

ADAPTABLE SECURITY ARCHITECTURE FRAMEWORK (ADSEC): DESIGNING ADAPTABLE AND SECURE SCADA SYSTEMS USING THE NFR APPROACH

Supervisory control and data acquisition (SCADA) systems are used to control and monitor infrastructure such as industrial automated production systems, electricity transmission and distribution systems, oil and gas distribution systems, building facilities, and highway transportation systems. Typically SCADA systems have several remote terminal units (RTU's) and one master station that are connected by a communication media. RTU's monitor and control distributed elements in the system such as monitoring pressure or changing a valve setting on pipeline distributing oil from a refinery to a retail location; the master station receives status and transmits control data to all RTU's using a wired or wireless communication media. Among the more important properties of a SCADA system are adaptability and security. Adaptability refers to the ability of the system to adapt to changes in its environment while security refers to the ability of the system to defend against intrusions, both logical and physical, into the system.

SCADA systems usually have several levels of adaptability: at the lowest level the concern is whether the analog parameters are represented by voltages or currents; at the intermediate level the concern is which types of protocols need be supported such as Fieldbus, Modbus, or Ethernet; at higher levels the concern is whether wired or wireless devices need be used or should data portability between different systems be supported; and at the highest level the concern is the types of standards to comply including enterprise architecture standards and industry standard architectures. Likewise, security of SCADA systems is an important design issue since they are increasingly being used for controlling and monitoring critical infrastructures. A secure SCADA system prevents vulnerabilities in the system from being exploited by threats: authentication prevents illegal users from accessing the system, alarm systems prevent remote terminal units from being physically accessed and tampered with, and encryption ensures that data illegally intercepted is not easily deciphered.

Typical approaches to designing security in SCADA systems include using trusted vendor sources, following industry design guidelines, using industry standards such as ISA SP99 or ISA S99, or following a process established by researchers such as using security patterns. However, an important issue faced by SCADA designers is the tradeoff between adaptability and security since high adaptability and high security may not be achieved at the same time: if wireless is provided as the communication medium between the RTU and the master for ease of device replacement and upgrade, then security is compromised since wireless transmissions may be intercepted.

At CPSR, we plan to employ a goal-oriented approach to designing adaptable and secure SCADA systems; specifically, we plan to use the NFR Approach where NFR stands for non-functional requirements. In this approach, both adaptability and security are treated as goal that must be achieved during the process of SCADA design; these (soft-)goals are successively decomposed into their constituent NFR's (or softgoals) for the domain of interest. Then each design element including components, connection, patterns, and styles, is analyzed for the extent to which it satisfies the softgoals and how a combination of the design elements satisfy the softgoals as well. To aid this process the well-defined propagation rules of the NFR Framework are used. Those elements that in combination with others improve the overall design's adaptable security are selected for inclusion in the final design while those that hinder the goals are dropped from further consideration. In either case the justifications are neatly captured in a structure called the softgoal interdependency graph (SIG). The NFR

Approach we employ for this analysis is called the Adaptable Security Framework (ADSEC), a derivative of the NFR Framework, and is oriented towards designing adaptable and secure systems. The ADSEC Framework is discussed below.

THE ADAPTABLE SECURITY ARCHITECTURE FRAMEWORK (ADSEC)

The Adaptable Security Architecture Framework (ADSEC) is based on the NFR Framework [1] and the Adaptable Software Architecture Framework [2]. ADSEC treats both NFR's, adaptability and security, as goals to be achieved during the process of system architecture development. In order to analyze and evaluate the adaptable security of SCADA architectures the following iterative and interactive steps are required as per ADSEC:

1. Decompose NFR security for the domain
2. Decompose NFR adaptability for the domain
3. Decompose the architecture of the SCADA system
4. Determine the contributions made by the architectural decompositions to the NFR decompositions
5. Evaluate the overall adaptable security by applying the propagation rules of the NFR Framework
6. Extract the design rules for the SCADA system by analyzing the adaptable security for the SCADA architecture.

