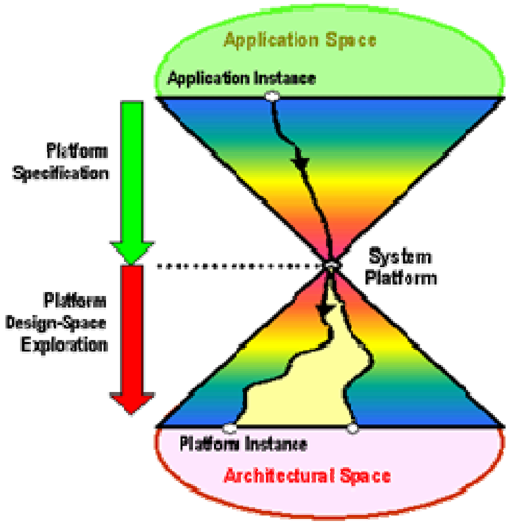
Platform-Based Design
# of Augmented Cognition Systems



Latosha Marshall & Colby Raley

ENSE 623
December 7, 2004

Table of Contents

Table of Figures

Introduction

Platform-based design principles can be used to allow faster and more efficient creation of augmented cognition systems from customer specifications. A modular design enables a higher product flexibility and reduction of development time, parallel development of system components, reduction of production time, reduced capital investment in production, reduced material and purchase costs, improved quality, easier service and upgrading, and easier administration of any system (Ericsson, 1999). Platform-based design takes that one step further, and allows systems to be fully developed, designed, validated, and verified while still at high levels of abstraction in the design process. It bridges the gap between custom-designing a new system for each platform instantiation and attempting to merely recycle system components in radically different platforms.

Platform-based design is important in almost any high-technology system, but is particularly relevant to augmented cognition systems for two major reasons:

- Technologies required to implement augmented cognition systems are still being developed, and will continue to improve dramatically over the coming years - for example, sensing technologies will never catch up with the knowledge of architecture design, so a system should be able to accommodate this.
- Augmented cognition systems will continuously be applied to new scenarios and situations; therefore, the systems should be able to adapt to any necessary environment.

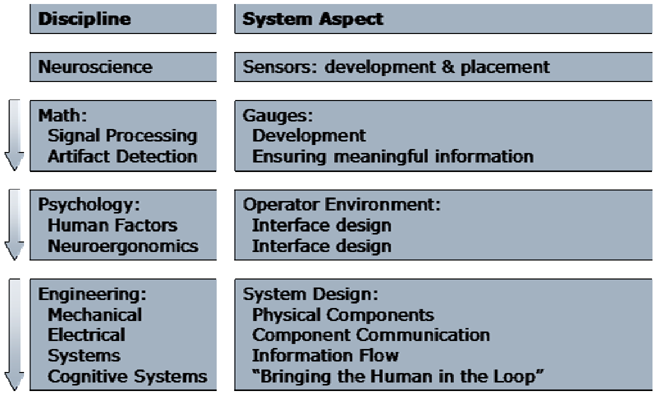
This paper will discuss the migration from modular design to platform-based design in augmented cognition systems, will show the methods of validation and verification of these systems using platform-based design, and will make an argument for the use of platform-based design in all such systems.

Augmented Cognition

Augmented Cognition is an emerging field of research that seeks to enhance users' abilities by developing technologies that designed to address bottlenecks, limitations, and biases in cognition. It strives to develop computational methods that address information processing bottlenecks intrinsic to human-computer interaction. The field is developing new technologies to noninvasively measure the cognitive state of humans and to use the knowledge of that state to adapt closed-loop computational systems to humans' needs.

This technology was initially investigated because of the continuing excess of information that people face every day. Not only is there too much valuable and useful information than can reasonably be digested, there is also an ever-growing expanse of information that is useless either because of its nature or because of who gets the information and when they get it. The AugCog program will dramatically increase the ratio of good information to "bad" information and will enable end users to do better and more complete work, faster.

The design of Augmented Cognition systems requires knowledge of many different disciplines, including neuroscience, human factors, electrical engineering, mechanical engineering, and systems engineering. Obviously, there are very few people in the world who can claim to be experts in all of these fields, so these systems are typically designed by large groups of people – making some aspects of platform-based design even more important. For example, though systems and cognitive systems engineers may design the overall architecture for an Augmented Cognition system, it is likely the neuroscientists, mathematicians, and electrical engineers who will eventually realize the components that will make up the final system. If there is not a formal mechanism for communicating the high-level system constraints and requirements to the end designers, it is likely that things that seem to be common sense at a high level (i.e. avoiding the use of heads-up displays in already cognitively taxing visual situations) will be ignored at the design level. Therefore, the use of identified components, general and specific architectures, and constraint-based requirements are all essential in the design of these systems.



