Resilience: an Essential Property for the Sustainability of Computing Systems and Infrastructures
— From Dependability to Resilience —

Jean-Claude Laprie

ReSIST Summer School
Resilience in Computing Systems and Information Infrastructures
— from Concepts to Practice —
24th-28th September 2007, Porquerolles

❖ Dependability

➢ Basic concepts

➢ State-of-the-art from statistics

❖ Resilience

➢ Definition and technologies
continuously evolving (complex) systems
**Dependability**: ability to deliver service that can justifiably be trusted

- Service delivered by a system: its behavior as it is perceived by its user(s)
- User: another system that interacts with the former
- Function of a system: what the system is intended to do
- (Functional) Specification: description of the system function
- Correct service: when the delivered service implements the system function
- (Service) Failure: event that occurs when the delivered service deviates from correct service, either because the system does not comply with the specification, or because the specification did not adequately describe its function
- Failure modes: the ways in which a system can fail, ranked according to failure severities
- Part of system state that may cause a subsequent service failure: error
- Adjudged or hypothesized cause of an error: fault

**Dependability**: ability to avoid failures that are unacceptably frequent or severe

- Failures are more frequent or more severe than acceptable: dependability failure

---

**Dependability**

- Readiness for usage
- Continuity of service
- Absence of catastrophic consequences on the user(s) and the environment
- Absence of unauthorized disclosure of information
- Absence of improper system alterations
- Ability to undergo repairs and evolutions

- Availability
- Reliability
- Safety
- Confidentiality
- Integrity
- Maintainability

**Security**

Absence of unauthorized access to, or handling of, system state
Dependability attributes

- Availability, Reliability, Safety, Confidentiality, Integrity, Maintainability: Primary attributes
- Secondary attributes
  - Specialization
    - Robustness: dependability with respect to external faults
    - Survivability: dependability in the presence of active fault(s)
  - Distinguishing among various types of (meta-)information
    - Accountability: availability and integrity of the person who performed an operation
    - Authenticity: integrity of a message content and origin, and possibly some other information, such as the time of emission
    - Non-repudiability: availability and integrity of the identity of the sender of a message (non-repudiation of the origin), or of the receiver (non-repudiation of reception)
Dependability Threats

Internal fault, dormant vulnerability → Error → Error → Error

Activation (computation process) → propagation → prop.

Correct Service → Failure → Incorrect Service: Outage

External fault occurrence → Internal fault, dormant vulnerability

Causation, Propagation → Correct Service → Failure → Incorrect Service: Outage

Service Interface
Means for Dependability

- Preventing occurrence or introduction of faults
- Delivering correct service in spite of faults
- Reducing the presence of faults
- Estimating the present number, the future incidence and the likely consequences of faults

**Fault Prevention**

**Fault Tolerance**

**Fault Removal**

**Fault Forecasting**

**Dependability definitions**

- **Original definition:** ability to deliver service that can justifiably be trusted
  - Enables to generalize availability, reliability, safety, confidentiality, integrity, maintainability, that are then attributes of dependability

- **Alternate definition:** ability to avoid service failures that are unacceptably frequent or severe
  - A system can, and usually does, fail. Is it however still dependable? When does it become undependable?

  criterion for deciding whether or not, in spite of service failures, a system is still to be regarded as dependable

- **Dependence** of system A on system B is the extent to which system A’s dependability is (or would be) affected by that of system B

- **Trust:** accepted dependence
  - Explicitly
  - Implicitly
## Dependability and similar notions

<table>
<thead>
<tr>
<th>Concept</th>
<th>Dependability</th>
<th>High Confidence</th>
<th>Survivability</th>
<th>Trustworthiness</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Goal</strong></td>
<td>1) ability to deliver service that can justifiably be trusted</td>
<td>consequences of the system behavior are well understood and predictable</td>
<td>capability of a system to fulfill its mission in a timely manner</td>
<td>assurance that a system will perform as expected</td>
</tr>
<tr>
<td></td>
<td>2) ability of a system to avoid service failures that are unacceptably frequent or severe</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Threats present</strong></td>
<td>1) development faults (e.g., software flaws, hardware errata, malicious logic)</td>
<td>• internal and external threats</td>
<td>1) attacks (e.g., intrusions, probes, denials of service)</td>
<td>1) hostile attacks (from hackers or insiders)</td>
</tr>
<tr>
<td></td>
<td>2) physical faults (e.g., production defects, physical deterioration)</td>
<td>• naturally occurring hazards and malicious attacks from a sophisticated and well-funded adversary</td>
<td>2) failures (internally generated events due to, e.g., software design errors, hardware degradation, human errors, corrupted data)</td>
<td>2) environmental disruptions (accidental disruptions, either man-made or natural)</td>
</tr>
<tr>
<td></td>
<td>3) interaction faults (e.g., physical interference, input mistakes, attacks, including viruses, worms, intrusions)</td>
<td></td>
<td>3) accidents (externally generated events such as natural disasters)</td>
<td>3) human and operator errors (e.g., software flaws, mistakes by human operators)</td>
</tr>
</tbody>
</table>

