Assignment 1

CSED 601 Dependable Computing

Submitted by: Sajan Maharjan, 20182095, [thesajan@postech.ac.kr](mailto:thesajan@postech.ac.kr)

**Summary of Basic Concepts and Taxonomy of Dependable and Secure Computing by Algirdas Avizienis et al.**

**Introduction:**

This paper introduces various terminologies and concepts relating to dependability and security of computing systems. Defining the dependability and security of system is complicated since failure in a system is imminent. Dependability encompasses the attributes of reliability, availability, safety, integrity and maintainability. Security on the other hand encompasses the attributes of confidentiality, availability and integrity. The authors hope to establish a consensus on the concepts regarding dependability and security that would facilitate use for organizational bodies and educational purposes.

**Literature Review:**

The coinage of the word “Dependable Computing” came into existence as early as 1980. The notion of “security” only came into existence during 1992. By then, the concept of security was pretty novel and was only beginning to be recognized as a composite attribute of confidentiality, integrity and availability. This paper aims to expand, refine and simplify the taxonomy of dependable and secure computing. The authors also expect that in the near future, the taxonomy of dependable and secure computing will be widened that will involve discussions on security and issues of trust and risk management.

**Basic Concepts:**

A system is defined as an entity that interacts with any other entity. Examples of entity include- hardware, software, humans and the physical world. Computing systems are characterized by their functionality, performance, dependability, security, usability, manageability and adaptability. The function(s) of a system, defined by its functional specification is what the system is intended to do. Behaviour of a system, described by a sequence of states, is what the system does to implement those functions. Structure of a system is what enables the system to generate its behaviour. Service delivered by a system is its behaviour as perceived by its users. The part of the system provider’s boundary where the service delivery takes place is the service interface. External state is the state that is perceived by or is visible to the system users at the interface, while the internal state is the state within the system. In a multi-component system, a single system could act both as a service provider and service user. Correct service is the delivery of intended system functions to the users. A service failure is an event when the delivered service deviates from the correct service. The period of absence of service or delivery of incorrect service is called service outage. Service restoration is the act of transitioning the system from incorrect service to correct service. The deviation of a system’s service from its intended service is called error. The cause of error is a fault. Faults in a system may also exist as vulnerabilities that will cause an error and ultimately leads to failure. A fault causing an error is termed as active else it resides in a system as a dormant fault.

Dependability is the ability to deliver service that can justifiably be trusted. It is also alternately defined as the ability to avoid service failures that are more frequent and more severe than is acceptable. Specifically, dependability encompasses the following attributes- availability (readiness for service), reliability (continuity of correct service), safety (absence of catastrophic sequences on the users and environment), integrity (absence of improper system alterations) and manageability (ability to undergo changes and modifications). Security encompasses the attributes- availability, integrity and confidentiality (absence of unauthorized disclosure of information). Dependability and security can be achieved by- fault prevention (preventing the occurrence of faults), fault tolerance (avoiding service failure in presence of faults), fault removal (reducing the no. of faults) and fault forecasting (estimating the likelihood of fault occurrence).

Faults in a system maybe introduced both during development and use phases. Development faults arise due to incompetence of developers and can be realized during system testing. Fault during usage require a service shutdown (intentional halt of system) or certain maintenance actions (repairs and modifications). Maintenance actions usually involves participation of external agent. Faults can also be classified on the basis of occurrence (development or operational/usage), system boundaries (internal or external), phenomenological causes (human-made or natural), dimension (software or hardware), objective (malicious or non-malicious), intent (deliberate or non-deliberate), capability (accidental or incompetence), etc. A fault that has occurred in the system can overlap with different types of faults as the faults are not exclusive. Knowledge of all possible fault classes allows the user to decide which classes should be included in a dependability and security specification.

The occurrence of an event leading to the deviation of a delivered service from a correct service is called a service failure. The different ways in which the deviation is manifested is the system’s service failure modes. There are cases in which a system’s functional specification cannot adequately describe the system’s function. In such cases failures cannot be estimated and only recognized after its occurrence. So failures can be subjective and disputable i.e. may require judgement to identify and characterize. Service failure modes incorporate- failure domain (content failure and/or timing failure), detectability (signalled or unsignalled), consistency (consistent or inconsistent) and consequences/severity (minor or catastrophic). The severity of failure can be determined by the factors- outage duration, possibility of human lives endangered, information unduly disclosed and ability to recover after being compromised. Systems can also be categorized in terms of failures- fail-controlled system (failures allowed only in specific modes and to an acceptable extent), fail-halt system (failure results in halt of system), fail-silent system (no response to failure), fail-safe system (failures are limited to minor failures). Development faults could lead to development failures which leads to assigned funds being overrun or projected delivery schedule lags behind. Some development failures come in the form of downgrading whereby a lower-grade, less functional system is delivered to the user. A dependability or security failure is when failures occur more frequently and severely than what is acceptable.

