Paper presented at the 2011 IEEE System, Men, and Cybernetics (SMC) Conference. Anchorage, AK. October 10-12

HMI Aspects of Automotive Climate Control Systems

Asaf Degani[1] *& Andrew Gellatly*[2]
General Motors Corporation
[1]Advanced Technical Center-Israel. Herzliyya
asaf.degani@gm.com
[2]Global Product Development, Warren, MI, USA
andrew.gellatly@gm.com

Michael Heymann
Department of Computer Science
Technion, Israel Institute of Technology
Haifa, Israel
heymann@cs.technion.ac.il

***Abstract*—In this paper we discuss a formal approach to the design and analysis of automotive systems, from a human-machine interaction (HMI) point of view. Specifically, we detail the behavior of a generic climate control system, present a statecharts model of this system, and discuss aspects of user interaction analysis. Several general principles for the design of climate control systems are illustrated and discussed. The topic of design patterns, in the context of a formal description of user interaction, is introduced, and two design patterns are illustrated and discussed.**

Keywords: Human Factors; Human-Machine Interaction, Automotive System; Formal Methods; Statecharts; Design Patterns.

I. INTRODUCTION

The goal of the research described in this paper is to develop a formal process for the design of human-machine interaction. There are a number of benefits to be derived from such a design approach: (1) obtaining a clear description of the (interaction) design so as to enable review and discussion among multidisciplinary teams, (2) articulating overarching design principles that the design must adhere to as well as generic design patterns, (3) employing a formal description for specifications, (4) establishing a platform for analysis, heuristic or otherwise, of the design, and (5) informing and supporting the design of the graphical user interface (e.g., screen layout). Thus, the overall intent is to provide a formal approach to the design of human-machine interaction, in the context of automotive systems, with the goal of improving not just the design but also the quality and rigor of the specifications. By quality we mean that the description is detailed and leaves nothing to interpretation or possible ambiguity. By rigor we mean that all system events and transitions are accounted for and described in the specifications [10].

In this paper we focus our attention on the first and second objectives (obtaining a clear description of the interaction, and articulation of overarching design principles and patterns) in the context of the third (employing a formal description for specifications). Here we use the statecharts language [6, 7, 8] as a basis for describing the system, and add a few semantic and syntactic elements to better represent user interaction features. (Other modeling formalisms could also be employed for this purpose.) By creating a formal model of the system, the foundations for an analytical platform are laid down, having the advantage of identifying design deficiencies early on in the design process [10]. Special emphasis is placed on identifying potential problem areas in the proposed system design. There is also the additional benefit of finding potential areas for simplification and enhancement of user interaction. Although beyond the scope of this paper, a formal approach also provides the basis for graphical user interface design and improved screen layout and overall information organization. To illustrate the proposed approach to description and analysis (as well as the discussion of some basic design principles), we describe here a typical climate control system found in a modern automobile. The system includes multi-zone temperature control and several levels of automated control.

The paper is organized as follows. In Section II, we discuss relevant design principles in light of this climate control system. In Section III we describe the general operation of this climate control system and discuss some of its design features. A behavioral analysis of the system, with emphasis on user interaction and experience, is provided in Section IV, and an introduction to design patterns and their applicability to various user interaction contexts is provided in Section V. We conclude with several observations on the utility of such a formal approach for human-machine interaction design and specification writing.

II. DESIGN PRINCIPLES FOR CLIMATE CONTROL SYSTEMS

Most automotive climate control systems are located in the cockpit's center stack. They are accessible to both the driver and the front seat passenger. Some systems have dual zone temperature control so that the side passenger can select his or her own desired temperature. The interface for a typical climate control system consists of a fixed-segment or LCD panel surrounded by several push buttons.

