Chapter

9

The Structural View

Module-Charts

This chapter deals with the language of Module-charts. Module-charts describes the structural view—sometimes called the *architectural view*—of the system under development. This view deals with the system's actual structure, that is, its implementation, and should be contrasted with the conceptual model described by the other views. Module-charts is typically used in the high-level design stage of the project.

9.1 Structural Description: High-Level Design

The structural view captures the system's high-level design. A structural description of the system specifies the components that implement the capabilities described by the functional and behavioral views. These components may eventually materialize as hardware, software, or even as humans. As in the other views, they may be arranged in a hierarchy. The structural view also specifies the communication lines that connect the components. These communication lines can be described in terms of physical links or flowing information.

Let us now present the structural description of our EWS example. The subsystems constituting the implementation are as follows:

CCU (control and computation unit)	The central CPU, within which the main control of the system and the basic computations take place.
SIGNAL_PROCESSOR	The subsystem that processes the signal produced by the sensor and computes the value to be checked. It consists of an analog-to-digital unit and a high-speed processor that works at the required checking rate.
MONITOR	The subsystem that communicates with the operator. It consists of a KEYBOARD for commands and data entry, and a SCREEN for displaying messages.
ALARM_SYSTEM	The subsystem that produces the alarm, visually, audibly, or both.
PRINTER	The subsystem that receives the messages (text and formatting instructions) and prints them.

The environment systems are the OPERATOR and the SENSOR. By identifying these elements, we define the borders of the system; that is, we determine which facilities are part of the system (e.g., the PRINTER), and which are external (e.g., the SENSOR). Note that the environment components are common to the functional and structural view. We shall discuss this matter further in Chap. 10.

Sometimes, there is a clear correspondence between the top-level activities in the functional view and the top-level subsystems in the structural view. Often, a particular subsystem is responsible for carrying out a single activity from the functional description. Here, for example, the subsystem SIGNAL_PROCESSOR implements the algorithm specified in the activity PROCESS_SIGNAL. However, in many cases the structural decomposition is quite different from the functional decomposition. Thus a single subsystem in the structural view may be responsible for carrying out a number of different activities in the functional view, or an activity may be distributed among several top-level subsystems. In the EWS example, the CCU subsystem carries out both the EWS_CONTROL and COMPARE activities, whereas the DISPLAY_FAULT activity is divided into subactivities that are distributed among the ALARM_SYSTEM and MONITOR subsystems.

The communication between the subsystems of the EWS is discussed in Sec. 9.3.

9.2 Internal and External Modules

In our approach, the structural view is represented by the language of Module-charts and the associated entries in the Data Dictionary. The components, or subsystems, are called *modules* and are depicted as boxes (rectangles or rectilinear polygons). Names appear inside the boxes, adhering to the naming conventions of App. A.1. As in Activity-charts and Statecharts, decomposition is captured by multilevel encapsulation. The general terminology is also similar: we have *basic modules, submodules, parent modules, descendants,* and *ancestors.*

There are two kinds of internal modules: *execution modules,* drawn with solid lines, and *storage modules,* drawn with dashed sidelines like the data-stores in an activity-chart. The *external modules* represent the systems that are outside the top-level module and are drawn with dotted lines. An external module retains this line convention even if it really functions as storage, such as a disk or computer memory. Like external activities in an activity-chart, the external modules may correspond to real environment modules that are external to the entire system under description or to internal modules in other module-charts. This issue is discussed in Chap. 12. As in the case of box elements in other charts, sibling internal modules cannot have the same name. Several external modules, however, can bear the same name, in which case they are all occurrences of the same external module.

Figure 9.1 shows the structural decomposition of the EWS, including a storage module `DISK`, which stores the fault messages. We have left the arrows unlabeled; they will be discussed in the next section.

The following rules govern the allowed encapsulations in a module-chart:

- Execution modules may be submodules of other execution modules only.
- Storage modules may be submodules of other storage modules or of execution modules.
- External modules are always external to an execution module or storage module, and there is no hierarchy of external modules.

