SOFTWARE REQUIREMENTS ANALYSIS (SWRA)

Instructor: Dr. Hany H. Ammar

Dept. of Computer Science and Electrical Engineering, WVU

OUTLINE

- Introduction to Requirements Analysis and the SW Requirements Specifications (SRS) document
- Structured Analysis for Real-Time (SART) Software Using ICASE

Introduction to Requirements Analysis, The SRS Doc

- The specification of SWRA phase in the DOD standard MIL-STD-498 also focuses on analyzing the requirements and developing a logical model for each computer software configuration item (CSCI)
- The output of this phase is the the Software Requirements Specification (SRS) document (See Table 1, section 3.1.3 of the notes)
- The SRS starts in the first section by identifying the scope of the CSCI, presenting a system overview, and a document overview

1. Scope.
    1.1 Identification
    1.2 System overview
    1.3 Document overview
2. Referenced documents.
3. Requirements
    3.1 Required states and modes
    3.2 CSCI capability requirements
        3.2.x (CSCI capability)
    3.3 CSCI external interface requirements
        3.3.1 Interface identification and diagrams
        3.3.x (Project-unique identifier of interface)
    3.4 CSCI internal interface requirements
    3.5 CSCI internal data requirements
    3.6 Adaptation requirements
    3.7 Safety requirements
    3.8 Security and privacy requirements
    3.9 CSCI environment requirements
    3.10 Computer resource requirements
        3.10.1 Computer hardware requirements
        3.10.2 Computer hardware resource utilization requirements
        3.10.3 Computer software requirements
        3.10.4 Computer communications requirements
    3.11 Software quality factors
    3.12 Design and implementation constraints
    3.13 Personnel-related requirements
    3.14 Training-related requirements
    3.15 Logistics-related requirements
    3.16 Precedence and criticality of requirements
4. Qualification provisions
5. Requirements traceability

Table 1: Outline of the Software Requirements Specification Documents

Introduction to Requirements Analysis, The SRS Doc

- The second section lists the number, title, revision, date, and source of all documents referenced in this specification.
- The third section, the largest and most important section contains the detailed specifications of the CSCI as follows
- The states and modes of operation of the CSCI are clearly specified (e.g., idle, ready, active, post-use analysis, training, degraded, emergency, backup, wartime, peacetime)

Introduction to Requirements Analysis, The SRS Doc

- Each requirement or group of requirements in this specification must be correlated to the states and modes in which they belong.
- The SRS specifies the capability or functionality requirements in terms of control processing and data processing capabilities of the CSCI (e.g., data flow/control flow diagrams are used in SART)
- Requirements pertaining to the CSCI's external interfaces may be presented in the SRS or in one or more Interface Requirements Specifications (IRSs) documents referenced from the SRS

Introduction to Requirements Analysis, The SRS Doc

- Internal interfaces and data requirements between capabilities of the CSCI must be specified
- Non-functional requirements such as *safety*, *Security and privacy* requirements, and quality factors such as *reliability and availability* requirements must also be specified along with the *environmental* requirements
- Computer resource requirements in terms of HW, SW, and Communication resources must be specified

Introduction to Requirements Analysis, The SRS Doc

- Design and implementation constraints (e.g., required databases,particular design or implementation standards or languages, Flexibility to changes in technology, and expendability)
- Precedence and criticality of requirements listed in the previous subsections
- Section 4 specifies the qualification methods used to ensure that the requirement in section 3 has been met.
- In Section 5, Traceability is established from each CSCI requirement in this specification to specific components and sections in the SSS and SSDD docs

Example SRS Doc:

Data Processing Unit (DPU) Flight Software (FSW)

OUTLINE

- Introduction to Requirements Analysis and the SW Requirements Specifications (SRS) document
- Structured Analysis for Real-Time (SART) Software Using ICASE

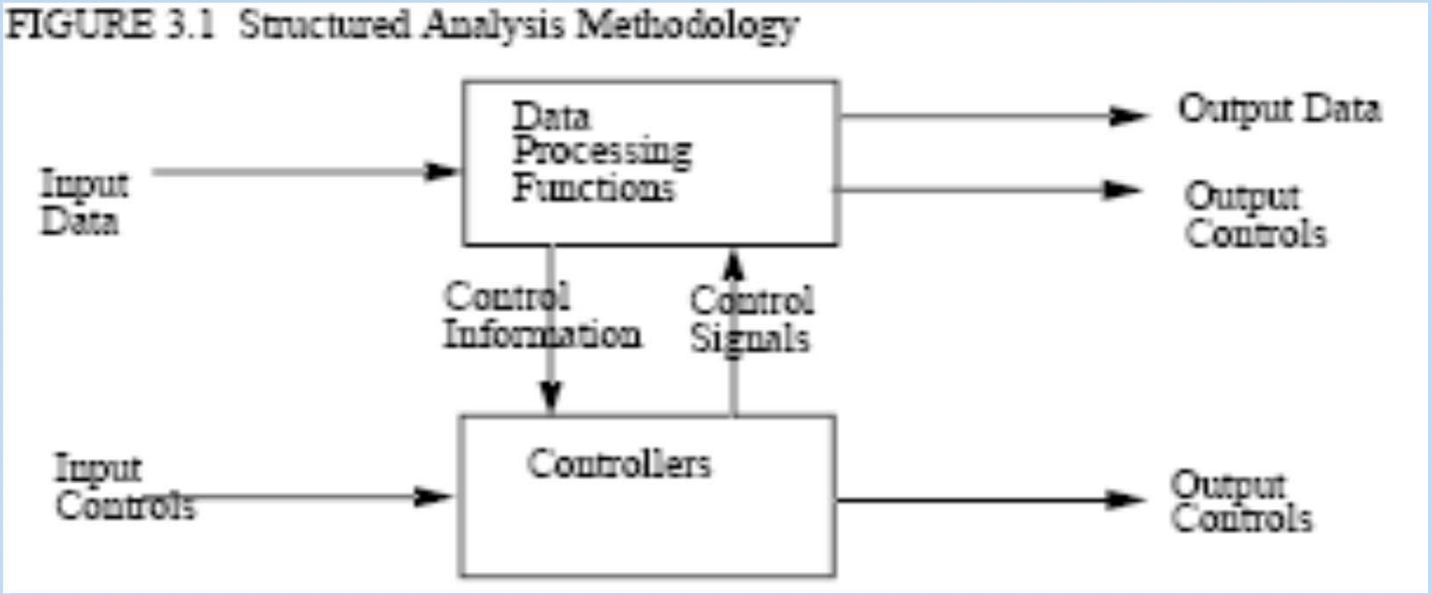
Structured Analysis for Real-Time (SART) Software

