3 CONCEPT OF OPERATIONS

This standard describes a general, field-located computing device that must be capable of executing applications software from various developers. Generally accepted systems engineering practices begin from user needs. This section identifies the presently known user requirements for an ATC and begins to identify the associated functions. Because the users' needs and applications are expected to expand in unknown ways in the future, the standard explicitly recognizes that the details of particular future applications use are not completely known at this writing. It is important nonetheless that the support and usage needs of the most commonly known and anticipated applications be defined.

As indicated above, it is the intent of this standard to describe a general-purpose computing device. As such, the ATC can be seen as analogous to a Personal Computer (PC). A difference between this standard and the PC is that a device meeting this standard must be able to withstand the harsh environment of a field-located device with no special cabinet or environmental conditioning beyond that specified separately in the ITS cabinet standard. Another difference is that the ATC must be able to operate remotely in a largely unattended mode. Similar to the PC, the ATC Controller must adhere to a set of programming conventions and interfaces standards such that the applications software that runs in the device can be developed using the Engine Board vendor-supplied BSP as guidance.

The ATC Controller must also have a high degree of reliability, and be easily maintained.

3.1 Problem Statement

One of the largest component costs of today's Intelligent Transportation Systems is associated with the development, testing, deployment and maintenance of applications software. As the current trend continues towards distributing more of the intelligence of ITS out closer to the field, there is an increasing demand for more and more capable field deployable devices. This hardware must run more sophisticated applications software and operate in modern networking environments. The ATC Controller is intended to address these needs.

The ATC Controller is intended as a next generation, "Open Systems" controller [in which hardware interfaces are generically defined, standardized, and adopted by multiple manufacturers] which follows the "Open Systems" lineage of the ATC 2070 and California Model 170 and New York Model 179 controllers. "Open Systems" in this context refers to the concept of separation of hardware from software by standardizing the interface between the two. This allows software to be developed independent of the hardware. "Open Systems" help protect an agency's investment by guarding against premature obsolescence due to a manufacturer's discontinuance of a particular line of equipment or the manufacturer's ceasing of business operations altogether. Additionally, "Open Systems" typically increase equipment procurement competition;

resulting in reduced procurement costs. Deployment, integration, and maintenance costs are also generally reduced because of the commonality and interchangeability of units between various manufacturers reducing spare inventories and technician training costs.

Another important need for "Open Systems" controllers has to do with the occasional need for custom, specially built, applications. Sometimes the demand for a particular application or custom feature is too small, from an industry-wide standpoint, to be of much interest as a product for manufacturers. Nonetheless, a particular problem or research need may require some unique functionality. With "Open Systems", software is written to satisfy a specified set of requirements without special support or permissions from the hardware manufacturer.

3.2 Historical Background

Many of the design choices in this standard are based on historical trends. This history is included to provide a framework for the decisions represented in this standard. It is also recognized that many legacy systems are presently deployed and that any new technology, such as that specified here, must be capable of interfacing accurately and readily within existing networks of deployed equipment. Therefore, it is appropriate to document the known characteristics of elements of the deployed network.

In the early 1970's two concurrent traffic controller standards efforts were initiated in North America. These were the Model 170 standard and the NEMA standard. A brief history of these two standards efforts and the later ATC 2070 standard are presented in the subsections below.

3.2.1 NEMA

The NEMA standard(s) stemmed from a group of manufacturers who joined the NEMA (National Electrical Manufacturers Association) and assembled a core of experienced traffic and electronic engineers to define the first NEMA traffic signal controller. The controller development consisted of an interchangeable electronic device with standard connectors. The NEMA standard further defined traffic terminology and minimum traffic signal control software functionality. Various user agencies that included State, City and County Government Officials were included in this initial definition of the standard.

The initial standard included the standardization of connectors and connections for three MS style connectors. The inputs and outputs were defined and standardized with respect to electrical levels as well as function.