Modern climate control systems can be operated either in fully automatic mode, in a variety of automatic and manual mode combinations, or in fully manual mode [4, 5]. Climate control systems are composed of several components, all working in parallel. These include the fan unit that circulates air inside the vehicle, the air-conditioning compressor, the air-distribution setting that delivers air to selected combinations of vents located in the floor, panel and windshield, the air-source unit that can be set to either fresh (outside) air or recirculated air, the temperature control and finally, the zone control that allows individual temperature settings (whenever the system has multiple individually controlled temperature regions).

In most modern cars, the preferred operating mode is the fully automatic mode, where the only user selectable settings are the zone selection (single or dual), and the respective temperature settings. In *auto* mode, the system engages the compressor, sets the fan speed and the air delivery mode automatically. In many systems, the air source is not part of the automatic control system and is selectable by the driver at will. In some systems, the air source is under automatic control as determined by inputs from air quality sensors in the vehicle, and designer tendency is to always default to fresh air.

The user, however, can override the fully automatic mode by taking manual control over any or all four automatic components (compressor, fan, air source, and air delivery). The driver can shut down the compressor and turn it on again when desired. The driver can also set the fan speed control by pressing the respective increase or decrease button, at which point the fan speed switches to manual control. Similarly, the user can manually override the automatic air delivery mode control by pressing the respective "mode" buttons, thus switching the latter to manual control. Thus the user can change the system to manual control component by component. Those components that are not overridden by the driver remain under automatic control.

One immediate observation about these systems, however, is their potential complexity that can arise from interactive operation of parallel components. While most low-end system designs have few if any parallel interactions, high-end systems may at times exhibit highly interactive behaviors, where the operation of one component triggers or is inhibited by the settings or mode changes of other (parallel) components. While such complex interactions may substantially enhance the operation of the system, they may also be a source of significant user confusion and potential ambiguity. This issue will be elaborated on later. To explicitly identify and highlight the interactive behaviors, our modeling formalism contains special features and notations as will be discussed and used in Section III.

Consider, for example, a system where the air source is made part of the automatic control system. In this execution, the designer preference is to operate the system under the fresh air setting. This situation can lead to a number of driver-designer conflict scenarios. For instance, whenever the engine is started, the air source defaults to the fresh air setting. A driver who prefers the recirculation mode must then reselect it every time the ignition is turned on. The recirculation mode is disabled and inaccessible as a function of the setting of the air delivery mode. Thus the driver is prevented from manually overriding these designer choices. Such enforcement may also be carried over to the air conditioner setting. For example, when in the recirculating air mode, the compressor is always turned on. Here again the driver cannot directly override this enforcement.

Users commonly perceive manual operation as a framework that enables them to configure every possible setting at will. Thus, many are surprised when the designers' choices inhibit their selection, when, in fact, they believe they are operating the system manually. In other words, drivers assume that manual is the equivalent of no enforcement of designer priorities (not even as defaults). Thus, in an ideal system (from the user's point of view), the air source, air delivery mode, a/c settings and fan speed should all be user-selectable without constraints. The design principles that can be derived from this observation are: (1) *under manual operation, do not inhibit driver choices, unless it is a safety imperative;* (2) *in this condition, when the driver's request is blocked, provide explicit and positive indications of this effect.*

Typically, users assume that when overriding a specific *auto* mode component and its setting, unaffected components will remain under automatic control (this is true for medical systems and flight control systems – see [3]. Similarly, in the context of climate control systems, when the fan speed setting is overridden, the compressor and air delivery modes should remain under automatic control (unless the driver intervenes explicitly). Thus the suggested design principle is: (3) *don't change what the driver has not explicitly asked for*.

III. A GENERIC CLIMATE CONTROL SYSTEM

In this section we describe the operation of a climate control system that exhibits both "simple" and "sophisticated" operating features. We focus on its behavior and user interaction aspects (the system's formal statechart description is shown in Figure 1). Several special notations and designations are used in our model description formalism. We use broken lines to denote "automatic" transitions that are triggered either by internal dynamics or as side effects of other transitions. We use color-coding to label transitions of special types. These include *concurrent*, *conditional*, or *guarded* transitions, all of which imply complex interactions between system components. We use the enclosed symbol (I) to denote the concept of *inheritance*. That is, the transfer of parameters and other relevant data that prevailed prior to the state change. We use the notation (I+) or (I-) to denote a single positive or negative shift of the inherited value.