As in the other graphical languages, we use the Data Dictionary to specify additional information. Figure 9.2 shows the Data Dictionary entry of the `SIGNAL_PROCESSOR`. In it, we have used an attribute name/value pair to specify that the module is to be implemented in hardware, and the synonym field to identify the component in some

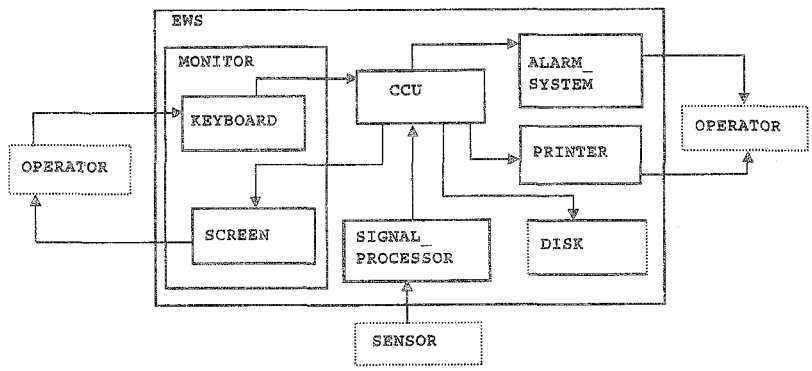


Figure 9.1 Structural decomposition of the EWS.

Module: SIGNAL_PROCESSOR
Defined in Chart: EWS
Synonym: FFT548
Description:
High speed FFT that processes the sensor's signal.

Described by Activity-Chart:

Attributes:

Name	Value
IMPLEMENTATION	HARDWARE

Long Description:
This subsystem processes the analog signal coming from the sensor. It is a standard FFT, that also contains an A/D unit.

Figure 9.2 A Data Dictionary entry of a module.

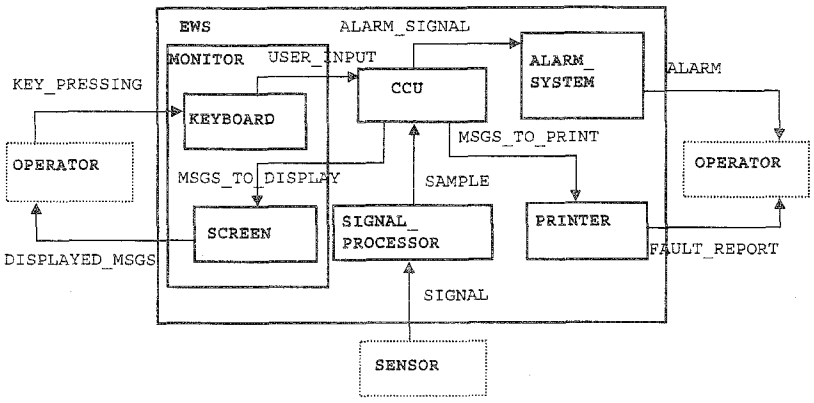


Figure 9.3 Flow of information among modules.

another hardware description. In addition to the standard fields, the Data Dictionary entry for a module contains a special field, `Described by Activity-Chart`, which is used to connect modules with their functional descriptions. This will be explained further in Chap. 10, where we discuss the connections between the functional and structural views. In Fig. 9.2 it is left empty.

9.3 Communication Lines between Modules

The communication between modules can be described to various levels of detail, from merely specifying the physical connections existing between the modules to specifying the details of the information items comprising the interfaces of the modules.

As in the other graphical languages, we draw labeled arrows between the modules. They are called *flow-lines,* as in activity-charts, or *m-flow-lines,* to emphasize that they connect modules. We do not use a different line style to distinguish lines that represent flow of information items from those that depict physical links.

As in activity-charts, lines attached to nonbasic modules carry special meaning. A flow-line emanating from a nonbasic module specifies that the information flowing along it can be produced by any of its descendant modules, and a flow-line leading to a nonbasic target module specifies that the information labeling it is available to any of its descendant modules.