### Dependability

- **Attributes**
  - Availability
  - Reliability
  - Safety
  - Confidentiality
  - Integrity
  - Maintainability

- **Means**
  - Fault Prevention
  - Fault Tolerance
  - Fault Removal
  - Fault Forecasting

- **Threats**
  - Faults
  - Errors
  - Failures

### Security
Faults: Solid faults, Elusive faults, Transient faults

Errors: Error reproducibility

Facility for stopping recursion

Interaction faults

Prior presence of a vulnerability: Internal fault that enables an external fault to harm the system

Solid (hard) faults

Elusive (soft) faults

Permanent faults (development, physical, interaction)

Transient faults (physical, interaction)

Elusive faults

Intermittent faults

June 1980: False alerts at the North American Air Defense (NORAD)

April 1981: First launch of the Space Shuttle postponed


August 1986 - 1987: The "wily hacker" penetrates several tens of sensitive computing facilities

November 1988: Internet worm

15 January 1990: 9 hours outage of the long-distance phone in the USA

February 1991: Scud missed by a Patriot (Dhahran, Gulf War)

November 1992: Crash of the communication system of the London ambulance service

26 and 27 June 1993: Authorization denial of credit card operations in France

4 June 1996: Failure of Ariane 5 maiden flight

13 April 1998: Crash of the AT&T data network

February 2000: Distributed denials of service on large Web sites

May 2000: Virus I love you

July 2001: Worm Code Red

August 2003: Propagation of the electricity blackout in the USA and Canada

October 2006: 83,000 e-mail addresses, credit card info, banking transaction files stolen in UK
- **Average outage costs**

<table>
<thead>
<tr>
<th>Industry sector</th>
<th>Energy</th>
<th>Manufacturing</th>
<th>Financial institutions</th>
<th>Insurance</th>
<th>Retail</th>
<th>Banking</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2.8</td>
<td>1.6</td>
<td>1.4</td>
<td>1.2</td>
<td>1.1</td>
<td>1</td>
</tr>
</tbody>
</table>

- **Yearly cost of failures**

<table>
<thead>
<tr>
<th>Estimates of insurance companies (2000)</th>
<th>France (private sector)</th>
<th>USA</th>
<th>UK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accidental faults</td>
<td>0.9 G€</td>
<td>4 G$</td>
<td>1,25 G€</td>
</tr>
<tr>
<td>Malicious faults</td>
<td>1 G€</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Global estimate | USA : 80 G$ | UE : 60 G€ |

- **Maintenance costs**

Space shuttle on-board software: 100 M $ / an

- **Cost of software project cancellation (failure of the development process)**

<table>
<thead>
<tr>
<th>USA [Standish Group, 2002, 13522 projects]</th>
<th>Success</th>
<th>Challenged</th>
<th>Cancelled</th>
</tr>
</thead>
<tbody>
<tr>
<td>Estimate 1983</td>
<td>34%</td>
<td>51%</td>
<td>15%</td>
</tr>
<tr>
<td>Estimate 1988 (contract passed)</td>
<td>4 G$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Estimate 1994</td>
<td>7 G$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Timing shift (estimate 1994) | 6 - 8 yrs |

- **Phone systems**

From J. Gray, 'Dependability in the Internet era'

- **Complexity**

- **Economic pressure**

<table>
<thead>
<tr>
<th>Availability</th>
<th>Outage duration/yr</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.999999</td>
<td>32s</td>
</tr>
<tr>
<td>0.99999</td>
<td>5mn 15s</td>
</tr>
<tr>
<td>0.9999</td>
<td>52mn 34s</td>
</tr>
<tr>
<td>0.999</td>
<td>8h 46mn</td>
</tr>
<tr>
<td>0.99</td>
<td>3d 16h</td>
</tr>
<tr>
<td>0.9</td>
<td>36d 12h</td>
</tr>
</tbody>
</table>
**Website uptime statistics (Netcraft)**