The SART Methodology (See Figure 3.1, Sec. 3.2.1, page 3-13)

- The SART model is divided into two main types of elements; *data processing functions* and *controllers*
- Data processing functions process input data, produce output data and controls, and send control information to the controllers
- Controllers process input controls, activate or deactivate data processing functions thr' control signals, and also produce output controls

The SART Methodology

FIGURE 3.1 Structured Analysis Methodology



Structured Analysis for Real-Time (SART) Software

SART models consist of the following Notation

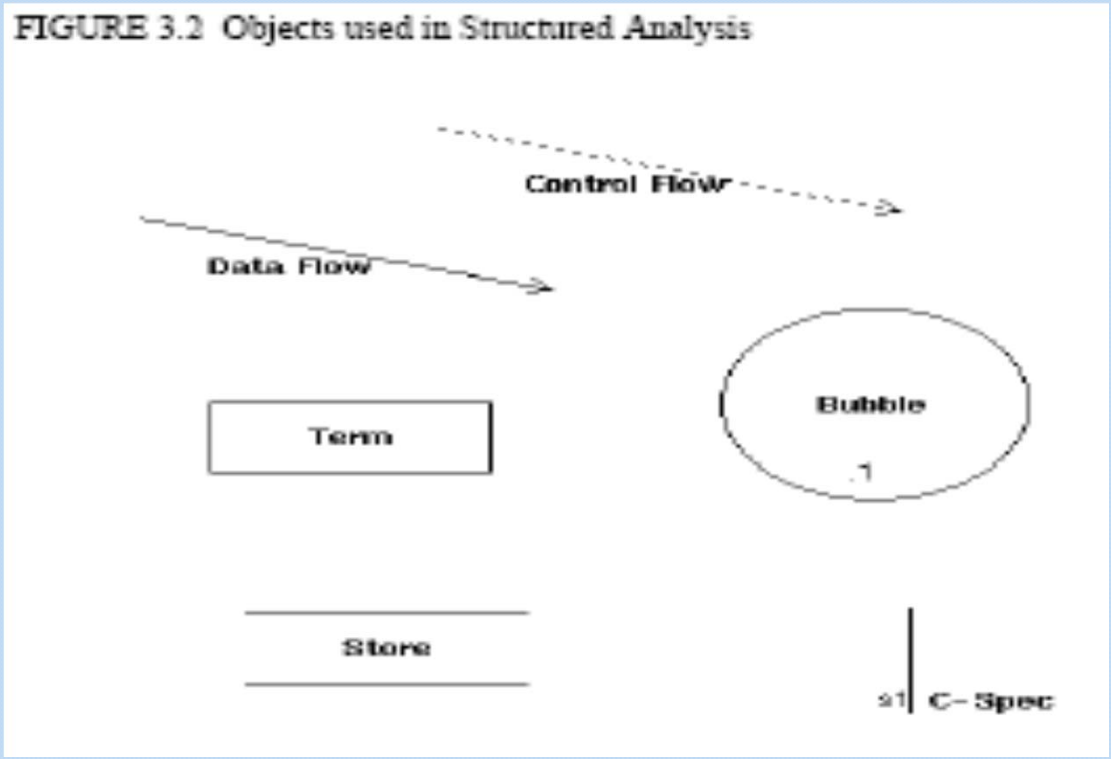
- **Data Flow and Control Flow Diagrams** (Fig. 3.2) **(DFDs/CFDs)** consisting of

  data processing nodes (bubbles), Terminators

  control nodes, or controllers (bars), and

  data/control flows and stores
- **Control specifications (C-specs)** are used to describe the details of controls nodes (state diagrams or Tables)
- **Process specifications (P-specs)** are used to described the details of the data processing nodes (text, scripts)

Data Flow and Control Flow Diagrams (Fig. 3.2, elements of DFD/CFD models)