The development process ultimately yielded a document labeled the "TS-1" *Traffic Controller Assemblies* - Standard in 1983. The NEMA standard also defined peripheral devices used in the controller industry and eventually defined the cabinet. The NEMA process requires that every six years the standard is updated and re-ratified. The standard did not cover communications between devices, nor did the standard provide for interchangeability of software functions.

During subsequent years the demand for communications to provide data transfers between local controllers and central control or on-street master systems increased rapidly. The original TS-1 standard had not defined communication and subsequently a non-standard fourth connector evolved that did not allow interchangeability. The TS-1 1989 revision defined/standardized actuated intersection control, provided standards for all cabinet components and added test procedures, and improved interchangeability between manufacturers equipment.

Over the years, further definitions were recommended to define a safer cabinet to controller interface. This new recommendation included a full SDLC communication protocol to allow the traffic controller and the conflict monitor to communicate between each device and check the intended output with what was actually being displayed by the cabinet.

This effort generated the most recent "TS-2" standard in 1992, later updated in 1998, and lastly updated in 2003. The standard outlines an expandable and interchangeable traffic controller, cabinets and peripherals. The TS2 standard replaced individual Parallel I/O lines with time slots in a high speed serial data stream, reducing the amount of cabinet wiring and allowing easier addition of new features. The standard however, did not accommodate interchangeable software among the various manufacturers. Features found in one software package were not available in another's package. Also the front panel displays and the information displayed were all different and non standard. The ATC standard addresses both the interchangeability of software, the standardization of displays and the reliance upon a single operating system.

3.2.2 The Model 170 Specification

The Model 170 specification was developed by Caltrans and New York State DOT to address needs for an “Open Systems” controller for transportation applications. Unlike the NEMA standard, the Model 170 defined controller hardware but not software functionality. The Model 170 approach allows software from any source to be loaded and executed on the controller. The Model 170 obtains its hardware / software independence by requiring, by part number specification, the use of specific integrated circuit chips (for CPU and Serial Communications functions). In addition, a memory map was defined so that software developers would know precisely where to address input and output functions regardless of who manufactured the hardware unit.

While the Model 170’s architecture has been enormously successful and achieves the desired independence of the hardware and software, the Model 170 relied heavily on the specific Motorola CPU and serial communications chips (or suitable substitutes). Unfortunately, these chips have been designated for phased-out obsolescence. The issue is further compounded by the relatively poor computational performance of the Model 170, compared to today’s controller systems. The applications software written for the Model 170 CU is written in assembly language which makes it difficult to move to a different CPU. Also, the Model 170, without a dedicated CPU for communications, cannot handle the performance demands of today's modern packet based high speed communications networks. Few options currently exist for those agencies heavily invested in Model 170 software/hardware to preserve their investments in Model 170 applications software.

3.2.3 The ATC 2070 Standard

The ATC 2070 is a current generation "Open Systems" controller system and is recognized explicitly within this standard. It was originally developed by Caltrans and City of Los Angeles to address some of the shortfalls associated with the Model 170 as discussed above. Its designers tried to mitigate some of the potential parts obsolescence issues associated with the Model 170. Instead of relying on the efficiency of assembly language programming, the ATC 2070 CU includes the necessary resources to execute programs written in high level programming languages such as ANSI C or C++. Such high-level language programs are more easily written and debugged, and are capable of being ported to other hardware platforms as necessary. The ATC 2070 also specifies the use of an O/S (OS-9 to separate the hardware from the application software). By specifying an O/S, the explicit mapping of User Memory and Field I/O, as was done with the Model 170, is no longer necessary. The O/S and associated standardized support functions take care of many of the basic execution management and scheduling tasks required by application software programs. The O/S further extends the hardware/software independence through I/O and memory resource sharing capabilities. These capabilities allow multiple independent applications to be run simultaneously on a single controller unit in a multi-tasking mode. This was not the case with a Model 170.

The ATC 2070 standard also provides for greater subcomponent interchangeability and modularity than the Model 170. ATC 2070 component modules are defined through specification such that they are interchangeable among different manufacturers.
With the Model 170 only the Modem/Communication and Memory modules are interchangeable among controllers produced by different manufacturers.