The climate control system has a power-off state where it is shut down. In this state the fan is turned off and the air conditioning compressor is inactive. Nevertheless, temperature is still maintained to the extent possible, with fresh air and the aid of the heating system. The air distribution vents also remain operational. Naturally, the system is completely inoperative when the ignition is turned off. As seen in Figure 1, there are five parallel components to this system:

A. *Fan Unit*

The fan unit has six speed levels (1-6). In manual operations, the system always remembers the last fan speed setting (history) that existed prior to ignition-off or power-off, and returns to this setting when the power or ignition is turned back on.

B. *Compressor Unit*

In the fully automatic (*auto*) mode, the compressor unit is always on. It can be turned *off* manually. Nevertheless, the compressor automatically turns on when the air-distribution mode is set to *DEFOG* or *DEFROST,* or when the air source is in the recirculating-air mode.

C. *Zone*

The system is a dual zone unit that allows the front seat passenger to set his or her own temperature. When in *DEFROST* mode, this dual control is eliminated (only a single zone is provided) and both temperature controllers can change this single setting, until *DEFROST* is disengaged.

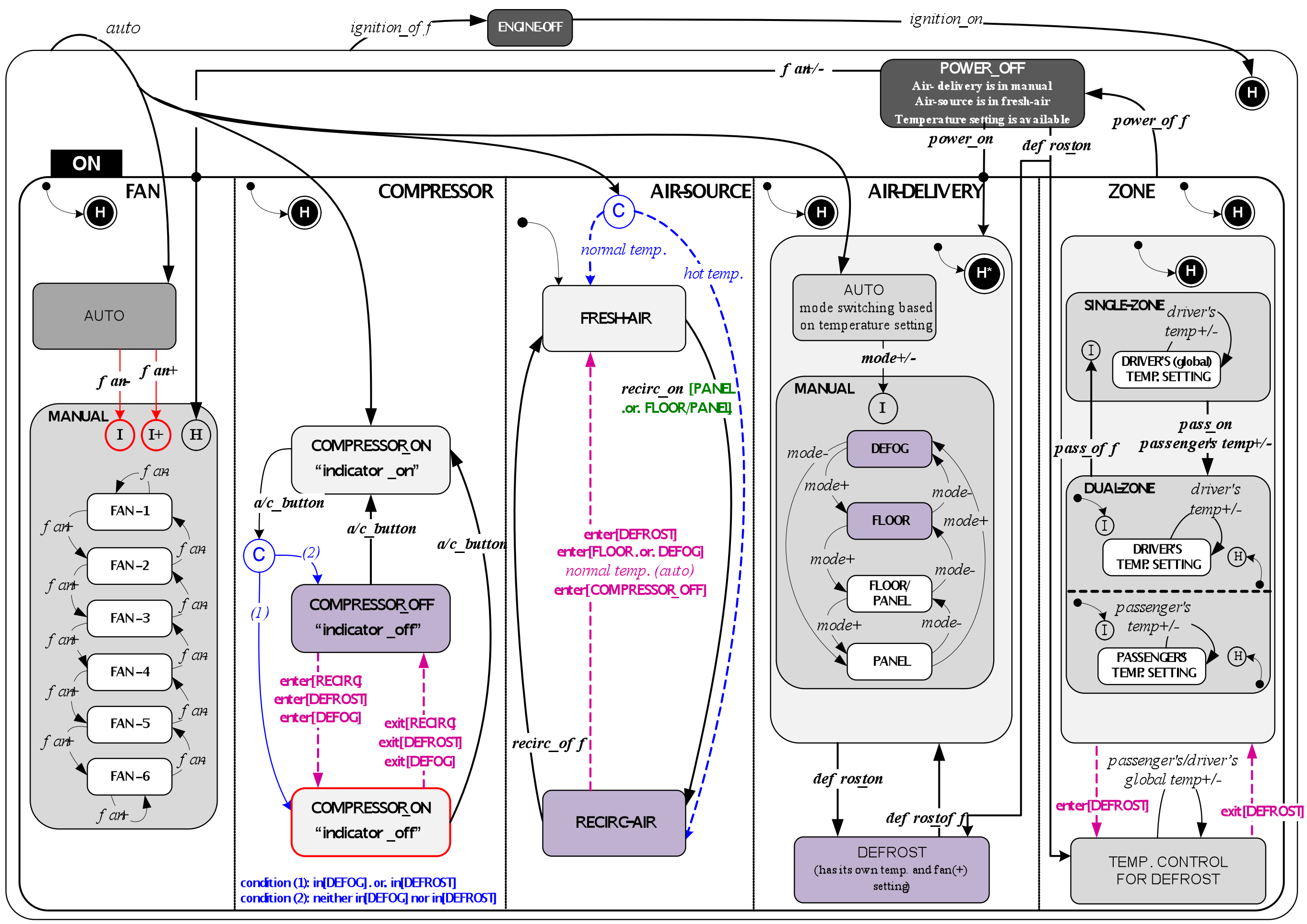

Figure 1. Statechart model of the climate control system. Broken lines (colored in magenta) denote *automatic* transitions that are triggered either by internal dynamics or as side-effects. Side effects occur when a given transition triggers another action (e.g., event) elsewhere in the system, or when the system enters or exits a specific state (also colored in magenta). Conditional transitions are colored in blue (as well as the condition itself). In situations where a transition is guarded (i.e., only when the condition inside the block brackets is true, will the transition take effect) we color the "guard" in green. Inconsistencies and discrepancies in the way the system behaves are marked in red.

D. Air Delivery