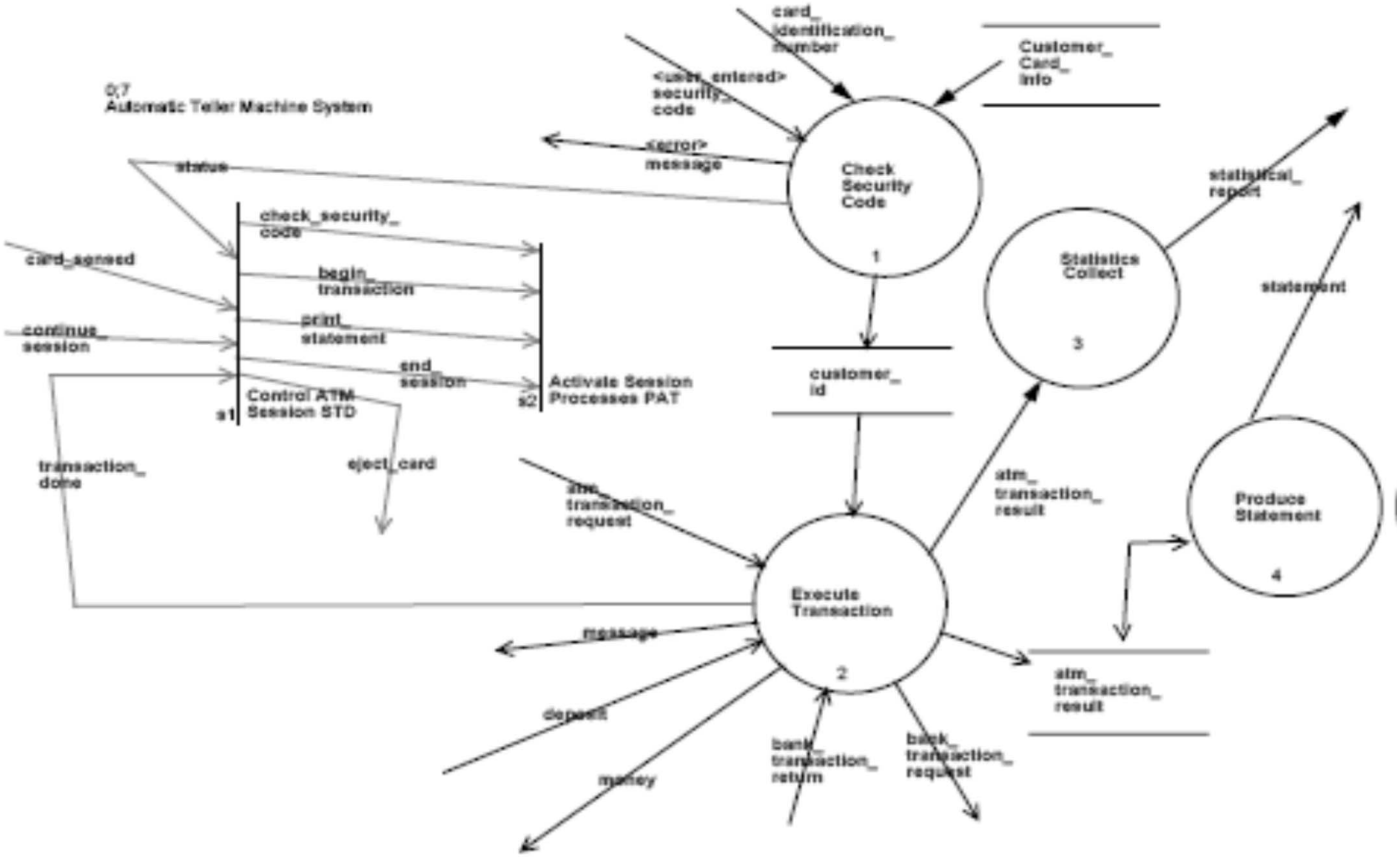


Structured Analysis for Real-Time (SART) Software

- **Data Dictionary (DD)** which defines all the information flows and the data and control stores in the system. It contains Diagrams/script/or text that define each information item and its value range
- **Control Flows vs. Data Flows:** any information item used directly for controlling the data processing activities or is specified as an input or an output of a control node must be designated as a control flow information item,

(see Figure 3.3, page 3-19 for an example of a DFD/CFD)

The DFD/CFD model of
The Automatic Teller Machine Example



Structured Analysis for Real-Time (SART) Software

- **Process Specifications (P-specs)** determines how the output data/control items from a process in the DFD are determined from the input data (see Fig. 3.4, page 3-21)
- Input controls to P-specs are Not allowed
- **Control Specifications (C-specs)** determines in detail how/when the out control flows of the control node are asserted
- **C-specs** also specifies the condition under which the processing nodes in the corresponding DFD is activated

P-Spec Example

FIGURE 3.4 P-Spec of Process 1.8

```
NAME:    1

TITLE:    Check Security Code


 INPUT/OUTPUT:
 Customer_Card_Info: data_in
 card_identification_number: data_in
 security_code: data_in
 customer_id: data_out
 message: data_out


 BODY:
 Look up the card identification number in the Customer_Card_Info
 store. If the atm card's security code matches the <user entered>
 security_code, send out the corresponding customer id. If the
 <user_entered>security_code does not match the assigned security_
 code, send an <error>message to the Customer.
```

Structured Analysis for Real-Time (SART) Software

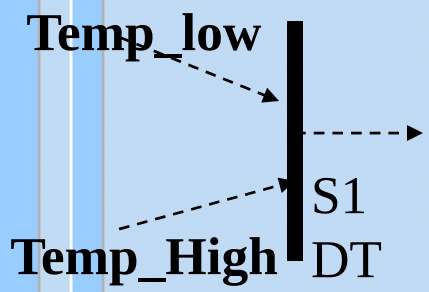
- The notations for C-specs are divided into four different types:
  - Decision Tables (DTs)
  - Process Activation Tables (PATs)
  - State Transition Diagrams (STDs)
  - State/Event Matrices (SEMs)