However, the ATC 2070 requires that a specific CPU chip and a specific commercial O/S be used. Unfortunately, the embedded hardware and O/S market place is not as large as is the PC marketplace. As a result, longevity concerns are surfacing for the ATC 2070 related to its particular O/S and CPU selections. Many users are concerned that additional retrofit and software porting costs would be required should either this O/S and/or the CPU become unavailable.

3.3 Functional Needs

ATC Engine Board manufacturers will provide a BSP that is compatible with their ATC hardware. Developers can port their software to various ATC controllers by compiling and linking their application with the appropriate drivers for the target controller as shown in Figure 3.1.

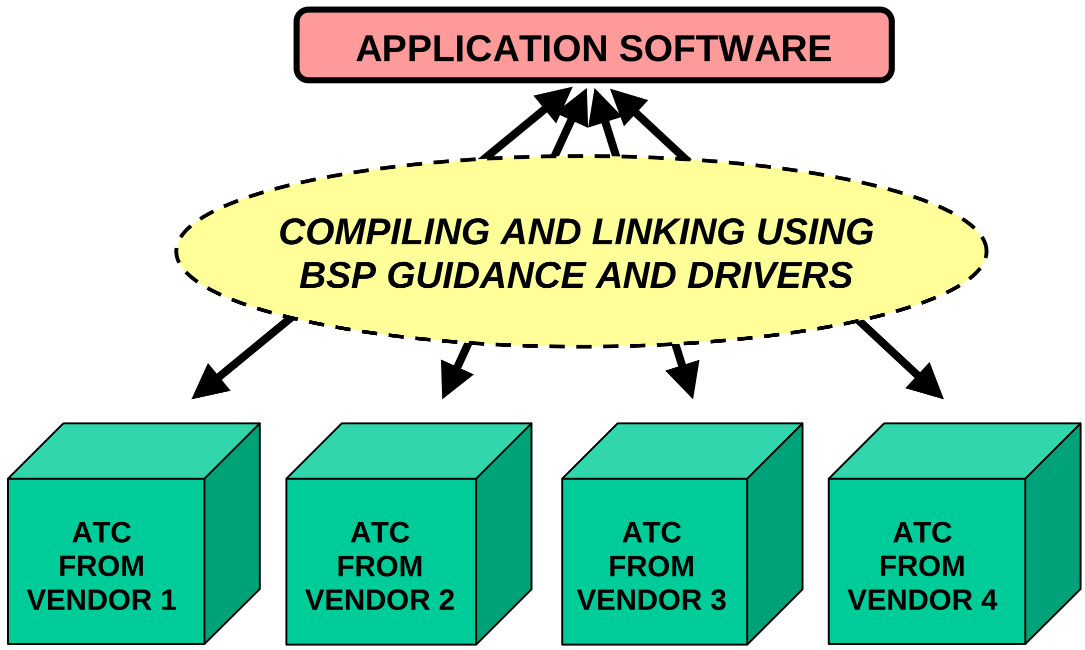

Figure 3-1: BSP in the ATC Architecture

Advanced communication capabilities are becoming increasingly important for ITS field controllers. ITS data communications networks are deploying NTCIP and Internet Protocol (IP) based data communications networks. Peer-to-peer networking capabilities are also increasingly required for advanced control algorithm implementations. For such networks, Ethernet is the connection interface of choice at field controllers.

3.4 Operational Environment

Typically, an operator interfaces to an ATC through one of three mechanisms:

Remote computer – this type of operation configures and manages ITS applications from a computer located at a traffic management location, such as a Transportation Management Center (TMC) or from a field located computer such as a traffic signal field master controller.

Local computer – this type of operation performs the same functions as a central computer does, but uses a portable interface device (e.g., laptop, PDA, etc.) connected directly to a port of the ATC.

Locally – this type of operation uses the front panel or portable interface devices (e.g., keyboard, displays, switches) at the ATC to perform the functions of configuring and managing the ITS applications.

