Software Reliability and Safety

CSE 8317 — Spring 2005

Prof. Jeff Tian, tian@engr.smu.edu
CSE, SMU, Dallas, TX 75275
(214) 768-2861; Fax: (214) 768-3085
www.engr.smu.edu/∼tian/class/8317.05s

SRE.4: Applications & Frontiers

- Non-traditional applications.
- Data treatment for better results
- Data clustering and DCRMs
- Research issues and improvement

Current SRE Assessment

- Reliability assessment:
  - ▷ IDRMs and SRGMs (TBRMs too)
  - ▷ Key: data reflect reliability

- Reliability prediction:
  - ▷ SRGMs
  - ▷ Key: OP and data

- Reliability improvement:
  - ▷ TBRMs and other emerging approaches
  - ▷ Key: early risk identification

- More research needed.
  (Why we are working on related topics.)

SRE Applications

- Traditional applications:
  - ▷ "medium-reliable" systems.
  - ▷ Telecommunication software/systems:
    – Musa/AT&T/Lucent, and others.
  - ▷ Operation systems and system software:
    – DEC/HP/IBM/etc.
  - ▷ Commercial software systems:
    – IBM examples in this class.
  - ▷ Other similar applications.

- New application domains:
  - ▷ Adapting existing models/techniques.
  - ▷ Data selection/treatment necessary?
  - ▷ New models/techniques needed?

Web Testing and SRE

- Project background:
  - ▷ Reliability, testing and QA for user-oriented systems — NSF award CCR-0204345
  - ▷ SMU research team: Nguyen, Shahdad, Kallepalli, Li, Ma, Rudraraju, Tian.
  - ▷ Key publications: 2 TSE papers
    – (Kallepalli and Tian, 2001)
    – (Tian/Rudraraju/Li, 2004)

- Key activities and results:
  - ▷ Usage-based testing: UMMs
    – unified Markov models
    – details in Tian/SQE book: Ch.10.
  - ▷ Statistical web testing and SRE:
    – (Kallepalli and Tian, 2001)
  - ▷ Web SRE: time/activity measurement:
    – (Tian/Rudraraju/Li, 2004)

Web Characteristics and QA

- Web applications and QA:
  - ▷ Large, diverse, general population.
  - ▷ Document/information vs. computation.
  - ▷ Diverse usage patterns/environments.
  - ▷ Reliability one of the key concerns.

- Usage environment:
  - ▷ Traditional hardware/software env.
  - ▷ Network/midware/server/browser/etc.
  - ▷ Layered structure.
  - ▷ Failure analysis necessary.

- SRE and statistical testing appropriate.

Web Failure, QA, and Testing

- Web failure: inability to deliver information or document required by a web user.

- Infrastructure failure
  - ▷ Host failure: hardware/OS.
  - ▷ Network failure: down/congested.
  - ▷ Browser failure: software problem.
  - ▷ Related hardware/software problems.
  - ▷ Existing hardware/software reliability

- Information source failure
  - ▷ Individual page problems
  - ▷ Overall reliability: focus here.

- User errors: beyond our control.

Web Components and Existing Testing

- Web component (at the source):
  - ▷ HTML document.
  - ▷ Java, JavaScript, ActiveX.
  - ▷ Cgi-Bin script.
  - ▷ Database.
  - ▷ Multi-media components.

- Existing testing: SQE Ch.12.
  - ▷ Above components and the overall system.
  - ▷ Functionality testing.
  - ▷ Load/performance testing.
  - ▷ Usability testing.
  - ▷ Browser rendering.