Structured Analysis for Real-Time (SART) Software

- Decision Tables (DTs) specify combinational controllers (i.e., controllers with only one state)
- Each row in a DT specifies the values for the output control items for a combination of input control items
- The combination of inputs not specified by a row in the table are assumed to be don’t cares

(see Figure 3.5, page 3-22)

Example: C-Spec showing a Decision table of Heating_Request_Controller



S1 is defined
in DFD/CDF 1
of an SART model

FIGURE 3.5 An Example a DTS

1-s1;1
Heating_Request_Controller

Temp_low	Temp_High	Heat_Req_1
"TRUE"		on
	"TRUE"	off

Structured Analysis for Real-Time (SART) Software

- **Process Activation Tables (PATs)**
- used to specify a combinational controller which has no explicit output controls. It used to specify process activation for a given combination of input controls
- A PAT is a special case of a DT in which the names of the processes to be activated are specified instead of the output control flows.

(see Figure 3.6, page 3-23)

Example: C-Spec showing a Process Activation table of Monitor Sensor controller

The Controller is labeled S1 in the DFD/CFD 1 of an SART model Of the Aircraft Monitoring System (AMS), it has 3 input control Signals, and triggers the activation of processes number 1,2, and 3

FIGURE 3.6 An example of a PAT.

1-s1;4

Monitor Sensor PAT

sensor_data_ received	fuel_data_ received	record_ timeout	2	3	1
'TRUE'			1		
	"TRUE"			1	
		'TRUE'	2		1

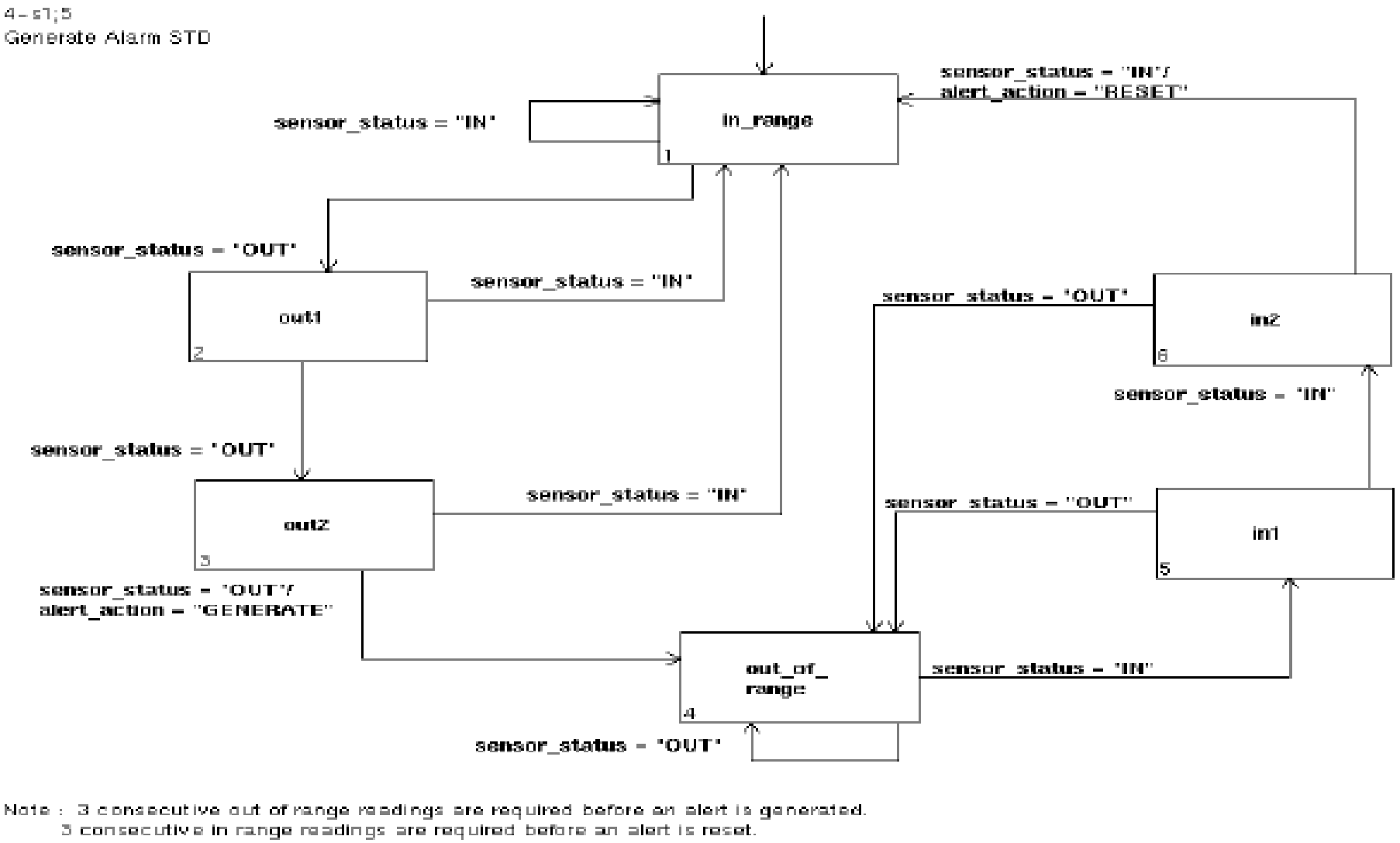
Structured Analysis for Real-Time (SART) Software

State Transition Diagrams (STDs) (Fig. 3.7, 3-25)

- STDs specify controllers consisting of a sequence of states for sequential controls
- A rectangle is used to define each state and directed arcs between rectangles specify transitions from one state to another
- A state transition is caused by a specific event consisting of a combination of input control values and produces actions
- Actions consist of process activations and/or a combination of output control values.