The first step decomposes the NFR security for the domain of interest (here, the domain of SCADA systems); the second step decomposes the NFR adaptability for the domain – these steps generate the NFR softgoal hierarchy for the domain where each NFR softgoal is represented by a cloud shape. The NFR softgoal hierarchy creates the AND-OR graph for the various NFR softgoals starting with NFR adaptable security at the root – in an AND-decomposition (which is represented by a single arc) all the child NFR softgoals need to be satisfied (ADSEC like the NFR Framework [1] and Adaptable Software Architecture Framework [2] uses the concept of satisficing where satisficing means good enough; NFR's cannot usually be satisfied absolutely and more often than not can be satisfied only within a range which is referred to as satisficing) for the parent NFR softgoal to be satisfied while in an OR-decomposition (which is represented by double arcs) the satisficing of even one of the child softgoals is sufficient for the satisficing of the parent NFR softgoal. Some of the NFR softgoals may be marked critical or high priority using '!' symbol next to the softgoal. The third step decomposes the system architecture for SCADA that is being evaluated – this generates the operationalization hierarchy for the architecture where each operationalizing softgoal is represented by a cloud shape with heavy-weighted borders. The fourth step determines the contributions that the operationalizing softgoals make to the various NFR softgoals – these can be of four types: strongly positive satisficing (also called MAKE contribution and is annotated by '++'), positively satisficing (HELP contribution annotated by '+'), negatively satisficing (HURT contribution annotated by '-'), and strongly negative contribution (BREAK contribution annotated by '- -'). The justifications for the various contributions are captured by claim softgoals which are represented by cloud shapes with dashed borders. The application of these steps 1 through 4

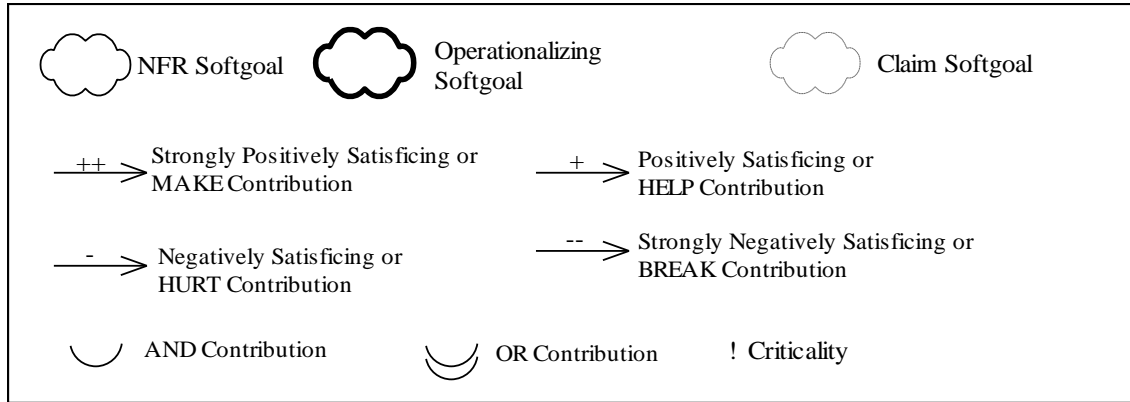


Figure 1. Ontology of ADSEC.

results in the development of the Softgoal Interdependency Graph or the SIG. The ontology of ADSEC is given in Figure 1. The fifth step of ADSEC is to evaluate the adaptable security for the SCADA system design using the propagation rules of ADSEC; the propagation rules are (in the following TYPE is used to stand for one of MAKE, HELP, HURT, or BREAK):

