6.0 – Future Work

W. Morven Gentleman

Morven.Gentleman@dal.ca

This workshop has highlighted a number of take-away actions for the scientific community interested in software system dependability.

A final report describing the results of the workshop was produced and made available to others interested in the topic. The report was published in April 2008 as RTO-TR-IST-047, "Building Robust Systems with Fallible Construction". It details research that has yet to be done, but that has been identified as needed in order for systems being developed today to be resilient to predictable problems.

Existing software fault tolerance technology is inadequate because of new perspectives on what is it means to build robust systems, and what is needed for systems to be robust:

- Robustness is needed, which is a different issue from correctness.
- People are part of the system.
- Dependability requirements depend on which stakeholder is considered.
- Automated correction of failures is not always feasible or appropriate.
- Autonomic computing, i.e. self-managed systems, has a role.
- Rollback is not always feasible or desirable.
- Service availability may outweigh correctness of individual service requests.
- Software development is not a single homogeneous activity.
- The software product may not be monolithic homogeneous code.
- The development organization may not be a monolithic homogeneous entity.
- Malicious attacks may be an essential concern, beyond accidental failures.
- Fault tolerance awareness needs to be ingrained in stakeholders.

These points are elaborated in the aforementioned report. Recovery-oriented computing has also been recognized as an important shift of perspective, and although some investigations have been undertaken based it, nevertheless a great deal more is needed.

Existing software fault tolerance technology is also inadequate because new technology creates new options, but in addition poses new situations requiring additional solutions.

We have identified four technologies that have become widely available in the past few years, each of which offers potential for new solutions to building more robust systems:

1) Surfeit of computing capacity.
2) Autonomic computing.
3) Virtual machines.
4) The discipline of software architecture.

The aforementioned report elaborates on these technologies. Because these technologies were not available when most software fault tolerance technology was being developed, their application was not taken into account in the software fault tolerance literature. The potential benefits of them need to be investigated.

We have also identified about a dozen technologies that have become prominent in the past few years that present new challenges for building robust systems:

- Software component-based engineering.
- Systems of Systems.
- Web and Internet technologies.
- Concurrent, parallel, and distributed computing.
- Exception handling.
- Non-imperative programming.
- Genetic and more generally exploratory computation.
- Massive datasets.
- Inadequacy of oracles.
- Security and privacy.
- Multimedia, especially time-based streaming media.
- Scalability and nonstop operation.
- Rapid rate of new releases.

Again the aforementioned report elaborates on these technologies and issues that they raise. Although some of the existing literature on software fault tolerance bears on these issues, it has many deficiencies and gaps, meaning that intense further study is required in order to provide guidance for developing systems that involve these technologies.

In short, existing software fault tolerance literature has serious shortcomings for today's systems, and simple extension of past work will not resolve that: new directions in research must be pursued. Results are needed for systems being built now. Funding such research is critical.