**Top 50 most requested sites (July 2006)**

- MTBF (yrs)
  - Maintenance
  - Environment
  - Hardware
  - Operations
  - Software
  - Total

**Top 50 longest running sites (July 2006)**

- Evolution over time

<table>
<thead>
<tr>
<th>MTTR</th>
<th>Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 min</td>
<td>99.98</td>
</tr>
<tr>
<td>10 mins</td>
<td>99.83</td>
</tr>
<tr>
<td>1 hour</td>
<td>99.01</td>
</tr>
<tr>
<td>8 hours</td>
<td>98.59</td>
</tr>
</tbody>
</table>
## Three large websites

[from D. Oppenheimer, A. Ganapathi, D.A. Patterson, ‘Why do Internet services fail, and what can be done about it?’, USISTS ‘03]

<table>
<thead>
<tr>
<th>Service characteristic</th>
<th>Website</th>
<th>Online (mature)</th>
<th>Readmosty (mature)</th>
<th>Content (bleeding edge)</th>
</tr>
</thead>
<tbody>
<tr>
<td>hits per day</td>
<td></td>
<td>~100 million</td>
<td>~100 million</td>
<td>~7 million</td>
</tr>
<tr>
<td># of machines</td>
<td></td>
<td>~500, 2 sites</td>
<td>&gt;2000, 4 sites</td>
<td>~500, ~15 sites</td>
</tr>
<tr>
<td>Front-end node</td>
<td></td>
<td>Solaris on SPARC and x86</td>
<td>Open-source OS on x86</td>
<td>Open-source OS on x86</td>
</tr>
<tr>
<td>Back-end node</td>
<td></td>
<td>Network Appliance filters</td>
<td>Open-source OS on x86</td>
<td>Open-source OS on x86</td>
</tr>
<tr>
<td>period of data stud.</td>
<td></td>
<td>7 months</td>
<td>6 months</td>
<td>3 months</td>
</tr>
<tr>
<td>Component failures</td>
<td></td>
<td>296</td>
<td>N/A</td>
<td>205</td>
</tr>
<tr>
<td>Service failures</td>
<td></td>
<td>40</td>
<td>21</td>
<td>56</td>
</tr>
<tr>
<td>MTTF</td>
<td></td>
<td>126 hours</td>
<td>206 hours</td>
<td>39 hours</td>
</tr>
</tbody>
</table>

| Service failure cause by location | Front-end | 77% | 0% | 66% |
|                                  | Back-end  | 3%  | 10%| 11% |
|                                  | Network   | 18% | 81%| 18% |
|                                  | Unknown   | 2%  | 9% | 4% |

| Average TTR by part of service (hrs) | Front-end | 9.4 (16 serv. fai.) | N/A | 2.5 (10 serv. fai.) |
|                                      | Back-end  | 7.3 (5 serv. fai.) | 0.2 (1 serv. fai.) | 14 (3 serv. fai.) |
|                                      | Network   | 7.8 (4 serv. fai.) | 1.2 (16 serv. fai.) | 1.2 (2 serv. fai.) |

| Average availability | 93.5% | 97.2% | 97.8% |

### Component failure to service failure

#### Online

![Component failure to service failure graph for Online](image)

#### Content

![Component failure to service failure graph for Content](image)
Malicious faults

SEI/CERT Statistics: vulnerabilities reported

Slammer/Sapphire worm


The fastest computer worm in history. As it began spreading throughout the Internet, it doubled in size every 8.5 seconds. It infected more than 90 percent of vulnerable hosts within 10 minutes. The worm began to infect hosts slightly before 05:30 UTC on Saturday, January 25, 2003. Sapphire exploited a buffer overflow vulnerability in computers on the Internet running Microsoft's SQL Server or MSDE 2000 (Microsoft SQL Server Desktop Engine). This weakness in an underlying indexing service was discovered in July 2002; Microsoft released a patch for the vulnerability before it was announced. The worm infected at least 75,000 hosts, perhaps considerably more, and caused network outages and such unforeseen consequences as canceled airline flights, interference with elections, and ATM failures.