- R1. If all the contributions received by a leaf NFR softgoal are TYPE then that leaf NFR softgoal is considered TYPE-satisficed.
- R2. If a leaf NFR softgoal receives at least one HELP contribution then that leaf NFR softgoal is HELP-satisficed.
- R3. If a leaf NFR softgoal receives at least one HURT contribution then that leaf NFR softgoal is HURT-satisficed.
- R4. If a leaf NFR softgoal receives at least one BREAK contribution then that leaf NFR softgoal is BREAK-satisficed.
- R5. If R2, R3, and R4 apply, then the tie is broken in the order R4, R3, and then R2.
- R6. If a leaf NFR softgoal does not receive a contribution then it is considered MAKE-satisficed if it is involved in an AND or EQUAL relations, and BREAK-satisficed if it is involved in an OR relation.
- R7. In the case of AND-related factors, if all child factors are TYPE-satisficed then the parent NFR softgoal is TYPE-satisficed.
- R8. In the case of AND-related factors, if even one of the child factors is TYPE-satisficed then the parent NFR softgoal is TYPE-satisficed; the priority decreasing in the order: BREAK > HURT > HELP > MAKE.
- R9. In the case of OR-related factors, if all child factors are TYPE-satisficed then the parent NFR softgoal is TYPE-satisficed.
- R10. In the case of OR-related factors, if even one of the child factors is TYPE-satisficed then the parent NFR softgoal is TYPE-satisficed; the priority decreasing in the order: MAKE > HELP > HURT > BREAK
- R11. In the case of EQUAL-related factors (only one child) the parent is TYPE-satisficed if the child is TYPE-satisficed.

After applying these propagation rules, if the basic NFR softgoal (either security or adaptability) is satisficed, then that architecture possesses that NFR, that is, that architecture is either secure or adaptable. However, upon applying these propagation rules one finds the basic NFR softgoals denied, then that architecture does not possess that NFR. Since we will be considered both security and adaptability NFR's at the same time for any SCADA architecture, an architecture can satisfice NFR's as per Table 1, that is, the architecture may be neither

adaptable nor secure, the architecture may satisfy only one of the NFR's, the architecture may be both adaptable and secure, or the architecture may satisfy one of these NFRs to a certain extent. In Table 1, 'no'

Table 1. Adaptable Security Possibilities for SCADA System Architecture.

Adaptability	Security	Inference
no	no	architecture neither adaptable nor secure
no	yes	architecture is not adaptable but secure
no	somewhat	architecture is not adaptable and not fully secure nor insecure
yes	no	architecture is adaptable but not secure
yes	yes	architecture is both adaptable and secure
yes	somewhat	architecture is adaptable but not fully secure nor insecure
somewhat	no	architecture is not secure and not fully adaptable or inadaptable
somewhat	yes	architecture is secure but not fully adaptable or inadaptable

stands for BREAK-satisficing of a basic NFR softgoal, 'yes' stands for MAKE-satisficing of a basic NFR softgoal, while 'somewhat' stands for either HURT or HELP-satisficing of a basic NFR softgoal – if it is HURT then the negative qualification dominates (either insecure or not adaptable – which we also refer to as 'inadaptable'); if it is HELP then the positive qualification dominates (either secure or adaptable).

The final step of ADSEC is the design rules extraction step. This step can be used to for determining one or more of the following for a given SCADA architecture:

1. Evaluation of adaptable security: determine to what degree an architecture is adaptably secure
2. Analysis of adaptable security: analyze why or why not an architecture is adaptable to a particular degree
3. Prediction of strengths and weaknesses in adaptable security: ADSEC can indicate strong points and the weak points in a SCADA architecture with respect to adaptable security
4. Extraction of design rules that need to be embedded in the SCADA system being developed so that the final implementation is adaptably secure.

In order to aid the final step the ordering among the contributions needs to be understood – the contributions are ordered as

MAKE > HELP > HURT > BREAK.

Therefore, any improvement in satisficing will mean going toward a higher (or better) contribution.

An example SIG is shown in Figure 2.

References

1. Chung, L., Nixon, B. A., Yu, E., and Mylopoulos, J. Non-Functional Requirements in Software Engineering. Kluwer Academic Publishers, Boston, 2000.
2. Subramanian, N., and Chung, L.: Supporting the Development of Adaptable and Secure Software Systems: An NFR Approach. In: Proceedings of the International Conference on Software Engineering Research and Practice, Las Vegas, pp. 108 – 114 (2005).