Existing Web Testing

- Document conformance to standards:
  - ▷ Problem classification (Bowers, 1996):
    1. syntax (wrong tags/elements)
    2. lexical (wrong char set/formatting)
    3. HTML usage (opt. elements)
    4. structural integrity (dead link)
    5. portability (platform/browser)
    6. stylistic (no recommended)
  - ▷ Online tools to check them
  - ▷ Web pages in isolation
  - ▷ 2-tiered strategy: (Tian/Nguyen, 1999)

- Other functionality testing:
  - ▷ link checking
  - ▷ form testing
  - ▷ end-to-end transaction verification
  - ▷ Java component testing

Statistical Web Testing: Why?

- Factors to consider:
  - ▷ Existing: coverage-based testing
  - ▷ Web size, complexity, user focus
  - ▷ Dynamic nature
  - ▷ Focus on source failures
  - ▷ Statistical web testing
    - – modeling, testing, result analysis

- Statistical testing models:
  - ▷ Usage model: operational profile (OP)
  - ▷ Flat (or staged) OP: Musa
  - ▷ Use UMMs (unified Markov models)
  - ▷ Usage/visibility by users

- Reliability assessment/improvement too!

New Web Testing Strategy

- Hierarchical web testing:
  - ▷ Top-tier: flat (Musa) OP
    – for simplicity and skewed distribution.
  - ▷ Middle-tier: UMMs
    – importance of highly used navigations.
  - ▷ Bottom-tier: existing web testing
    – no need to re-invent wheels

- Implementation: SQE Fig 12.2 (p.218)
  - ▷ Model construction via access-log analysis.
  - ▷ TAR (top access report) ⇒ top-tier.
  - ▷ CPR (call-pair report) to form clusters ⇒ middle tier UMMs.
  - ▷ UMM refinement ⇒ bottom-tier.
  - ▷ Reliability/other analysis: error-log

Web Logs and Usage/Reliability

- Access and error logs: Information source for usage modeling and reliability analysis (Tian/Rudraraju/Li, 2004)

- Access log: hits
  - ▷ loading a HTML file
  - ▷ loading graphics etc.
  - ▷ but not operations using local cache
  - ▷ specific information recorded at server
  - ▷ sample entries: Table 1

- Error log: problems
  - ▷ sample entries: Table 2
  - ▷ problem type: Table 3
  - ▷ similar info and format

Web Logs and Usage Modeling

- Access log analysis:
  - ▷ Access frequency from different users.
  - ▷ Timing analysis of accesses.
  - ▷ Network traffic and performance.
  - ▷ Both existing tool (e.g., FastStats) and specialized programs used.
- For usage-based web testing?
  - ▷ Usage patterns and frequencies.
  - ▷ Usage model: Hierarchical web testing.
- Used in reliability analysis.

Error Logs and Reliability Analysis

- Error logs
  - ▷ Detailed problem information
  - ▷ "failures" for reliability analysis
  - ▷ In conjunction with other measurements
  - ▷ When absent: use response code.

- Reliability analysis
  - ▷ Reliability by Nelson model
  - ▷ Mean-time-between-failures

    $$\text{MTBF} = \frac{1}{f} \sum_i t_i$$

    for usage time t_i
  - ▷ MTBF when t_i not available

    $$\text{MTBF} = \frac{n}{f}$$
  - ▷ Reliability growth using SRGMs (software reliability growth models)

Case Studies

- Site, information sources, and tools
  - ▷ Site: www.seas.smu.edu
  - ▷ Information sources: access/error logs (Apache web server)
  - ▷ Analysis tools:
    - – FastStat and Perl programs
  - ▷ Manual analysis also
- Cross-validation:
  - ▷ www.kde.org.
  - ▷ Different types of web site.
  - ▷ Heavier traffic.

