Characterizing Patterns for Real-Time Systems and Their Environments

Lonnie R. Welch[1] *and Michael W. Masters*[2]

I. Templates for Common Patterns in Real-Time Systems

This paper presents generic templates that represent patterns for real-time subsystems and describes common instances of the templates.

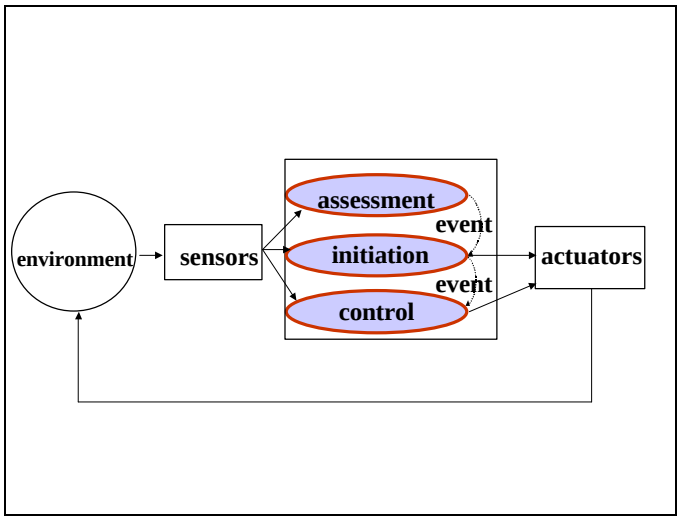

Figure1. Generic real-time system.

The authors have been involved with the engineering of real-time systems for air traffic control, satellite command and control, satellite constellations, air defense and autonomous mobile robots. Each of these systems can be characterized by the architecture shown in Figure 1. The physical environment is monitored by sensors, which provide data for *assessment.* Significant events cause *initiation* of an action to effect the environment through an actuator. *Control* of the actuator is performed to cause the action to complete successfully in highly dynamic environments. The patterns for assessment, initiation and control are detailed in the remainder of this section.

[1]Laboratory for Intelligent, Real-Time, Secure Systems; Ohio University; welch@ohio.edu.

[2] Naval Surface Warfare Center; Dahlgren Division; MastersMW@NSWC.NAVY.MIL.

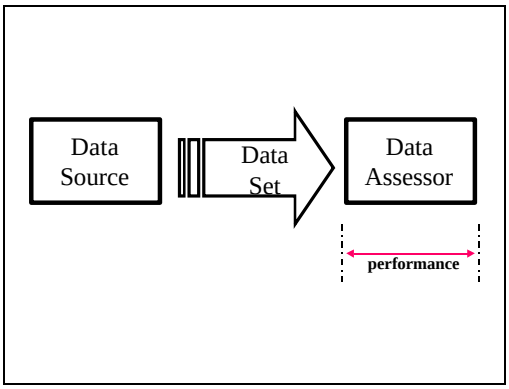

Figure 2. Template for the assessment pattern.

As shown in Figure 2, a generic assessment pattern contains a *data source*, a *data set* and a *data assessor*. A data source is typically a sensor, which *periodically* produces a set of *data elements* that characterize some aspect of the physical environment. The size of the data set can be *variable* and the elements of the data set may be *heterogeneous.* The period, or rate, at which the data set is produced can also be variable. The data elements are evaluated by a data assessor, which decides whether the actuators should perform actions to control the environment. The characteristics of the system and the environment often dictate that the assessment be completed before a *deadline*.

The initiation pattern is similar in form to the assessment pattern. Its behavior, however, is event-driven. Thus, it consists of an *event source*, an *event*, and an *event handler*. An assessment component (the event source) produces an event. The arrival rate for the events is *variable* and the types of events may be *heterogeneous*. Arrival of an event causes the event handler to plan and initiate an action for an actuator. The characteristics of the system and the environment require that the planning and initiation of the action occur in a timely manner.

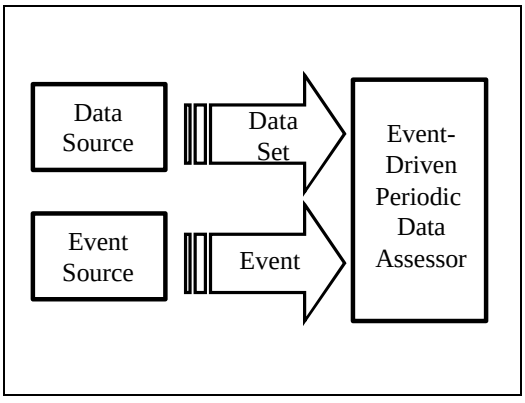

Figure 3. Template for the control pattern.