The connection between the central computer and the ATC runs over a communications network. This can be either hard-wired (cables) or wireless. The network interface at the ATC can be either a serial communications port or Ethernet port. Figure 3-2 depicts the physical architecture of the key components related to a typical ATC based system run from a central location.

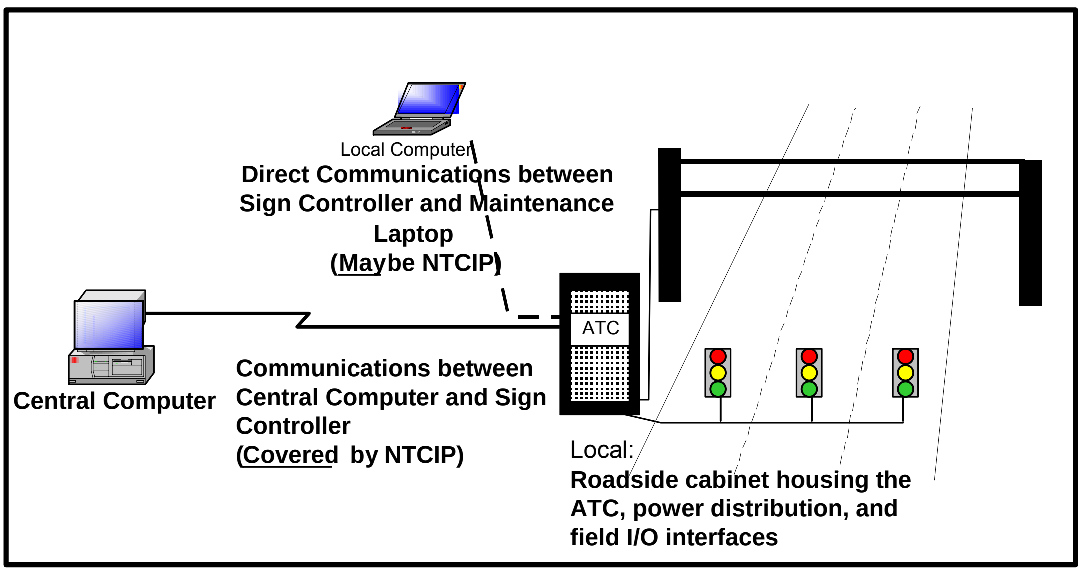

Figure 3-2: View of a Typical ATC System Environment

The ATC is enclosed in a field-located cabinet. The ATC connects to other cabinet-located input/output devices (i.e. load switches, detector sensors, etc.) through serial and or parallel connections. Cabinet input/output devices, in turn, connect to field-located elements (i.e. signal head, dynamic message sign, sensors, etc.).

In practice, there are additional components in a field-located cabinet which support the system including power distribution equipment, monitoring devices, and terminal facilities. The exact device interfaces and cabinet configuration depends on the particular ITS application and type of equipment being deployed.

As a minimum, the ATC must provide the necessary interfaces to support the ITS Cabinet standard. Additionally, the ATC should provide optional interface support for common legacy cabinets including Model 332, NEMA TS1, and NEMA TS2 types.

3.5 Representative Usage

As previously indicated, the functionality of a deployed ATC will depend on the applications software loaded into it. Typical ITS applications to be hosted on the ATC are listed in Table 3-1.

- Traffic Signal
- Traffic Surveillance
- Lane Use Signals
- Communications
- Field Masters
- Ramp Meter
- Variable/Dynamic Message Signs
- General ITS beacons
- CCTV Cameras
- Highway Rail Intersections
- Speed Monitoring
- Incident Management
- Highway Advisory Radio
- Freeway Lane Control
- High Occupancy Vehicle Systems
- Access Control
- Roadway Weather Information Systems
- Irrigation Control

Table 3-1: Anticipated ATC Applications

Due to its general-purpose nature, an ATC may be used for future ITS applications that are not currently anticipated. These expanded functions may, over time, expand the operational user needs for an ATC. Nonetheless, a number of basic operational usage scenarios can be discerned from present day applications.