Case Study: Testing SMU/SEAS

- Results summarized in Tian/SQE book:
  - ▷ Hierarchical strategy: Section 12.4.
  - ▷ Top level OP: Table 8.4 (p.112) and Fig 8.1 (p.113)
  - ▷ UMMs for web testing: below. (Section 10.6)

- UMMs for SMU/SEAS
  - ▷ Overall model: Fig 10.6
  - ▷ Detailed (lower-level) models:
    – produced, but not shown
  - ▷ Hits instead of probabilities
  - ▷ Omitted low hit nodes (captured in other models)

Case Study: Error Analysis

- General error analysis result:
  - ▷ Summary: Table 4
  - ▷ Types A through K, but two key types:
    - Type A: permission denied
    - Type E: file does not exist

- Further analysis of errors
  - ▷ Type A: may or may not be considered as failures
  - ▷ Type E: "failures"
  - ▷ Further analysis of Type E errors
  - ▷ Relating to usage information
  - ▷ Reliability analysis

Case Study: Reliability Analysis

- Error over time: Fig 1.
  - ▷ Ups and downs (calendar time)
  - ▷ Impact of traffic/workload
  - ▷ Conclusion: proper workload/usage measurement for reliability analysis

- Possible workload/usage measurements:
  - ▷ Hits: already done in (Kallepalli and Tian, 2001)
  - ▷ Bytes (some difficulties)
  - ▷ Sessions ($\approx$ to data grouping)
  - ▷ Users (meaningful to service providers)

- Measurement results: Figs 3-7.

Case Study: Reliability Analysis

- Overall reliability:
  - ▷ Relating failures to usage
    – errors vs workload measurements
  - ▷ Example plot: Fig 8
    – errors vs. bytes

- Reliability evaluation results:
  - ▷ Application of Nelson model.
  - ▷ Error rate: 0.0379 error/hit.
  - ▷ MTBF = 26.6
  - ▷ Reliability = 0.962
  - ▷ Results in other units possible
    – but need to be cautious.
  - ▷ Comparison: Table 5

Case Study: Reliability Analysis

- Reliability growth for statistical testing

- Hypothetical situation:
  - ▷ usage-based testing
  - ▷ immediate defect removal
  - ▷ studied over 26 days
  - ▷ calculated from error/access logs
  - ▷ computation: unique error sequence

- SRGM results:
  - ▷ GO model for errors vs. bytes: Fig. 10
  - ▷ Reliability growth: 74.8% (defect reduction)
  - ▷ Other results: Table 6.
  - ▷ Purification level ρ (from SRE.3, and SQE Ch.22)

Case Study: Reliability Analysis

- Cross validation using KDE data.
- Overall workload and reliability:
  - ▷ Profiles: Fig 11 (4 profiles)
  - ▷ Session profile: Fig 12 – two variations (2 hr vs. 15 min)
    – only 2 hrs used for SMU/SEAS
  - ▷ Hourly traffic: Fig 13.
  - ▷ Overall results: Table 7.
  - ▷ Similar results (better reliability)
- Reliability growth:
  - ▷ GO model for errors vs. bytes: Fig. 14
  - ▷ $\rho = 86.7\%$ to 88.9%
  - ▷ consistency↑ and reliability growth↑

Web SRE (& Testing) Summary

- What has been done?
  - ▷ Characteristics of web testing
  - ▷ hierarchical web testing strategy
  - ▷ Reliability assessment/prediction by analyzing both access and error logs
  - ▷ Case study to demonstrate viability and effectiveness

- SRE specific results:
  - ▷ Data and modeling from existing sources.
  - ▷ Good operational reliability assessment.
  - ▷ Reliability growth potential assessment.
  - ▷ 2 diverse web sites ⇒ generalization.
  - ▷ Future research: – change impact, risk identification, byte traffic measurement.

Data-Model Mismatch

- Data-model mismatch:
  - ▷ Assumption mismatch.
  - ▷ Data appropriate?
    $\Rightarrow$ data selection and/or treatment
  - ▷ Model appropriate?
    $\Rightarrow$ choose alternative model
    $\Rightarrow$ develop new model
  - ▷ Research community: new models (but often impractical)
  - ▷ Industry: model/data selection/treatment
  - ▷ Examples from Module IV (& other work)

- Data treatment:
  - ▷ Censoring techniques
  - ▷ Grouping techniques

Data Treatment

- Data censoring techniques:
  - ▷ Key idea: skip gaps in data
    ⇒ censored data reflects usage
  - ▷ Technique: K.-Y. Cai, IEEE Trans. Reliability 46(1):69-75, 1997.