The air delivery modes include the following air flow possibilities: *FLOOR*, *FLOOR+PANEL*, *PANEL*, *FLOOR+WINDSHIELD (defog),* and *WINDSHIELD ONLY (defrost)*. In the *auto* mode, the air delivery is selected automatically depending on the desired temperature. (The general idea is that hot air is delivered from the lower vents and cold air from the upper ones, and as a consequence the air delivery setting is dynamic). Selection of air delivery modes can also be performed manually. There are, however, built-in operational constraints that prevent entry into *recirculating* air while in *FLOOR* and *FLOOR+ WINDSHIELD (defog)* modes. The *defrost* setting is an operational mode of its own, in which the air-conditioning compressor is engaged, the fan speed is set to a higher level than was previously selected, and the air source is set to fresh air. While in *defrost* mode, the dual zone temperature control is inhibited and a single temperature regime is enforced until this mode is exited.

E. Air Source

The default setting of the air source is *fresh-air*. The *recirculation* mode is never entered automatically and can be entered manually only if the air delivery mode settings permit. When these conditions are violated, the system will exit the *recirculation* mode automatically.

IV. Analysis of the Design

In this section we discuss some of the features of the climate control system described above, with special emphasis on its design, interaction style, and overall design principles.

A. Transition from Automatic to Manual

When the system is operational, the automatic mode is entered by pressing the *auto* button. The *auto* mode is exited when either the fan speed or the air delivery mode is changed by the driver. The transition from automatic to manual operation, however, takes place in steps. Specifically, only those settings that the driver overrides are released to manual control whereas the others remain under automatic operation unless the driver intervenes. This design approach reflects principle (3) mentioned earlier: *don't change what the driver has not explicitly asked for*.

B. *Complex Interactions*

A key issue in the design of successful automotive human-machine interaction systems is simplicity of operation. The *auto* mode of the climate control system achieves simplicity with efficiency. A single press of a button activates this operation, and only the desired temperature must be selected. Complex interactions between components (e.g., automatic change of the fan-speed and the air delivery mode based on the current temperature, and defaulting the air source to *fresh air* based on the air delivery mode) to achieve optimal operation are performed by the automation without any driver intervention.