The geographic spread of Sapphire in the 30 minutes after release. The diameter of each circle is a function of the logarithm of the number of infected machines, so large circles visually underrepresent the number of infected cases in order to minimize overlap with adjacent locations.
Global Information Security Survey 2004 — Ernst & Young

Loss of availability: Top ten incidents

Percentage of respondents that indicated the following incidents resulted in an unexpected or unscheduled outage of their critical business

- Hardware failures
- Major virus, Trojan horse, or Internet worms
- Telecommunications failure
- Software failure
- Third party failure, e.g., service provider
- System capacity issues
- Operational errors, e.g., wrong software loaded
- Infrastructure failure, e.g., fire, blackout
- Former or current employee misconduct
- Distributed Denial of Service (DDoS) attacks


Occurrences

Non malicious causes 71%
Malicious causes 29%

Risk perception

Non malicious causes 29%
About Resilience

- Social psychology
  - Elasticity, spirit, resource, good mood
- Material science
  - Robustness and elasticity
- Child psychiatry and psychology
  - Living, developing successfully when facing adversity
- Ecology
  - Moving from a stability domain to another one
- Business
  - Capacity to reinvent a business model before circumstances force to
- Industrial safety
  - Anticipating risk changes before damage occurrence

Adaptation to evolutionary changes, and getting back after a setback
Fault and evolution tolerance


- «resilience determines the persistence of relationships within a system and is a measure of the ability of these systems to absorb changes of state variables, driving variables, and parameters, and still persist»

- Relationship between resilience and stability in open systems
  - «a system can be very resilient and still fluctuate greatly, i.e., have low stability »
  - «low stability seems to introduce high resilience»

- diversity pointed out as of significant influence on both stability (decreasing it, as it may create several stability domains) and resilience (increasing it)
In computing systems

- Resilient
  - In use for 30+ years
  - Recently, escalating use → buzzword
  - Used essentially as synonym to fault tolerant
    "A resilient computing system is capable of providing dependable service to its users over a wide range of potentially adverse circumstances. The two key attributes here are dependability and robustness. [...] A computing system can be said to be robust if it retains its ability to deliver service in conditions which are beyond its normal domain of operation, whether due to harsh treatment, or unreasonable service requests, or misoperation, or the impact of faults, or lack of maintenance »

- Fault-tolerant computing systems are known for exhibiting some robustness with respect to fault and error handling, in the above sense, i.e., for situations exceeding their specification, e.g.:
  - Tolerance of elusive software faults thanks to loosely-coupled architectures in Tandem systems
  - Tolerance errors that escaped detection and thus did not trigger recovery in Delta-4

- This of course should not lead to forget that, contrariwise, total coverage with respect to specified faults is hardly achievable

Moving to ubiquitous systems

Large, networked, evolving systems constituting complex information infrastructures — perhaps involving everything from super-computers and huge server farms to myriads of small mobile computers and tiny embedded devices

**At stake:** maintain dependability in spite of continuous evolutionary changes

functional environmental technological

Examples of changes:
- Dynamically changing systems, e.g., spontaneous, or ‘ad-hoc’, networks of mobile nodes and sensors
- Growth of systems as demand increases
- Interactions between systems of differing natures, e.g., large-scale information infrastructure on the one hand and networks of sensors on the other
- Merging of systems, e.g., in company acquisitions, or coupling of systems, e.g., in military coalitions
- Ever-evolving and growing problem of attacks both by amateur hackers and by professional criminals
Definition of resilience for computing systems and information infrastructures

The persistence of dependability when facing evolutionary changes

The persistence of the ability to deliver service that can justifiably be trusted, when facing evolutionary changes

- Functional
  - Foreseen, e.g., new versioning
  - Foreseeable, e.g., advent of new hardware platforms
- Environmental
  - Unforeseen, e.g., drastic changes in service requests or new types of threats
- Technological
  - Short term, e.g., seconds to hours, as in dynamicity or mobility
  - Medium term, e.g., hours to months, as in new versioning or reconfigurations
  - Long term, e.g., months to years, as in reorganisations

The definition does not exclude the possibility of failure

Alternate definition of dependability

Ability to avoid service failures that are unacceptably frequent or severe

- Especially relevant in the context of evolutionary changes, as the changes can be directly a source of failure
- Incompatibilities between the formerly existing systems and the augmentations performed
Technologies for resilience

Evolutionary changes → **Evolvability**
- **Adaptation**

Trusted service → **Assessability**
- Verification and evaluation

Ubiquitous systems → **Usability**
- Human and system users

Complex systems → **Diversity**
- Taking advantage of existing diversity for avoiding single points of failure, and augmenting diversity

Means for dependability
- Fault Prevention
- Fault Tolerance
- Fault Removal
- Fault Forecasting

Evolvability  Assessability  Usability  Diversity