This section identifies and describes some of the most common “use cases” to be supported by the ATC and its applications software. Figure 3-3 provides a top-level view of the operational features offered by a typical ITS application using an ATC. The definition of each feature is provided after the presentation of the diagram. The features in this diagram are subdivided into more detailed features in the text below. For these “use cases”, a more detailed “use case” feature diagram is presented along with corresponding definitions. Section 4 then uses these definitions to organize and define the various functional requirements of an ATC.

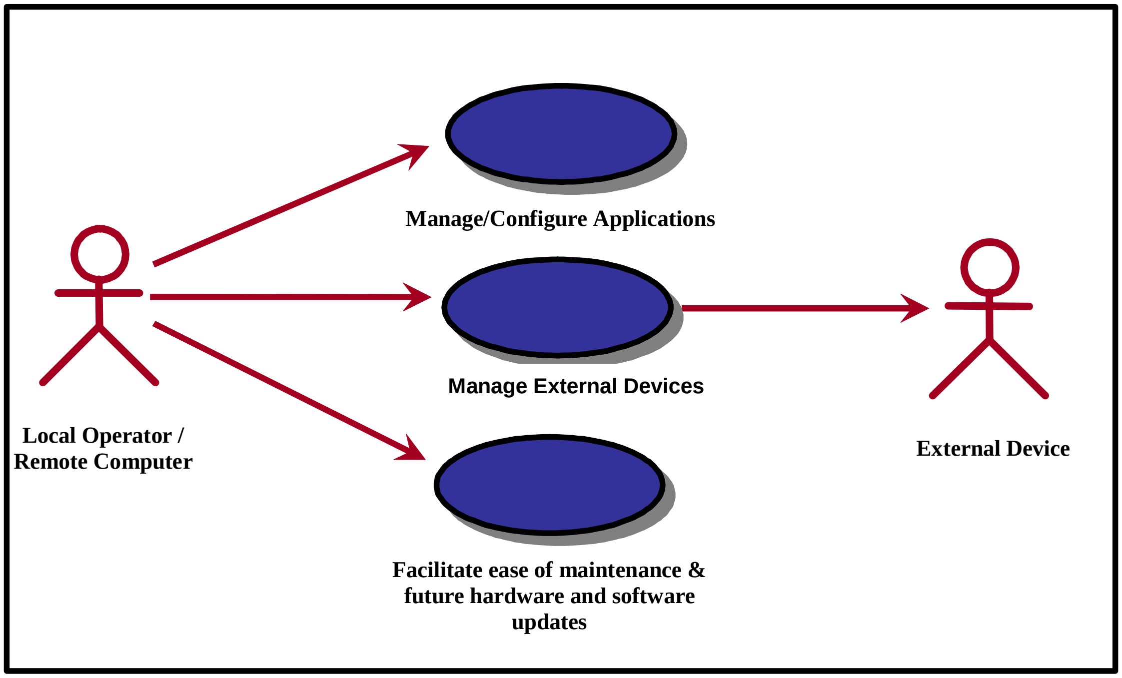

Figure 3-3: Main Maintenance/ Support Diagram

The generalized operational features of an ATC can be categorized into three major areas:

- Manage/Configure Applications
- Manage External Devices
- Facilitate ease of maintenance & future hardware and software updates

The Maintenance and Support function includes features for maintenance and update/enhancement of the controller unit's hardware and/or software.

3.5.1 Manage/Configure Controller Applications

The various sub-features for managing and configuring software applications are shown in the following figure. The subsequent sections detail these sub-features