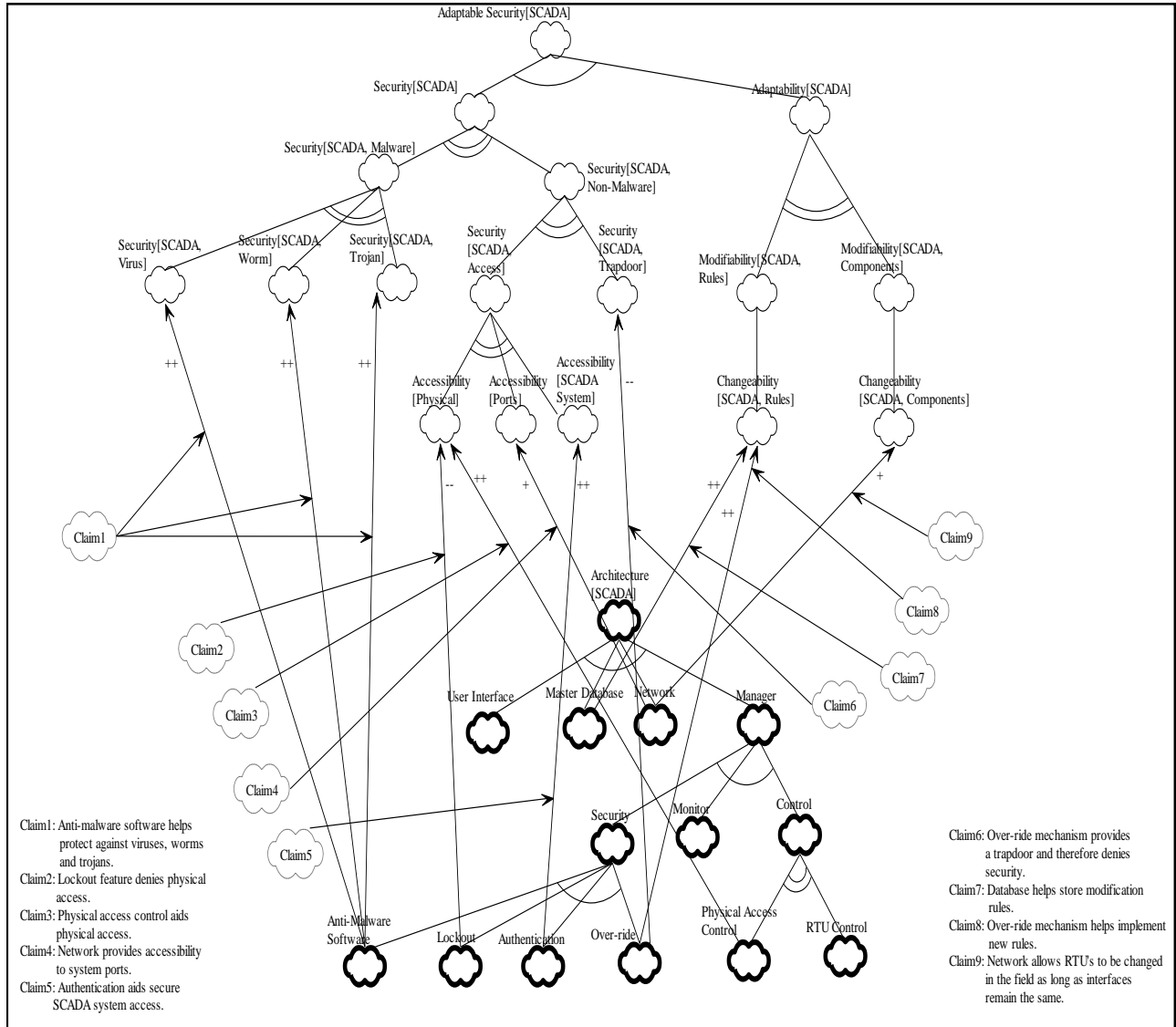


Figure 2. Example SIG for ADSEC Application to SCADA System