The control pattern is an aggregation of the assessment and initiation patterns (see Figure 3). It is activated by an event. Once activated, it performs the assessment pattern, periodically processing the items in the data set before a deadline. It also incorporates the action pattern, providing control to the actuator whenever its assessment determines that action adjustment is required. Control is *deactivated* by an event – the completion of the action. Thus, control also has a *deactivation deadline* - the time at which the action must succeed. A deactivation deadline is often a function of parameters that can only be determined dynamically. Since parameters used to compute the deadline may change, the deactivation deadline may change after it is estimated initially.

II. Characterizing Template Instances

Actual (or planned) real-time systems can be considered as containing instances of the assessment, initiation and control pattern templates. Additionally, each real-time technology can viewed as embodying one or more pattern instances in the paradigm that it supports. This section presents a taxonomy for characterizing *instances* of the pattern templates. The taxonomy represents the properties of a real-time component and the properties of its environment.

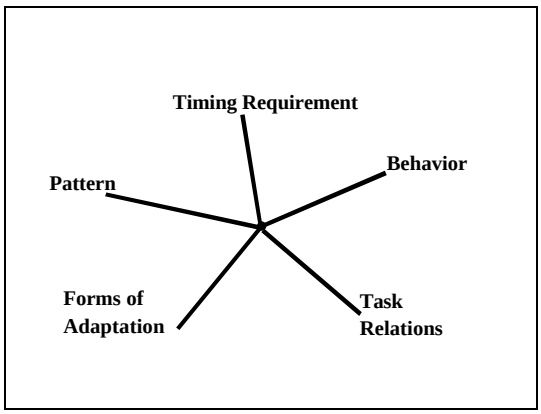

Figure 4. Properties of a real-time component.

As shown in Figure 4, a real-time component can be characterized by its *behavior, task relations, forms of adaptation*, *pattern* and *timing requirement.* The behavior may be periodic, event-driven, or event-driven-periodic. Since a pattern instance often consists of multiple tasks and objects, it is important to determine if, and how, these are dependent (e.g., for communication, execution order, or initialization). To meet performance requirements in dynamic environments, a real-time component may be adaptable in a variety of ways, including its resource allocation, degree of concurrency, slack and

fidelity. A component can also be classified as an instance of the assessment, initiation or control pattern. The timing requirement of a component is described in terms of its strictness (hard, firm, soft, importance, or utility); complexity (a single requirement or multiple requirements); granularity (hours, minutes, seconds, ...); and level of abstraction (instruction, function, object, task or task group).

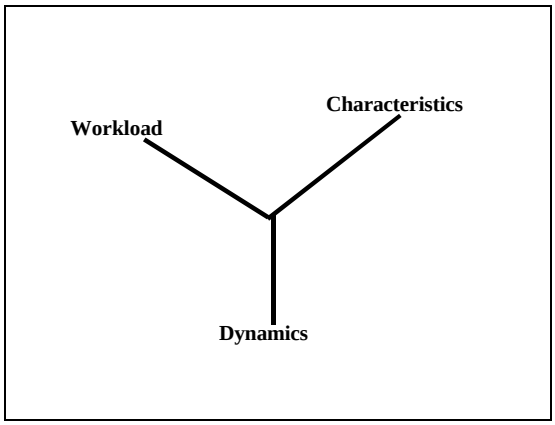

Figure 5. Properties of the environment.

The properties of a real-time system's environment are depicted in Figure 5. Environment *characteristics* can be described (1) statically, (2) with a time invariant stochastic model, (3) with a time variant stochastic model, or (4) with none of the above (if it is highly dynamic). For non-static environments, their *dynamics* can be classified as gradually changing, bursty or hybrid. The *workload* of a real-time system is dependent on the environment. The dependency is with respect to *data set size* or *event arrival rate* (each described by a constant, a set, an interval, a distribution, or as dynamic), *period* (fixed, set, interval or unconstrained) and *data set element type* (which is either homogeneous, a set, an interval, a distribution, or unknown).

III. Conclusions

This paper describes templates and instances for common patterns found in real-time systems and their environments. The patterns have emerged from the study of real-time systems in several domains. The concepts are useful for technology selection. A real-time system and a candidate technology can be characterized according to the taxonomy described in Section II to determine if they match. Furthermore, the taxonomy defines a space for consideration by researchers who would like to identify and to articulate open problems.