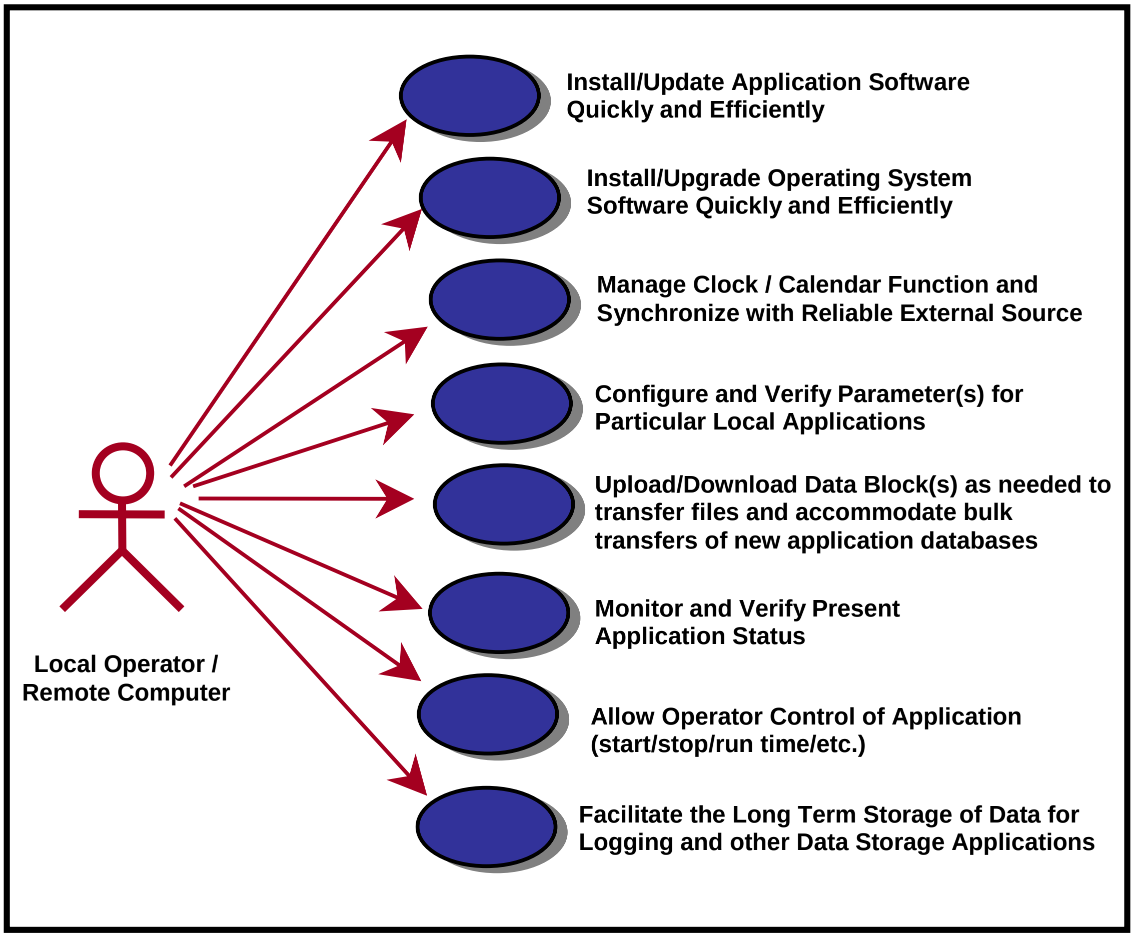

Figure 3-4: Manage/ Configure Applications' Sub-feature Areas

3.5.1.1 Install/Update Applications Software Quickly and Efficiently

This feature allows the local operator or a remote computer to install or update the application software resident on the ATC.

3.5.1.2 Install/Upgrade O/S Quickly and Efficiently

This feature allows the local operator to install or update the O/S resident on the ATC. Local upgrade capability is required while remote upgrade capability is considered an optional feature.

3.5.1.3 Manage Clock / Calendar Function and Synchronize with Reliable External Source

This feature is responsible for management of a real-time clock calendar function within the ATC. It allows the operator or a remote computer to interrogate and/or update the current time and date information kept by the ATC. It is responsible for synchronizing the ATC O/S clock to an AC power source or other suitable locally available reference to adjust for internal ATC clock drift.

3.5.1.4 Configure and Verify Parameters for Particular Local Applications

This feature allows the operator or a remote computer to manage and update the currently operational applications data stored in the ATC.

3.5.1.5 Upload/Download Data Block(s) as needed to Transfer Files and Accommodate Bulk Transfers of new Application Databases

This feature allows an operator to remotely or locally download or upload complete data blocks or data files from another computer device. It supports the operator's ability to do bulk transfers of complete application databases to and from the ATC.