- Data compression:
  - ▷ Compression/expansion vs skipping
  - ▷ Basis: coverage
    – less likely to fail if tested
    – coverage as multiplier
  - ▷ Technique: M.-H. Chen et al, IEEE Trans. Reliability 50(2):165-170, 2001.

- Works with individual data points directly.

Data Grouping

- Need for data grouping:
  - ▷ Already grouped from applications:
    - – hourly/daily/weekly/monthly data
    - – data collection practicality
  - ▷ Local fluctuations
  - ▷ Data dependencies
  - ▷ Use PFC instead of TBF models

- Basis for data grouping:
  - ▷ External clock/time (most of the existing work)
  - ▷ Model (result) optimization
    - – Schneidewind approach TSE 19(11):1095-1104, 11/1995
  - ▷ Data clustering

DCRM

- General information/strategy:
  - ▷ Tian TSE 28(10):997-1007, 10/2002.
  - ▷ DCRM: DCRM1 + DCRM2
    data cluster based reliability models
  - ▷ Automatic clustering
  - ▷ DCRM1: direct usage
  - ▷ DCRM2: use with existing SRGMs
    (grouped data as input)

- Basic ideas: How?
  - ▷ Clustering of homogeneous runs.
  - ▷ Data driven/sensitive partitions.
  - ▷ Method: Tree-based modeling (TBM).

The Case for Grouping

- Scenario-based vs. random testing:
  - ▷ Parallelism/interleaving in testing.
  - ▷ Randomized workload.
  - ▷ Similar overall picture.
  - ▷ ⇒ Data grouping.

- Defect fixing and run dependencies:
  - ▷ Strong short term dependency.
  - ▷ Lack of long term dependency.
  - ▷ ⇒ Clustering.

- Develop DCRMs

DCRM Construction

- Clustering/grouping test runs:
  - ▷ By similar failure intensity.
  - ▷ Computation: Tree-based modeling (TBM) supported by S-PLUS.

- Generic procedure:
  - ▷ Identify period, runs, and time.
  - ▷ Failure intensity = failure / time,
  - ▷ Simple algebraic mean for segment:

$$\frac{\sum_{j=1}^{n_i} f_{ij}}{n_i} = \frac{f_i}{n_i} = \lambda_i$$

  - ▷ Weighted average for segment:

$$\frac{\sum_{j,l_i<d_j\le u_i} t_j \lambda_j}{\sum_{j,l_i<d_j\le u_i} t_j} = \frac{\sum_{j,l_i<d_j\le u_i} f_j}{\sum_{j,l_i<d_j\le u_i} t_j} = \frac{F_i}{T_i} = \Lambda_i$$

  - ▷ Other: as special cases of above.

Model Usage and Performance

- Direct usage: DCRM1
  - ▷ Reliability for each segment.
  - ▷ Overall trend assessment.
  - ▷ Current reliability: last segment.
  - ▷ Prediction: extrapolation.
  - ▷ Risk/anomaly identification.

- DCRM1 performance:
  - ▷ Goodness-of-fit: R^2
    - 304 vs. 6329 for Goel-Okumoto.
  - ▷ Prediction comparison:
    - use training and testing sets.
    - linear extrapolation.
    - good short term results
  - ▷ Key advantage: early/wide applicability

DCRMs vs Other Models

- DCRM1 vs IDRMs:
  - ▷ Partition by failure intensity in runs
    – similar to Nelson model.
  - ▷ Partition by general failure intensity
    – similar to Brown-Lipow model.