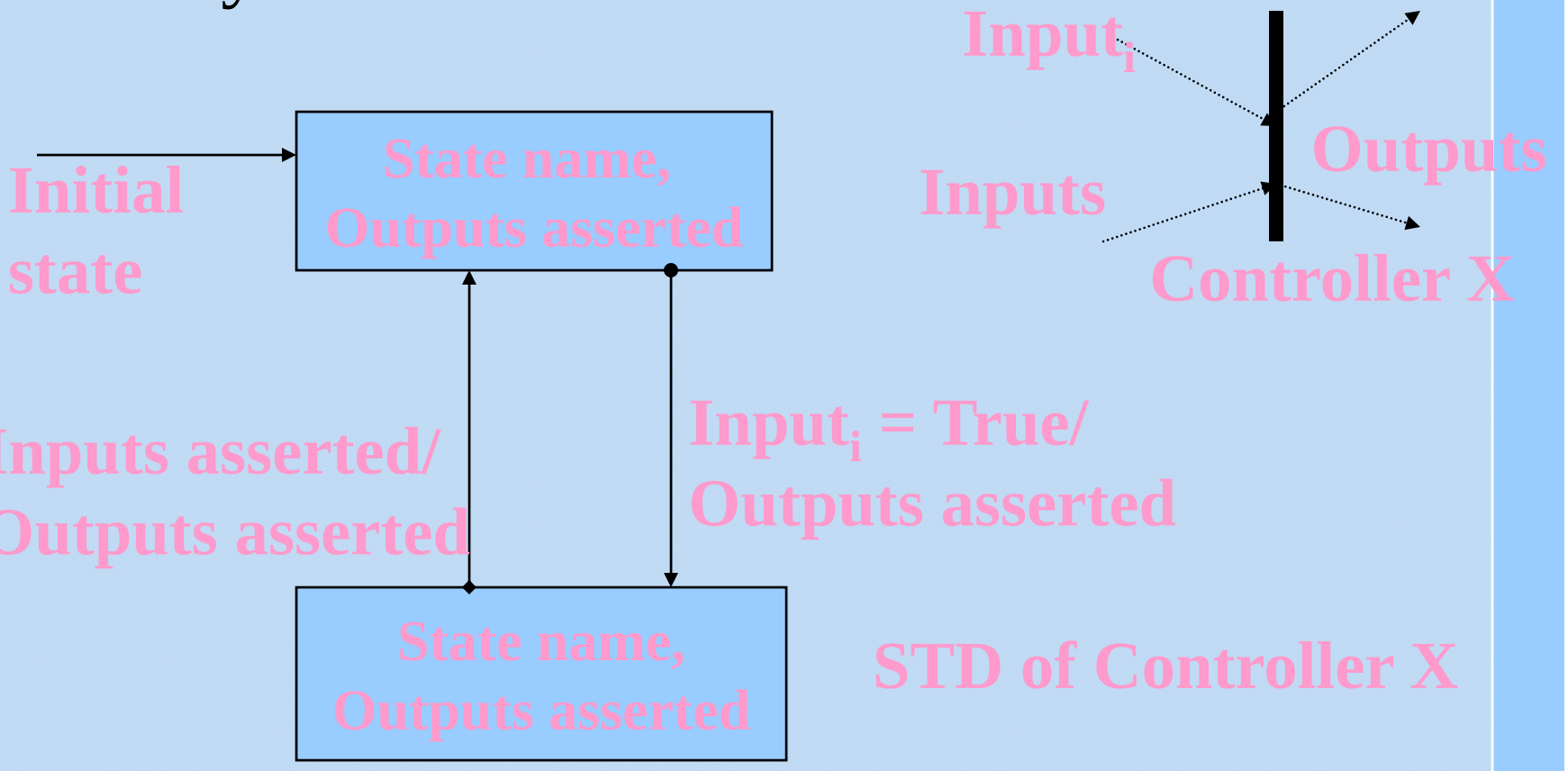
Example: A C-Spec using an STD of the Generate Alarm Controller S1 in DFD/CFD 4 in the SART model

Of the Aircraft Monitoring System, The controller has one input control (sensor_status) and one output control (alert_action)



Structured Analysis for Real-Time (SART) Software

- Every STD contains an initial state



Structured Analysis for Real-Time (SART) Software

State/Event Matrices (SEMs) (Fig. 3.8, 3-26)

- contain the same information contained in STDs but in a tabular form
- Each row in an SEM corresponds to a particular state of the controller
- The set of columns consists of event columns followed by an actions column
- Each input of the controller is represented by an event column in the table
- The table cells for the actions column specify the actions performed for each state (following the Moore model)

Example: A C-Spec showing a state event matrix of the Monitor Sensor controller S2 in DFD/CFD 1of AMS. The controller has two states and three events.

FIGURE 3.8 An example of an SEM

1–s2;2

Monitor Sensor SEM

States/ Events	sensor_data_ received	timeout	one_second_ interrupt
Idle	*don't care*	/Idle	reading_request; set_timer/ Polling Sensor
Polling Sensor	/Idle	record_timeout; sensor_data_ received/Idle	*don't care*

Structured Analysis for Real-Time (SART) Software

- The table cells for the event columns are filled to specify the actions performed/next state for a given event and a given current state
- For cases in which the event at a particular state is “ignored” or simply “can't happen” the cell is left blank with no specifications of actions or next state.

