"Civilization advances by extending the number of important operations which we can perform without thinking about them." – Alfred Whitehead

The topic of administration cost is growing more important every year and the problems that require our research and development are changing. Why is that? The rise of the Internet, online transaction processing, online banking, and the ability to connect heterogeneous systems have all contributed to the massive growth in data volumes over the past 15 years. Terabyte sized databases have become commonplace. Concurrent with this data growth, have come dramatic increases in CPU performance, spurred by Moores Law, and improvements in disk fabrication which have improved data density limits for persistent disk storage. At the same time there has been a large shift in the nature of the data we manage. Over the past decade data has made a dramatic shift from being predominantly well structured, to being increasingly unstructured (such as image, audio data streams, and video) and semi-structured data (such as XML). Users naturally expect the same benefits that relational databases achieved with structured data to apply to new data types as well – concurrency control, performance, data manipulation, rich language support, availability and recovery, security and authority to name a few. The increasing volumes of data, and the changing kinds of data we manage also spur "creeping featurism" as clever scientists and engineers find new ways to provide function over volume, scale and content. Creeping featurism adds administrative complexity to systems. We end up with more data, more features, more options, more things to think about and a lot more complexity. End users end up spending more on the staff to manage the complexity of information technology than on the products they purchased. Because featurism will not cease to expand our paradigm must change.

The urgent need to reduce administrative complexity is not unique to database systems and is pervasive throughout the IT industry. I'd like to share with you a very simple and hopefully entertaining analysis: In a brief examination of 3 popular middleware products produced by different companies, we observed the number of configuration/registry parameters ranged from 384 to 1200. These parameters were used to specify everything from memory configuration to connection and process limits. Many of the parameters have a dynamic range of potential values (for example, configuring memory allocations to the database caching areas can have a wide range of values). The most extreme simplification of the configuration space considers each configuration parameter as having a binary setting (ON/OFF or TRUE/FALSE etc). Using this gross oversimplification, and completely excluding the complexity of logical and physical design, the possible configurations for a product with 384 parameters is 10^{115} while a product having 1200 binary parameters would have 10^{361} configurations. These numbers are monstrous, far beyond the ability of human beings to assess. Astrophysicists estimate the upper bound on the number of atoms in the universe[1] to be 10^{81}, many orders of magnitude less than the number of ways one can configure the three middleware products we studied, even with dramatic oversimplifying assumptions. It matters because the alternatives are bleak: poorly administered high cost systems that run at fractions of their potential. The answer to this is self-managing autonomic systems that just work without fuss and bother.

The following topics generally frame the domain for self-managing information management systems and are the areas where continued research and development are required for our industry to be successful in significantly reducing administrative cost.

1. *Self-Configuring*: Information management systems that are trivial to set up.
2. *Self-Healing*: Information systems that know their own problems.
3. *Self-Optimizing*: Information systems that maximize their own performance/efficiency.
4. *Trust in self-managing systems*: How is trust in automation fostered when, especially when the financial stakes are high?

[1]Current estimates are in the range of 10^{63} to 10^{81} atoms. I have used the upper bound (worst case) to strengthen the argument.

5. *Benchmarking of self-managing information systems*: How are self-managing information systems best evaluated?

6. *Self-Protecting*: Databases that protect themselves from security attacks.

7. *System-wide self-management*: Systems that are self-managing as an integrated whole, not only by components.

To advance research and development in self-managing database and information management systems, the IEEE Data Engineering Workgroup on Self Managing Database Systems (which was formally announced a few months ago) will sponsor workshops in conjunction with the ICDE, foster publications, and maintain an online collection of links to key resources. I am pleased that our initial executive committee includes some of the leading advocates of self-managing database technology from across geographies in both industry and academia. Dr. Guy Lohman is organizing a workshop on Self Managing Database Systems at ICDE 2007. We invite readers to follow the news and announcements about the workgroup's activities at our website: http://db.uwaterloo.ca/tcde-smdb/.

I'm delighted to see the this special edition of the bulletin in print, and extend my thanks and appreciation to Prof. Anastassia Ailamaki of Carnegie Mellon University, and David Lomet of Microsoft Research for their efforts. I look forward to welcoming you to the ICDE 2007 Workshop on Self Managing Database Systems in Istanbul.