Under manual operation, the enforcement of interactive conditions comes inevitably at the expense of operational simplicity. Thus, while potentially providing better performance and overall efficiency, an increase in interaction complexity ensues. There is an unavoidable tradeoff between designer priorities and handling tractability. Enforcement of lockouts and side effects that stem from the designer's priorities may be inconsistent with the driver's preferences. In situations where the driver assumes that he or she is in full control of the system, under certain conditions, this is, in fact, no longer the case. Below are several examples of such complex interactions in the generic system described in Figure 1:

- It is not possible to manually switch to the *recirculating-air* mode when the air delivery setting is *FLOOR*, *DEFOG* or *DEFROST*.
- It is not possible to manually turn off the compressor when in the *DEFOG* or *DEFROST* mode.
- Although the system automatically switches the air source from *recirculating-air* to *fresh-air* when entering the *FLOOR* or *DEFOG* mode, it does not automatically switch itself back to *recirculating-air* when exiting these modes, in spite of the driver's preference. (Thus, when exiting the *FLOOR* or *DEFOG* mode, recirculation must be re-selected manually.)
- If the compressor is manually turned off while the air source is set to *recirculating-air*, the system automatically reverts to *fresh-air*.

Design priorities that are withheld from the user tend to create confusion and user frustration. For example, in the system described in Figure 1, there is a strong design priority to have the system operate with the air source set to *fresh air*. The priority is so strong that the system never transitions to recirculating mode automatically. In particular, while switching out of recirculating air automatically (when operational conditions demand), the system never returns to recirculating air.

The behaviors described above constitute a violation of design principle (1): *Under manual operation, do not inhibit driver choices, unless it is a safety imperative*.

C. *Unpredictable Responses*

In user interactive systems, especially of the safety-critical kind, the user counts on being able to predict the system's response to his or her interaction [5, 14]. This expectation, by the way, is unrelated to the user's satisfaction or dissatisfaction with the designer's priorities; it relates to the predictability of the system given its behavior, interface indications, and prior knowledge.

Sometimes, in order to simplify interfaces, information concerning the current state or mode of the system is not displayed. Specifically, when in automatic operation, the air delivery modes (PANEL, FLOOR+PANEL, FLOOR, DEFOG) might not be displayed to the driver. The design philosophy behind this is that *when under full automatic control, there is no need to burden the user with the underlying automatic mode changes.* If the driver is dissatisfied with the current setting of, say, the air delivery mode, this setting should appear in the display when the respective button is pressed. However, when the system is in semi-automatic control, complex interactions may come to haunt the driver in the sense that the user might not be able to predict side-effects and automation inhibitions. Specifically, if the current setting is PANEL or FLOOR+PANEL, pressing the recirculating button will switch the air source mode accordingly. However, if the setting is FLOOR or DEFOG, the system will not respond to the driver's request. In this case, even a knowledgeable user can understand the system's response only retrospectively.

V. DESIGN PATTERNS

By *design patterns* we refer to a standardized framework for the solution of specific design problems. A pattern describes a "good" solution to a common problem within a specific context, with the idea that the solution can be used over and over. Patterns are thus reusable and lead to ease of design and consistency in aiding users to interact in a predictable manner with the system. The general concept of design patterns was formulated by the architect Christopher Alexander [1] and later adopted by computer scientists. Design patterns gained further popularity in computer science after the book *Design Patterns: Elements of Reusable Object-Oriented Software* [2] was published. Design patterns are typically used today in computer science to build software systems and to communicate designs (for example in preliminary design reviews). Patterns were first suggested as an approach to interaction design by Norman and Draper [11]. What is appealing about patterns in the context of interaction design is their potential generality across the entire (infotainment, e.g.) system, as well as the ease by which they can be translated into specific software solutions and specifications [15].