Structured Analysis for Real-Time (SART) Software

- The ***Data Dictionary*** contains the definition of all the information items consisting of flows as well as the stores both for data and controls
- Information items are divided into two types: primitive data items and compound data items
- Primitive information items are those items not composed of any other data items
- Examples of primitive data/control items are temperature sensor reading, a binary switch reading, operation status, or an identification number

Structured Analysis for Real-Time (SART) Software

- Compound data items on the other hand may be composed of other compound data items and/or primitive data items
- They must be specified using
  - scripts (see Table 2, 3-27for Notations used in some ICASE tools such as teamwork),
  - data structure diagram (see the Stp Data Structure editor on-line docs
- Examples of compound data items are operator command which consists of several different types of commands, sensor data consisting of the readings of different sensors (see examples, 3-28)

Structured Analysis for Real-Time (SART) Software

Examples of compound data definition using scripts

sensor_data = sensor_id + sensor_value

sensor_id = sensor_type + sensor_No

sensor_type = [“Temp” | “Pressure” | “Fuel” | “Smoke”]

* four types of sensors are assumed, these are temperature sensors, engine pressure sensors, fuel capacity sensors, and smoke detection sensors *

sensor_No = 2{decimal_digit}5

the sensor number is 2 to 5 decimal digits

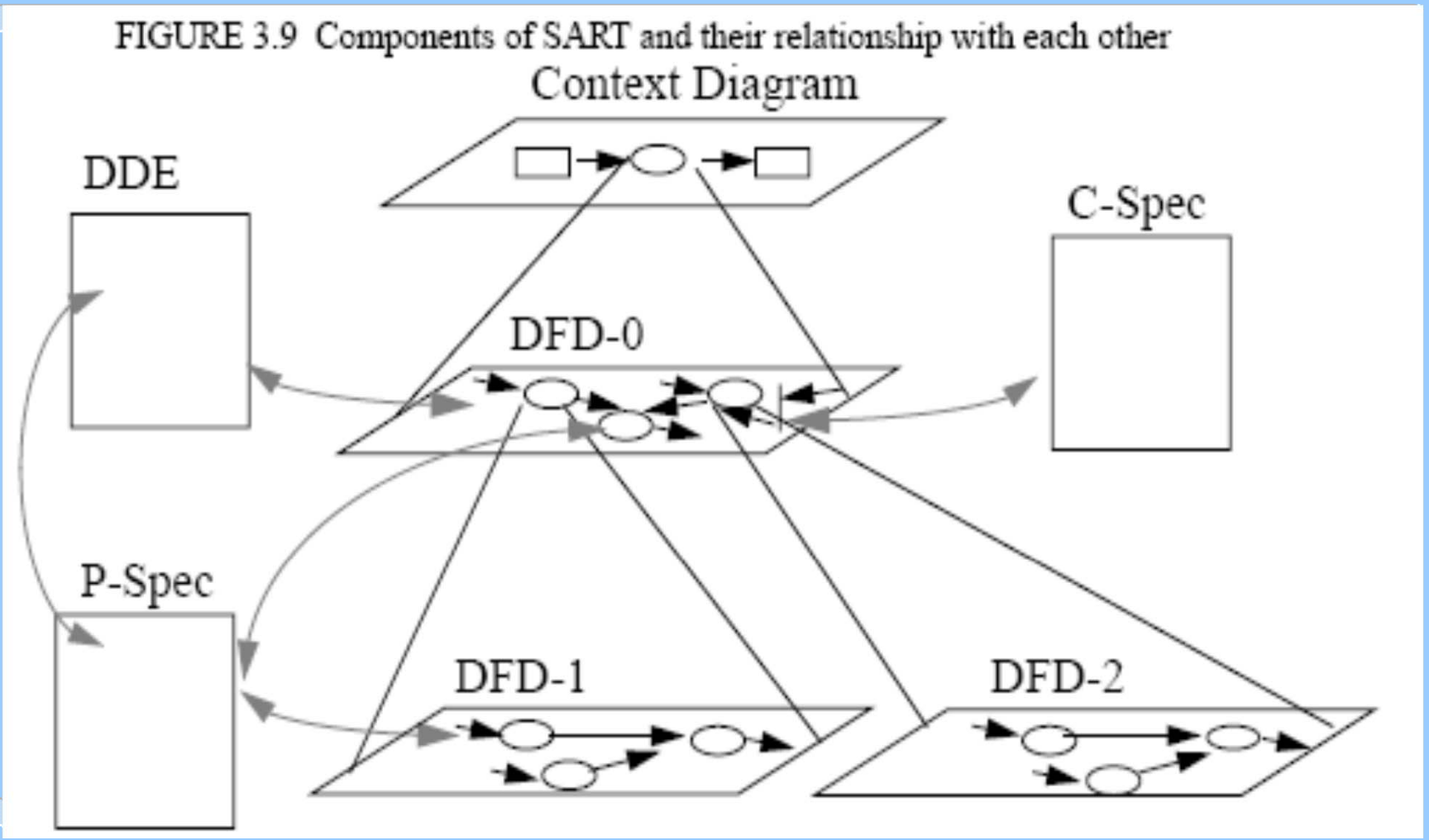
Structured Analysis for Real-Time (SART) Software

The Structured Analysis Model (Fig. 3.9, 3-29)

