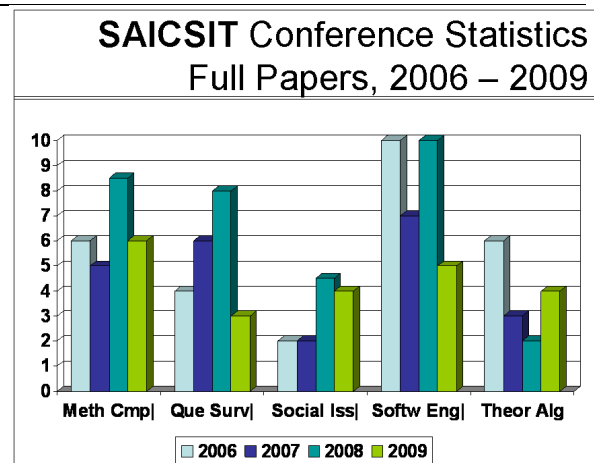


Figure 2.

Figure 2 shows my analysis of the topic frequencies, in total numbers (not percentages) at the SAICSIT symposium during those 4 years. For this purpose I have defined the following five topic clusters:

- Methodological discussions and comparisons
- Surveys and questionnaire-based research
- Social issues (including: specifically African problems)
- Software engineering and software tool development
- Theory and algorithms, (including: core Computer Science, AI, data mining, etc.)

The smallest unit of counting was 0.5 such that I could better take into account “hybrid” papers with overlapping topic areas. For example, one paper with a questionnaire survey about social issues of IT in Africa would be “split” and count 0.5 towards the category “social issues” and 0.5 towards the category “surveys and questionnaire-based research”. Similarly there were other overlaps between different categories, for example between core computer science (algorithmics) and software engineering, or between software engineering and methodological comparisons, in which I have also done split-counting on the basis of  $(0.5 + 0.5) = 1$ .

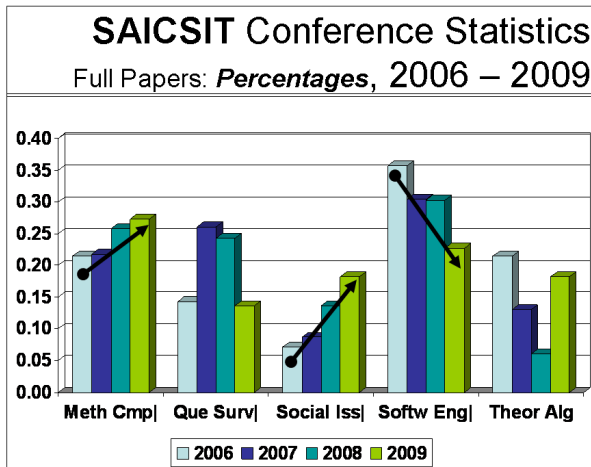


Figure 3.

Figure 3 shows the same data, but now in percentages, normalised against the total number of full papers at the symposium in every year. In this perspective, the recent trends at SAICSIT come out very clearly: Rather “soft” or “easy” topics, such as discursive method comparisons, or studies about social issues, were clearly on the rise during those four years, whereas the rather “hard” or “difficult” topic of software engineering has been on the decline throughout. No obvious trend, however, can be seen from Figure 3 in the category of core computer science (Theor./Alg.); ditto in the category of surveys and questionnaire-based research, during those four years.

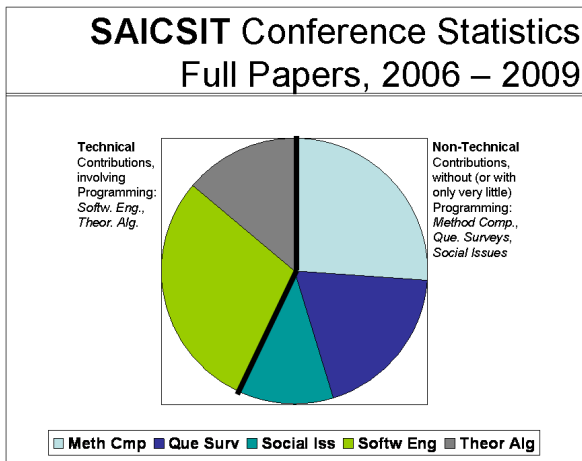


Figure 4.

Figure 4 shows how I have further clustered those five topic areas, as mentioned above, into two larger groups, namely

- Technical papers (left-hand side of Figure 4), and
- Non-technical papers (right-hand side of Figure 4),

whereby the technical papers are those ones based on the construction of technical artefacts, such as algorithms or software prototypes. In other words: the technical papers rely strongly on implementation and programming, whereas the non-technical papers do not (or at least not so much).

Here we see very clearly that, over the observed period of the last four years, the non-technical papers at SAICSIT hold the absolute majority over the technical contributions at the symposium.

**Questions about Software Engineering**

The observed decline of software engineering, in combination with the anyway rather small total number of papers, at SAICSIT during the past four years seems somewhat worrying to me. In this context we ought to ask ourselves:

- whether this trend at SAICSIT is representative of the status of software engineering in the South African computer science research institutions in general?
- or whether South African software engineering – though thriving elsewhere– has moved away from SAICSIT for whatever reason?

**Recommendation to SAICSIT’s Steering Committee**

On the basis of my little review, with its results as shown above, I would like to suggest to the respected Steering Committee of SAICSIT to split the annual symposium into two co-located but otherwise independent sub-conferences, each one with its own independent Programme Committee: one for the technical contributions (see Figure 4), and another one for the non-technical contributions (see Figure 4 again).

To strengthen the important technical field of software engineering in South Africa, a dedicated Programme Committee of a new SAICSIT sub-conference would be needed; such a goal seems rather hard to achieve within the SAICSIT symposium’s current, “mixed” CS/IS structure and its clearly visible bias towards the non-technical kind of contributions. Such a split-move might then perhaps turn the trend and might perhaps lead to a re-increasing attractiveness of SAICSIT for South Africa’s software engineers – of course I cannot know this with certainty, but it might perhaps be worth a try, just to find it out.

A first step into such a direction was already done at SAICSIT 2009, with its co-located (but otherwise independent) workshop for bio-informatics. A similar, independent but co-located SAICSIT sub-conference for “core” computer science and software engineering should also endorse and encourage special demo-sessions for software prototype presentations, to showcase South Africa’s implemented contributions to this increasingly important branch of South Africa’s national economy, which is software development.