Figure 2 depicts a pattern for increasing and decreasing levels (e.g. fan speed). It is activated by two buttons, for *up* and *down* (labeled "+" and "–" in this case). Pressing the up (+) button scrolls the fan speed up until it reaches the highest level (in this case level 6). From there, any subsequent pressing on the up button does not lead to any change. The same applies for down ("–"). This pattern works well when the levels are on an *ordinal* scale [13]; that is, where there is an ordered difference between levels (speeds, light intensity, volume settings, etc.).

Generally speaking, this pattern is useful in situations when the user can see or feel the current state of the system so that any pressing on the up ("+") or down ("–") button will not only change the level accordingly but will also wake up the system. The pattern should be somewhat different when the user *cannot* see or feel the current state of the system. In such a case, pressing the button should wake the system to manual operation at the current state (but not change it).

Figure 3 depicts a pattern for switching between modes. The relation is commonly nominal, as no real physical order exists. In this respect, note that there might be no need to provide two separate (up and down or "+" and "–") buttons; one would suffice just as well.

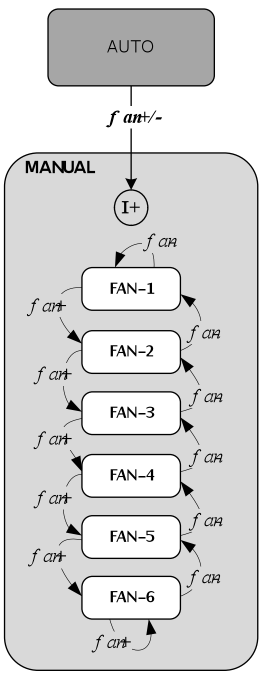

Figure 2. *Ordinal values.*

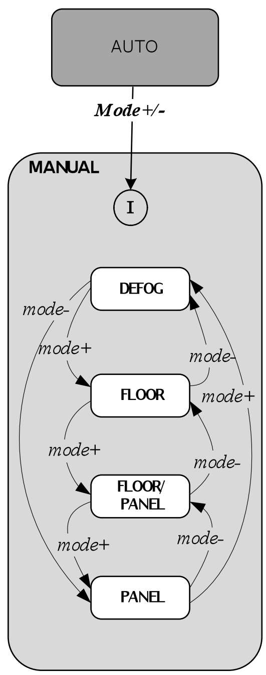

Figure 3. *Nominal values.*

The pattern of Figure 3 is sometimes used in climate control systems. Nevertheless, the use of this particular pattern can be problematic. First, although the different air delivery modes are organized in a quasi-ordinal way, and the arrows on the display give a feeling of ordinal scaling, it is not clear whether non-technical savvy users would recognize this organization and the functionality of the up and down buttons. As such, the circularity contradicts the implied ordinality to some extent. The pattern is very appropriate to actual behavior and interface design in the transition from *auto*. Here, because it is quite difficult to determine the actual mode of the system when the display is blank (in *auto*), waking up the system (and not switching the current mode) is a reasonable design solution.

Another pattern (not shown here) that emerges from the system and the model is the use of *auto* to activate automatic modes in the components, as well to activate default states in components that do not have automatic modes (compressor, air source, zone). Although such a pattern is clear with respect to activation, release from automatic control is not trivial and deserves considerable attention.

To conclude, patterns can be quite useful structures in user interaction design. It is important, however, to create not only a repository of patterns but also to define criteria for pattern selection, especially in light of the overall design philosophy, principles and specific sub-system designs (e.g., climate control systems). An important facet of behavioral pattern design is how it relates to the graphical user interface (or GUI) solution used. In this context, note that in this paper we did not discuss at all the necessary correspondence between the behavioral description of the system and the design of the interface per se. This issue deserves specific attention and is beyond the scope of the current paper (but see [4] for an initial approach).