- Consists of several levels of hierarchy, The specification developed at each level is simple and readable
- The top level contains the Context Diagram (CD) which defines the external entities interacting with the modeled software
- The modeled software is represented by a single process(or bubble) in the middle of CD
- Input/Output data/control flows are specified using DFD/CFD notation between the external entities (represented by Terms) and the modeled software

The SART Model

FIGURE 3.9 Components of SART and their relationship with each other



Structured Analysis for Real-Time (SART) Software

- All data/control flows represented as external interfaces should be defined in the data dictionary
- The following level of hierarchy defines the major processes and controllers in the modeled software this level is named DFD0/CFD0
- The input/outputs defined in the CD at the top level are shown as inputs/outputs to/from processes or controllers at DFD0/CFD0 level
- Each process in DFD0 is either specified by a lower level DFD/CFD if it is a non-primitive process, or by a P-spec if it a primitive process

Structured Analysis for Real-Time (SART) Software

- The lower level DFD/CFD (or P-spec) is numbered (or named) after the number (or name) of its process in the higher level DFD
- Controllers in a DFD/CFD are specified by C-spec sheets containing a DT,PAT,STD, or an SEM
- The C-spec Sheet is named after the controller name in the DFD/CFD
- The model hierarchy can span many levels in large scale systems, where the lowest level contains all primitive processes

Example of a Context Diagram of an ATM system

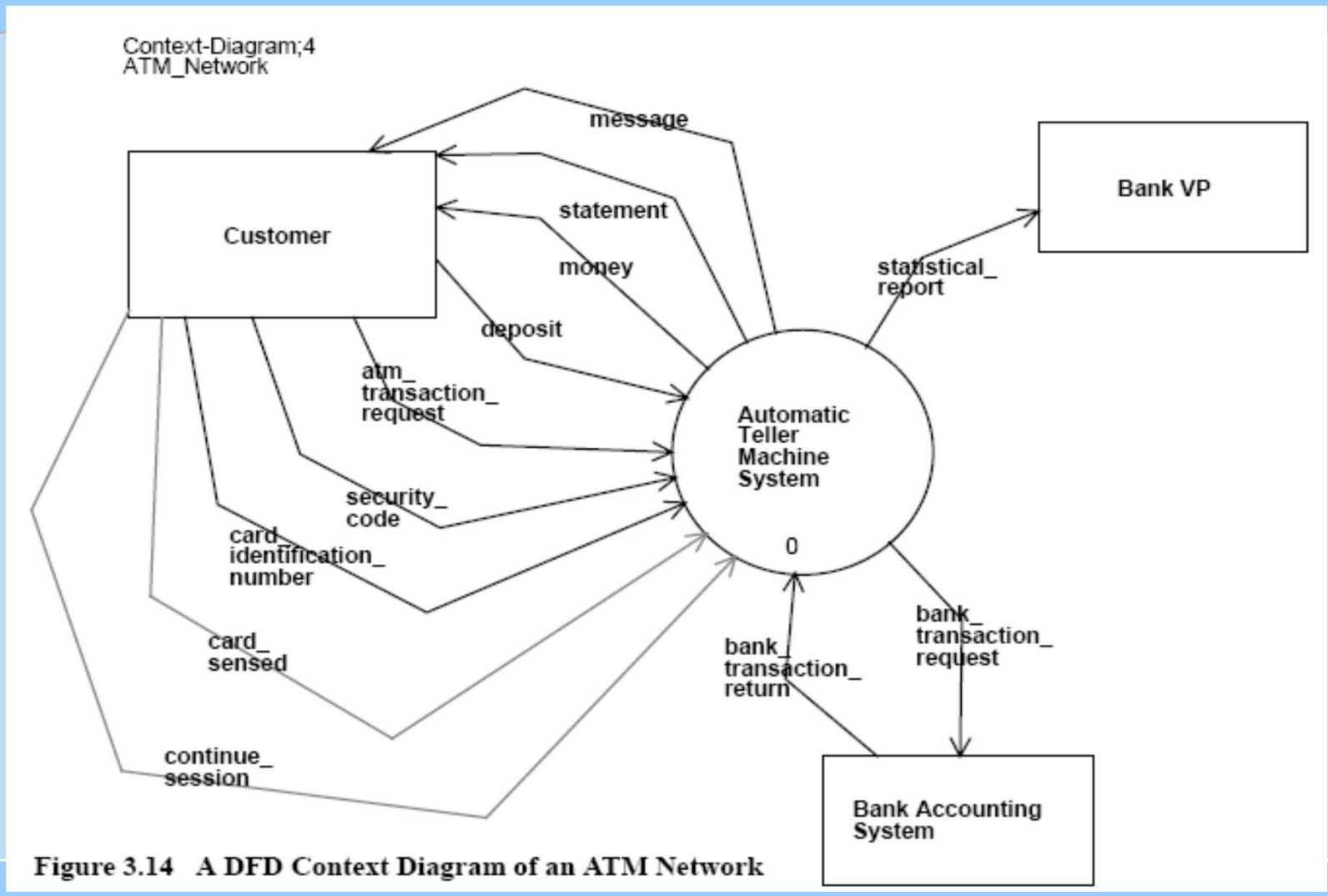
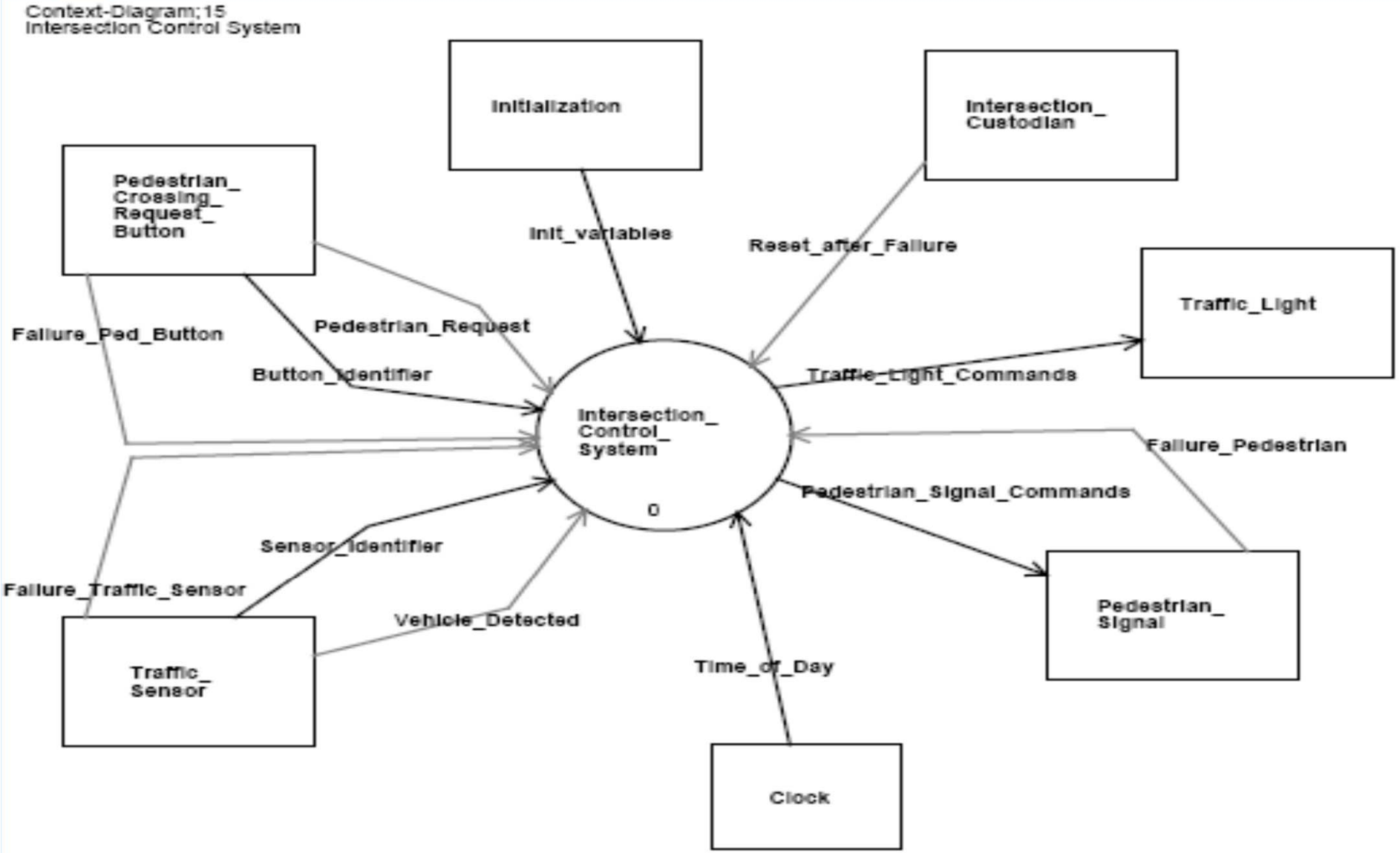