3.5.1.6 Monitor and Verify Present Applications Status

This feature allows an operator to remotely or locally view real-time reports of current applications status. The feature, depending on the application, would allow the operator to view status indicators such as operating modes, failure status, event logs, operation algorithm outputs, input and output states, timer countdowns, etc.

3.5.1.7 Allow Operator Control of Application Execution

This feature allows the operator to manage the starting, stopping, and scheduling of one or more applications on the ATC.

3.5.1.8 Facilitate the Long Term Storage of Data for Logging and other Data Storage Applications

This feature facilitates the long-term storage of data for logging and other data storage applications

3.5.2 Manage External Devices

The various sub-features for "managing external devices" are shown in the following figure. The subsequent sections detail these sub-features.

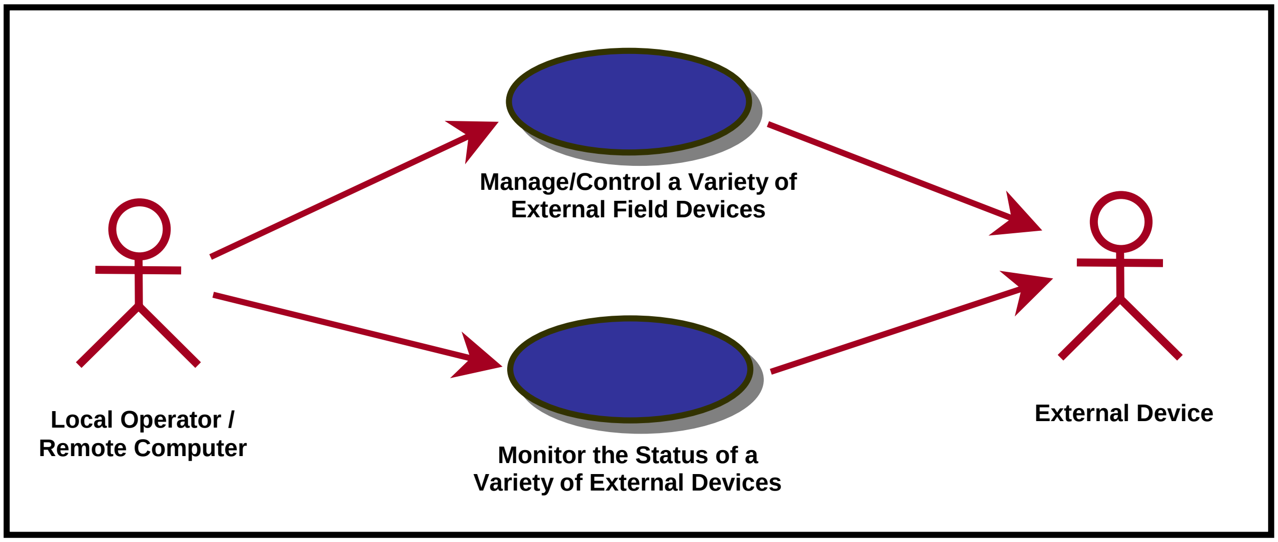

Figure 3-5: Manage External Devices' Sub-feature Areas

3.5.2.1 Manage/Control a Variety of External Field Devices

This feature addresses the need for external devices to be controlled remotely (through a local controller using commands from a central computer), locally (from a laptop computer connected to the controller), or from an unattended controller.

3.5.2.2 Monitor the Output and Status of a Variety of External Field Devices

This feature provides the capability for the controller to monitor device output and status and to use that status for local control configuration, failure diagnosis, logging and/or reporting to a local operator or remote computer.

3.5.3 Facilitate Ease of Maintenance & Future Hardware and Software Updates

The various sub-features for "facilitating ease of maintenance & future hardware and software" are shown in the following figure. The subsequent sections detail these sub-features.

