completely defined and optimized, the result of which will be efficient, accurate, complete and consistent.

[00036] The disclosed embodiments describe the safety development in a computing system consisting of the reliability development, availability development and quality control. The goal of safety development is to prevent the system and software from impacting the safety even if there is the presence of development mistake (human error) or non-foreseeable dynamic error, such as devices' defects, external interferences.

[00037] Reliability: it means that a software acts as implemented. Taking the famous "Hello World" software code as an example that is implemented to output the sentence of "Hello World", if the code can always output the sentence: "Hello World", then it can be said that the code is reliable because it does what is implemented. If there is a typo or mistake in the programming that wrote "World" as "Word", which results in that the software code will output: "Hello Word", and if the code can always output the sentence: "Hello Word", then it can still be said that the code is reliable because it does what is implemented, as well.

[00038] The developing reliability is defined as that both said data systematic attributes of each output data are matched between their implementations and their definitions, consisting of:

(1). detecting errors consisting only of data value error and data timing error, consisting of:

a). data value error detections, consisting of:

1). input data value errors detections consisting of checking each said input data implemented value from said input device against said input data transmission protocol; and

2). middle data value errors detections consisting of checking each said middle data implemented value from said memory devices against said middle data defined value; and

3). output data value errors detections consisting of checking each said output data implemented value against said output data defined value; and

b). data timing errors detections, consisting of:

1). input data timing errors detections consisting of checking each said input data available time from said input device against said input data transmission protocol; and

2). middle data timing errors detections consisting of checking each said middle data available time from said memory device against defined middle data available time; and

3). output data timing errors detections consisting of checking each said output data implemented timing attribute against said output data defined time attribute; and

(2). errors reactions, consisting of:

a). stopping output of said output data if there is any error detected from said output data, or from middle data or input data that are used in said output data's calculation; and

b). informing said output data's users which are receivers of said output data outside said system about said detected errors; and

c). recording said detected errors and said errors causes consisting only of:

1) device defects;

2) interferences;

3) development mistakes; and

d). recovering from said errors if said errors do not exist anymore.

The detailed contents above are specified by ISO 26262, and the specific implementations will depend on the specific errors detected.

[00039] The system error detection is important for the reliability, which can be fully covered by two types of error detections: data value error detection and data timing error detection, which further can be fully covered by three types of data error detections: input data error detection, middle data error detection and output data error detection, in which the data are defined in the system operation concept.

[00040] Input data are input from outside of said computer using the input transmission protocol which includes the data timing checking and data integrity checking information, so the input data error detections can be done by the following:

a. Data Value Error detection: since the input data are from outside, so only the data transmission errors need to be detected, which can be done by checking the data integrity information, such as the CRC or checksum values that are embedded in the input data based on the input transmission protocol.

b. Data Tim Error detection: the timing of said input data is defined by the data transmission protocol, so the input data arriving time at the input port will be check by following the input transmission protocol to detect the timing errors.

[00041] Middle data and output data error detection:

a. Data Time Error: this type error cannot be detected exactly by the system under development, because all the data in a derivation and the instructions that operate said data in a computer will be processed by the Arithmetic Logic Unit (ALU) of said computer in serial, so they don’t have the common time as a reference to detect the time error directly. However, the system under development can check each said middle data available time against the designed

time and check each said output data derivation time against the designed time to make sure that said middle data available time and said output data derivation time meet the designed times using measures, such as timeout monitoring, internal or external watchdog, window watchdog, carefully designing the schedulers and task arrangements.

b. Data Value Error: this type error consists of data transmission error and data transforming error, wherein the data transmission error is detected and corrected by the computer's built-in mechanism, such as Error Correcting Code (ECC), so in most cases, the system under development does not need to detect said transmission errors, however, if the system needs to run the safety relevant software on the devices whose safety levels are lower than said software, then the system must ensure that said devices meet the safety requirements by explicitly running the data transmission error detection software routines. Data transforming error cannot be detected directly because the data should be changed by the operations defined in the calculation formulas in a computer, however, the system under development can check each said middle data value against the designed value; and check each said output data derivation value against the result plausibility using measures such as result value range checks, comparing the result values with the experience model values, comparing the result values with the simulation values, comparing the result values with the redundant storage values or redundant calculation result values.

[00042] Availability Development

[00043] Availability: the availability is a software's ability to provide the required functionalities even when something goes wrong in the system, which requires that the software should have the redundant mechanism for certain important functionalities. For example, to detect the objects on the road, the autonomous driving vehicle will have at least two redundant

mechanisms for such object detection, one uses the radar, another uses the camera, and they are independent each other, so that the object detection ability is increased in cases where either the radar or the camera is out of order.

[00044] Another example is the braking system in a vehicle, which consists of two sub-systems: the Electronic Control Braking System (ECBS) that is the main brake system and the Electronic Parking Braking System (EPBS) that is the backup brake system, i.e., in the case where the main braking system: Electronic Control Braking System (ECBS) goes wrong, then the Electronic Parking Braking System (EPBS) can be used to decelerate the vehicle.

[00045] The availability can be enhanced by the recovery from the fault states, which requires that the software should recover from the faults in the manner that the system functionality is still acceptable while the safety is not impacted, though it will not meet the system functional availability requirements in some cases, especially in the real time systems.

[00046] Quality Control

[00047] Quality control: the goals of quality control in the development are to make sure that the development designs what are required and implements what are designed. In another words, quality control is to prevent the development from deviations between what are implemented and what are required that are caused by any failure cause and especially development mistakes, which can be done from both the technical aspect and the management aspect consisting of:

(1). verifying if said definitions of said input data and said output data meet said system's requirements; and

(2). verifying if said constructions of both systematic attributes of all said output data satisfy said output data's definitions which are derived from said system's requirements; and

(3). verifying if executions of both systematic attributes of all said output data meet said output data calculation definitions.

[00048] How to develop the accurate and qualified required software is the technical aspect, such as executing the system integration test, system black box verification. How to avoid making mistakes in the development is the management aspect, for which the organizations need to set up the development processes, such as ASPICE, ISO/TS 16949.

[00049] From the technical aspect, the method to ensure the development quality is to do the verifications or tests, and methods to test said software consist of:

a. to run the product if it is a piece of software source code or hardware component, or

b. to simulate the software if it is a design concept, or

c. to review the product if it is a document, or

d. to plan the activities and review the execution if it is a development procedure.

[00050] From the management aspect, the quality control is to have the qualified development process in place, which demonstrates that the organizations have the established procedures to develop the safety software, which should include:

a. The established development processes, such as ASPICE, to direct the software developments, or re-use the existed software and technology for the development.

b. The established development processes are compliant with the industrial standards, such as ISO/TS 16949, ISO 26262, UN ECE 155 / 156.

[00051] The benefits of disclosed embodiments for the safety development in a computing system are:

a. Full system error detection: provide the solution to fully detect the errors in the system under development, which consist of only two types of errors: data value error and data timing error.

b. Approach to achieve the safety: provide the defined solution to achieve the three aspects that are needed by the safety in the system under development, which will reuse the system operation concept and will mainly focus on the data in the system operation concept, so the development will be efficient, and the result will be accurate and consistent.