Figure 3.14 A DFD Context Diagram of an ATM Network

Example of a Context Diagram of a Traffic intersection control system



Structured Analysis for Real-Time (SART) Software

Guidelines for Developing the SART Model

- In the CD, External entities and data/control flow can be abstracted by defining super types or generic types.

Decomposition Criteria

- In DFD/CFDs, Partition a process into lower level processes and controllers such that each have a well defined task
- Define lower level functions needed to input, monitor, or consume and validate the input data or control flows specified for the higher level process

Structured Analysis for Real-Time (SART) Software

- Define lower level processes needed to operate on the processed input data to produce the output data specified for the upper level processes
- Partition a process to lower level processes in such a way that tends to minimize the interconnections (in terms of direct data/control flows) between them.

Structured Analysis for Real-Time (SART) Software

Aggregate a set of processes in a DFD into one process such that

- functions that work together to accomplish a specific task (e.g., the task of interacting with the operator)
- functions that have strong interconnections such as having access to common data stores or a large number of direct data/control flows
- functions that have common input and/or outputs to the same external entities

Structured Analysis for Real-Time (SART) Software

The SART model of Aircraft Monitoring System

- DFD Context-Diagram Aircraft Monitoring System
- DFD 0 Monitor Aircraft
- PAT 0-s1 Monitor Aircraft PAT
- DFD 1 Monitor Sensor
- PAT 1-s1 Monitor Sensor PAT
- SEM 1-s2 Monitor Sensor SEM
- PS 1.1 Record Time-out
- PS 1.2 Determine Range

Structured Analysis for Real-Time (SART) Software

- PS 1.3 Determine Fuel Capacity
- PS 1.4 Receive Sensor Data
- DFD 4 Generate Alarm
- SEM 4-s1 Generate Alarm STD
- PAT 4-s2 Generate Alarm PAT
- STD 4-s12 Generate Alarm STD
- PS 4.1 Add Out-of-Range Alert to Queue
- PS 4.2 Reset Lamp
- PS 4.3 Add Smoke Detector Alert to Queue
- PS 4.4 Add Fuel Alert to Queue