- DCRM1 vs SRGMs:
  - ▷ Constant λ for given period
    – similar to Jelinski-Moranda model.
  - ▷ But variable steps in consecutive steps
    – similar to Littlewood-Verrall model.
  - ▷ Non-function form for progression of λ's

- PFC-SRGMs: used in DCRM2.

DCRM2

- DCRM2: SRGMs with grouped data (each segment as a data point)

- Choosing SRGMs for DCRM2:
  - ▷ Only PFC (FC) models usage
  - ▷ NHPP choices:
    - Goel-Okumoto (GO)
    - Musa-Okumoto (MO, log Poisson)
    - Schneidewind and data req.
    - S-shaped as descriptive model
  - ▷ GO and MO choices

- More about GO and MO choices:
  - ▷ Lower/upper bound on estimates
  - ▷ Past experience at IBM
  - ▷ Empirical data elsewhere

DCRM Performance

- Product and comparison points:
  - ▷ Products E (and D) from IBM
  - ▷ E: last 8 weeks
    - – 7 point comparison for DCRM2
  - ▷ DCRM1,
    GO, MO, DCRM2.GO, DCRM2.MO

- Applicability:
  - ▷ DCRM1 clearly superior
  - ▷ Others about equal

- Goodness-of-fit:
  - ▷ DCRM1 clearly superior
  - ▷ Others about equal
  - ▷ Caution: use more important

DCRM Performance

- Reliability assessment:
  - ▷ DCRM1 not as stable but available early
  - ▷ Convergence of others
  - ▷ DCRM2 provide tighter bound
    (more stable also, see prediction)

- Reliability prediction:
  - ▷ DCRM1 only for short term
    (mixed results)
    – only one available early
  - ▷ Prediction accuracy tables
  - ▷ Direct comparison graphs
  - ▷ Conclusion: DCRM2 better

- Model stability: DCRM2 better

DCRM Summary

- Easily satisfiable assumptions:
  - ▷ Rough operational profiles.
  - ▷ No long term dependencies
    – but short term dependencies
  - ▷ Failure intensity clusters.

- Implementation and applications:
  - ▷ Model construction: S-PLUS.
  - ▷ Practical applications.
  - ▷ Better/wider applicability.
  - ▷ Robust/stable results.
  - ▷ Further studies underway.

Other Data/Models

- Trend analysis:
  - ▷ Qualitative/visual inspection:
    – curvature (super-/sub-additive)
  - ▷ Quantitative analysis.
  - ▷ Avoid meaningless modeling results.

- Other data/models:
  - ▷ Reliability simulation:
    especially for mixed h/w-s/w systems
  - ▷ Composite models.
  - ▷ Fault seeding technique.
  - ▷ General models for correlated data:
    Goceva-Popstojanova and Trivedi, IEEE Trans. Reliability, 49(1):37-48, 3/2000.
  - ▷ etc., Lyu book.

Improvement Strategies

- Traditional models/techniques:
  - ▷ Assessment/prediction focus.
  - ▷ Limited used in improvement.
  - ▷ Testing/QA as semi-separate.
  - ▷ (assessment in Module IV)
- TBRM and extensions:
  - ▷ TBRMs for risk identification.
  - ▷ Focused improvement during testing.
  - ▷ Extension to other phases:
    - – analyze other (e.g., inspection) data
    - – reliability composition

Other Research Topics

- Linking SRE with metrics/analysis.
  - ▷ Musa prescriptive models refinement.
  - ▷ Metrics-SRE: still separate
    - – risk (problem-prone) identification
  - ▷ Quantitative linkage?

- Reliability composition:
  - ▷ Small scale: Hamlet/Mason/Woit work.
  - ▷ RE to requirement: Smidts work.
  - ▷ Work at SMU: OP-mapping, fault-injection, embedded systems.

- Reliability optimization.
  - ▷ Other factors: cost, schedule, etc.
  - ▷ Lyu/Rangarajan/van Moorsel work.