9.3.1 Flow of information between modules

When a flow-line is used to denote information flowing between modules, the label is as in an activity-chart. That is, it can be a data-item, an event, a condition, or an information-flow that may contain several types. These elements are described in Chap. 3. As in activity-charts, the labels cannot contain compound elements. Also, recall that additional information about these elements (such as their physical implementation) can be specified in their Data Dictionary entries.

Figure 9.3 depicts the module-chart for our EWS example, with labels describing the information on the arrows.

Note that some of the elements appearing here appeared along the flow-lines of the corresponding activity-chart in Fig. 2.5, and some are information-flows that contain elements appearing therein. Here, `USER_INPUT` contains the information-flow `COMMANDS`, the data-item `RANGE_LIMITS`, and the condition `SENSOR_CONNECTED`. The precise relationship between the flows in activity-charts and module-charts is discussed in Chap. 10.

9.3.2 Physical links between modules

Arrows in a module-chart may also denote physical communication links, or *channels,* between modules. In this case, information-flows are used to name the links. The Data Dictionary entry for such a flow can be used to specify the type of link and the way the data is represented along it. The actual information elements that flow along the link can be specified in the `Consists of` field.

Figure 9.4 contains an alternative module-chart for the EWS example, showing the physical links. Some of them are really wires (or cables) of various types.

The interface with the user in the EWS is carried out by pressing buttons or by audio or visual outputs. Although these elements are not associated with physical links, they are also shown in the figure.

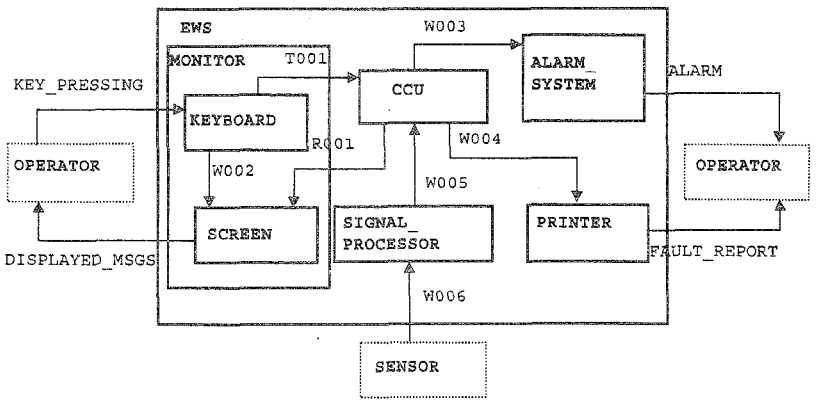


Figure 9.4 Physical links among modules.

Information-Flow: W005
Defined in Chart: EWS
Description: The bus linking SP to the CCU

Consists of:
SAMPLE

Figure 9.5 Information-flow describing a physical link.

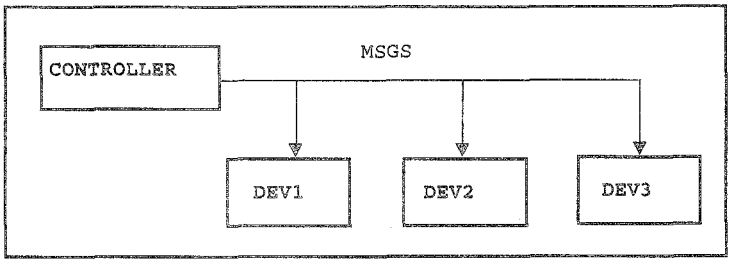


Figure 9.6 Communication link to several devices.

Figure 9.5 shows the Data Dictionary entry of the information-flow W005, which connects the SIGNAL_PROCESSOR and the CCU. The data-item SAMPLE flows inside this communication link.

9.4 Connectors and Compound Flow-Lines

Module-charts contain features that help in preparing clear and uncluttered charts, as do the two other types of charts. Connectors

and compound flow-lines are allowed in module-charts exactly as they are in activity-charts. See Chap. 2.

Joint connectors are often used to depict a flow-line that links several modules. An example is shown in Fig. 9.6, where several peripheral devices listen out for messages arriving along a communication link that emanates from the central controlling unit.