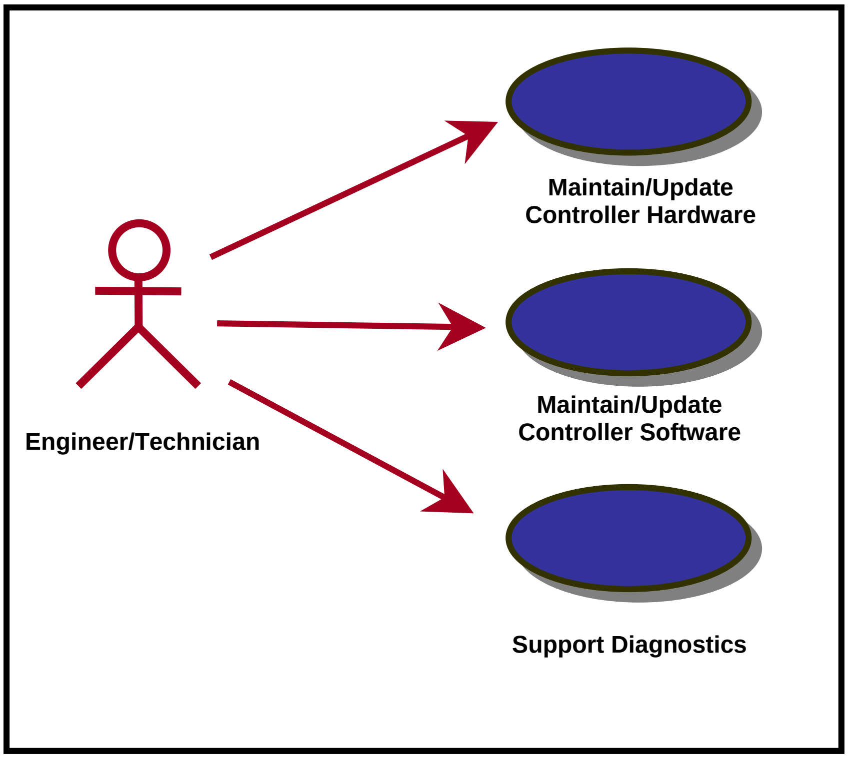

Figure 3-6: Facilitate Ease of Maintenance & Future Hardware and Software Updates' Sub-feature Areas

3.5.3.1 Maintain/Update Controller Hardware

This feature addresses the need for controller unit hardware to be maintained and updated as technology changes and additional functional and performance capabilities are needed.

3.5.3.2 Maintain/Update Controller Software

This feature addresses the need for controller applications software to easily be maintained, updated, or ported between different manufacturers' hardware units.

3.5.3.3 Support Diagnostics

This feature addresses the need for the controller to support diagnostic capabilities.

3.6 Security

The standard does not explicitly address security issues. However, network communication interfaces have been defined with provisions for data security in mind. If individual applications require it, security should be addressed either through the software hosted by the ATC or by physically protecting access to the ATC and its interfaces. These are outside the scope of this particular standard.

3.7 Modes of Operation

The features identified above were developed with the following three modes of operation in mind: standalone, direct, and distributed. Each of these is discussed below.

The "standalone" control mode assumes that the ATC is operating in the field without remote monitoring by a central computer or master controller. In this mode, application software is loaded into non-volatile controller memory and used to control and/or monitor externally connected devices such as gates, signals, beacons, signs, etc. Device control is based on locally stored schedule, predefined control algorithms or manual operation by a person present at the controller. Device monitoring might include processing of remote sensor inputs and/or monitoring the results of the controller's control actions. Under this mode, no communications is assumed to exist between the ATC and central computer or remote master. Local operator interactions take place through the ATC front panel interface, laptop computer, or similar portable device.

The "direct" control mode assumes that a remote control center or master device controls the external device(s) via commands to the ATC. In this mode, commands are sent from control center/master to the ATC via communications network to affect the operation of local device(s) connected to the ATC.

The "distributed" control mode is a combination of the first two. Here the local ATC applications software exercises normal control but the operation is managed and synchronized through a communication network connection with a central computer or master. Local control operations may frequently be overridden remotely to meet current needs and situations.