Error is the deviation of the system’s delivered service from the intended service. Faults in a system is the cause of errors. Errors can be detected if signalled or lay undetected- called latent errors. Whether or not an error will lead to a failure is determined by- structure of the system (presence of protective redundancy) and behaviour of the system (removal of unused parts that could lead to failure, implementation of error correcting codes). Errors maybe classified on the basis of consistency, detectability, severity as like failures mentioned above.

There is a subtlety between faults, errors and failures and their inter-relationship with one another. Dormant faults become active in the system by exploiting vulnerabilities or some computation process or environmental conditions. The activation of faults results in an error. The error propagates between the components of the system until it reaches the system component connected to the user interface. Due to this, the interface is unable to perform its desired service to the user and then results in service failure. The connection between faults, errors and failures can be understood as causal relationship. Also, in a system where there are multiple components whereby service provided by one component acts as input to another component. The faults in one of the compoents will cause a failure in one of the system interface, whose inability to assume its service will cause a fault in order component and thus the chain of relationship will continue and could cause failure in multiple components.

Faults are reproducible if we are able to identify the pattern of fault activation. Such faults that are reproducible following a certain activation patterns are called solid or hard faults. While those that don’t have fixed/systematic reproducible pattern is called elusive or soft faults. Threats unlike faults, errors or failures retains the notion of potentiality to disrupt a system and is commonly viewed through security aspect of the system.

Traditionally, dependability has been defined around the term availability. However, recent definitions of dependability, involves the notion of reliance and ability to continue its services despite service failures. Also security is not an aspect of dependability but in itself is a composite of attributes- confidentiality, integrity and availability. The extent of dependability and security attributes of a system must be considered in a relative, probabilistic sense and not in an absolute, deterministic sense. Dependability of a system can also be accounted through robustness which characterizes how a system reacts to a specific class of faults. Security can also be defined in terms of secondary attributes such as- accountability (availability and integrity of the identity of the person who performed an operation), authenticity (integrity of message content) and non-repudiability (availability and integrity of the identity of the sender of the message). A security policy is a set of security motivated constraints that are to be adhered by an organization or a computer system which if unadhered could result in security failure. High confidence, survivability and trustworthiness are other terms similar to dependability that have similar goals and address similar threats.

Dependability and security can be attained through the mechanism of fault prevention, fault tolerance, fault removal and fault forecasting. Fault prevention tries to prevent fault occurrence in a system (which seems impossible), although it is the obvious aim for development methodologies in both software and hardware. Fault tolerance aims at failure avoidance, carried out through error detection and system recovery. Error handling (rollback, roll-forward and compensation) and fault handling (diagnosis, isolation, reconfiguration, re-initialization) together form system recovery. The process of error detection, error handling and fault handling depends upon the underlying fault assumptions of the system. Not all fault tolerant techniques are equally effective. The measure of effectiveness of any fault tolerant technique is called its coverage. While co-ordinating activities in a system consisting of multiple components, it is necessary to ensure the prevention of error propagation from affecting the operation of non-failed components. Non failed components must reach an agreement as to how the information they obtain should be employed in a mutually consistent way. This is known as the consensus problem. A fault tolerant system is also synonymously addressed as self-repair, self-healing or resilient system. Another means of attaining dependability and security is through the mechanism of fault removal. Fault removal during use phase achieved through corrective (removing faults that produced one or more errors) or preventive maintenance (discovery and removal of faults before occurrence). Fault removal during development phase of a system life cycle consists of- verification, diagnosis and correction. Verification is the process of checking whether the system adheres to given terms and properties. If the system does not meet its verification conditions, the process of diagnosis takes place, which is followed by necessary corrective actions. Once the corrective actions have taken place, there is a need to perform verification once again to ensure that the fault removal process did not lead to undesired consequences. Such verifications are usually called regression testing/verification, which is the measure of the change introduced in the overall system when a small change is introduced. The verification process can be either static (doesn’t require the execution of system) or dynamic (requires executing the system). Also during verification process, faults are introduced to the system as a part of the test patterns, which is referred as fault injection. Fault forecasting is the measure of the likelihood of fault occurrence in the future by carrying out an evaluation of the system behaviour with respect to fault occurrence or activation. Evaluation can be qualitative (identify, classify and rank the failure modes) or quantitative (numerical, measured in terms of probabilities). Two of the commonly used probabilistic forecasting models are modeling and testing. These mentioned means of attaining dependability and security in the system through fault prevention, fault removal, fault tolerance and fault forecasting work better when one or more means are combined and utilized together and provide more confidence in the delivery of service that can be trusted.

**Conclusion:**

Today, computing systems are being used in a wide range of domain- from recording transactions in a local convenience store to airspace shuttle launch. All these computing systems demand high dependability and security. Dependability and security include properties such as availability, reliability, safety, confidentiality, integrity and maintainability. Understanding of fault-error-failure model is vital to understand the various threats that may affect a system.