VI. SUMMARY AND CONCLUSIONS

The introduction of interaction applications and automation features into modern vehicles is moving at a fast pace, and is only bound to accelerate in the coming years. As such, the design processes of human-machine interaction are becoming more and more important in automotive systems, especially in light of future plans for autonomous driving. Comprehensive and rigorous methods and tools are urgently needed to make such systems not only safe, effective, and efficient, but also elegant to interact with. While many of the processes to create user interaction designs are artistic in nature, the process of transforming an initial design into a real product can be made more rigorous, systematic, and definitely more comprehensive.

This paper describes a behavioral analysis of an automotive human-machine system. Using a typical climate control system that incorporates several design philosophies and interaction styles, we formulate several observations about its operational design. The paper uses the statecharts language to describe the system and to illustrate the particularities of the interaction. We argue that a formal description of user interaction is an essential ingredient for correct design. It also provides a foundation for formal analysis and for automatic testing and verification. We introduce the notion of behavioral design patterns for creating a consistent design (and improved implementation of both the design and the software engineering). One of the main problems in advancing the ideas of reusable design patterns in user interface design is the lack of a "good" representation. Here we suggest a formal representation, statecharts, as a way to capture behavioral design patterns, and then illustrate two patterns, along with some criteria for their selection. Based on our observations about the behavior of the climate control system, we discuss and suggest general design principles of user interaction that can be incorporated into future designs to respond to inconsistencies and unmet behavioral expectations.

This framework of a formal description, design patterns, and design principles opens the door for a two-pronged approach for the design and analysis of human-machine interaction. This approach is comprised of a top-down component where an overall design philosophy and policies provide a general design direction and a bottom-up component where the details are worked out. Formal methods can then be used to evaluate not only the correctness, consistency, and completeness of the design but also the degree to which the actual design adheres to design principles.

VII. References

[1] Alexander, C. (1964). *Notes on the Synthesis of Form*. Cambridge, MA: Harvard University Press.

[2] Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). *Design Patterns: Elements of Reusable Object-Oriented Software*. New York: Addison-Wesley.

[3] Degani, A. (2004). *Taming HAL: Designing Interfaces Beyond 2001*. New York: St. Martin's Press (Palgrave-Macmillan).

[4] Degani, A. & Gellatly, A. (in press). Modeling Human-Machine Interactions Using Statecharts: One hot and cool example. *Proceedings of the SAE 2011 World Congress*. Detroit, MI.

[5] Degani, A., & Heymann, M. (2002). Formal Verification of Human-Automation Interaction. *Human Factors, 44*(1), 28–43.

[6] Harel, D., & Politi, M. (1998). *Modeling Reactive Systems with Statecharts: The STATEMATE Approach*. New York: McGraw-Hill.

[7] Harel, D. (1987). Statecharts: A Visual Formalism for Complex Systems. *Science of Computer Programming, 8* (pp. 231-274).

[8] Harel, D. (2009). Statecharts in the Making: A Personal Account, *Communications of the ACM, 52*(3).

[9] Heymann, M. & Degani, A. (2007). Formal Analysis and Automatic Generation of User Interfaces: Approach, Methodology, and an Algorithm. *Human Factors, 49*(2), pp. 311-330.

[10] Leveson, N. (1995). *Safeware: System Safety and Computers*. New York: Addison-Wesley.

[11] Norman, D. & Draper, S. (Eds.) (1986). *User Centered System Design*. Hillsdale, NJ: Lawrence Erlbaum Associates.

[12] Simon, H. A. (1969/1981). *The Sciences of the Artificial* (2nd ed.). Cambridge, MA: MIT Press.

[13] Stevens, S. S. (1946). "On the Theory of Scales of Measurement". *Science, 103*(2684) (pp. 677–680).

[14] Woods, D., Sarter, N., & Billings, C. (1997). Automation Surprises. In G. Salvendy (Ed.), *Handbook of Human Factors and Ergonomics* (pp. 1926-1943). New York: John Wiley.

[15] Van Welie, M., Van Der Veer, G. & Eliens, A. (2000). Patterns as Tools for User Interface Design. In Vanderdonckt J. & Farenc C. (Eds.), *Tools for Working with Guidelines*. London, UK: Springer.