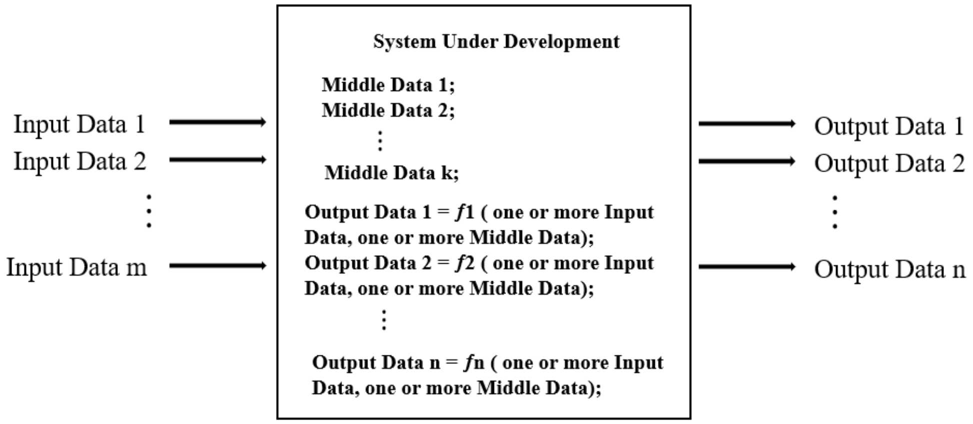
Data-Driven Approach of Safety Development

Introduction

In modern automotive ECU systems, safety development relies heavily on understanding and managing data flow. By leveraging a data-driven approach, systems can be designed to ensure reliability, availability, and quality. This methodology focuses on deriving output signals from input signals through intermediate results, as illustrated in the figure below.


System Under Development
Middle Data 1;
Middle Data 2;
Middle Data k;
Output Data 1 = ƒ1 (one or more Input Data, one or more Middle Data);
Output Data 2 = ƒ2 (one or more Input Data, one or more Middle Data);
Output Data n = ƒn (one or more Input Data, one or more Middle Data);
Input Data 1
Input Data 2
Input Data m
Output Data 1
Output Data 2
Output Data n

Data Relationships

The relationships between data in a computing system can be expressed as:

Output Data 1 = ƒ1 (Input Data 11, …, Input Data 1i, Middle Data 11, …, Middle Data 1j);

Output Data 2 = ƒ2 (Input Data 21, …, Input Data 2l, Middle Data 21, …, Middle Data 2p);

…

Output Data n= ƒn (Input Data n1, …, Input Data nq, Middle Data 1n, …, Middle Data nr).

These relationships form the basis of the **System Operation Concept**, which is central to every system development process. This concept defines:

- How data is transmitted and transformed.
- Which devices or functions handle specific data.

Key Steps in System Development:

1. **Design Middle Data**: Define intermediate results required to derive outputs from inputs.
2. **Establish Relationships**: Develop mathematical or logical relationships (ƒ1, ƒ2, … ƒn) connecting input, middle, and output data.
3. **Perform FMEA**: Analyze failure modes, causes, and effects for each data relationship.

Error Management

Errors in data handling can be categorized as **Data Value Errors** or **Data Timing Errors**, with distinct strategies for detection and correction depending on the data type:

1. **Input Signals**:
    - **Data Value Errors**: Detected using CRC or checksum embedded in communication protocols.
    - **Data Timing Errors**: Monitored using timing parameters defined in databases like DBC or XML files.
2. **Output Signals and Middle Results**:
    - **Data Timing Errors**: Difficult to detect due to the serial nature of ALU processing. Mitigation includes:
        - Internal or external watchdogs.
        - Carefully designed schedulers and task arrangements.
    - **Data Value Errors**:
        - Detected using ECC for transmission errors.
        - Mitigated using range checks, plausibility models, and redundancy for transformation errors.

Safety Mechanisms

Safety mechanisms in automotive ECU development are designed to achieve:

1. **Reliability**:
    - Ensures the system behaves as implemented.
    - Includes detecting and correcting errors or issuing warnings.
2. **Availability**:
    - Maintains functionality through redundancy, e.g., dual object detection mechanisms (radar and camera).
3. **Quality**:
    - Ensures the development process adheres to standards like ASPICE and ISO/TS 16949.

Example: A braking system includes:

- **Primary System**: Electronic Control Braking System (ECBS).
- **Backup System**: Electronic Parking Braking System (EPBS).

Redundancy increases safety but also adds complexity and cost. Effective system engineering balances these factors.

Balancing Redundancy and Complexity

While redundancy enhances availability, it introduces challenges:

- **Increased Costs**: Additional components raise manufacturing expenses.
- **Complexity Risks**: More components can lead to conflicting signals (e.g., discrepancies between radar and camera data).

Solution: Use triple redundancy with voting mechanisms to improve reliability while mitigating complexity.

Quality Assurance

Quality assurance is essential to safety and is achieved through:

1. **Management**:
   - Establishing robust development processes (e.g., ASPICE).
2. **Technical Measures**:
   - Conducting system integration tests and black-box verifications.

Key Objectives:

- Ensure requirements meet customer demands.
- Verify design specifications align with requirements.
- Confirm implementations adhere to specifications.

Conclusion

The **Data-Driven Approach** provides a comprehensive framework for safety development by focusing on:

- Data flow from input to output.
- Relationships between data.
- Systematic error detection and correction.

This methodology ensures:

- **Reliability**: Systems operate as implemented.

- **Availability**: Redundant mechanisms maintain functionality.
- **Quality**: Development adheres to rigorous standards.

By aligning safety development with data-driven principles, automotive systems can meet the challenges of modern autonomous driving scenarios, ensuring safety, reliability, and efficiency.