Discipline	System Aspect
Neuroscience	Sensors: development & placement
Math: Signal Processing Artifact Detection	Gauges: Development Ensuring meaningful information
Psychology: Human Factors Neuroergonomics	Operator Environment: Interface design Interface design
Engineering: Mechanical Electrical Systems Cognitive Systems	System Design: Physical Components Component Communication Information Flow "Bringing the Human in the Loop"

Figure 1: Augmented Cognition Enabling Technology Areas

Augmented Cognition systems are extremely complex and exhibit attributes of both embedded and reactive systems, and are driven by both temporal and spatial logic. These systems also comprise the most advanced technologies available in cognitive state sensing and information processing bottleneck mitigation, meaning that the systems are often working with unproven and possibly unstable technologies. All of these reasons increase the need for the inclusion of platform-based principles in the design process.

Platform-Based Design

Both component- and platform-based design capitalize on the principal of modularity. Modularity involves grouping together specific functions into small building blocks of a product to make a complex system more manageable. It helps shorten development, deliver customer-tailored products, and improve profit.

Platform-based design moves one step beyond that, utilizing both top down and bottom up design approaches. In the top down approach, special attention is paid to high level views and ideas of implementation but the entire system is constrained within the framework of these implementations. In the bottom up approach, special consideration is paid to the components and the rest of the system is constrained based upon the component specifications. However, in platform-based design, both views important. A real world example is the Dell Company. Dell uses platform-based design to create customized computers. The implementation of this product is a computing system that fits consumer's needs. The specifications of the system are based upon consumer needs including specific operating system, RAM, storage space, etc. From the Dell example, one can see how platform-based design can be utilized to create a family of products that are tailored to a consumer's specific needs.

The advantages of platform-based design are reuse of design components, reduced design time, and early system verification and validation. In platform-based design, the similarities of the family of products are emphasized in an attempt to reuse key components. For instance with Dell, though the specifications of each system are different, it is known that a hard drive should be a certain size, a monitor is needed, etc… Therefore, when designing a family of products at a general level, engineers can design certain features once and reuse those requirements for all systems, thus eliminating redundant designing and reducing the design time. Additionally, with platform-based design, engineers are able to perform system verification and validation in earlier stages of product design. Using higher levels of abstraction, engineers can ask basic questions such as:

- Will this do what I originally wanted it to do?
- Are all requirements accounted for in the design process?
- Are the system communications in sync with one other?

Many of these questions can be answered by looking at requirements, system architecture, communication flow, and system logic functions before component is actually built.

Platform-Based Design for Augmented Cognition Systems

Most of the platforms explored thus far for augmented cognition systems are in the military domain. This is primarily because of DARPA's substantial investment in the *Improving Warfighter Information Intake Under Stress* program, beginning in 2000. Currently, that program is looking at the platforms of: stationary Command and Control (C2); cockpits (both driving and aviation); and mobile individuals (the dismounted soldier). An additional platform that looks promising and will be explored in the near future is training, particularly in virtual environments. It is assumed that as the enabling technologies develop and advance, that Augmented Cognition technologies will also be applied to nonmilitary domains, such as office work or traditional learning environments.

In order to highlight the differences and similarities between seemingly disparate platform instantiations, two platforms will be discussed here: driving and learning (in a traditional environment). Driving is a platform to which the technologies have already been applied, and is highly technology driven already. Learning environments have not yet been explored by Augmented Cognition technologists, and it is a highly "analog" environment, which demands that designers consider creative ways of enabling a closed-loop computational system that includes the students.

In examining these two platforms, it is helpful with a "catalog" of potential system components, including sensors, models, and interfaces. This catalog of components can be used when designing any augmented cognition system and will definitely grow over time. A non-exhaustive list includes:

- Sensors
  - Cognitive
    - Direct Brain Measures: EEG, fNIR
    - Psychophysiological Measures
      - HR, EKG
      - Pulse Ox
      - Posture
      - GSR
- Interfaces
  - Visual
    - Heads up display
    - Traditional display
    - Alert
    - Warning
    - Picture
    - Text

-
    -
        - Temperature
        - EOG
        - Pupilometry
        - Gaze Tracking
- Environmental
    - Platform Measures
        - Location
        - Internal Conditions
        - Fuel
        - Weapons
    - External Measures
        - Weather
        - Presence of Chemical or Biological Agents
        - Situational Awareness
        - Hostility
        - Obstacles
- Task
    - Status
- Auditory
    - Voice
    - Warning
    - Spatially locatable
- Tactile
    - Warning
    - Directional cue

Starting with this catalog of components enables the bottom up portion of the design process. Designers starting with a complete version of this list can feel confident that they have exhausted the component research portion of the design process.

Once the design elements have been identified, we must begin implementing constraint-based requirements – this begins the top down portion of the design process. For example, in a driving environment where it expected that a driver would maintain physical contact with the wheel (using her hands), the seat (using her body) and the pedals (using her feet). Therefore, it is reasonable to expect that her hands, body, and feet are the only opportunities for haptic interfaces. Also, knowing that using the pedals requires dexterous action on the part of the feet, one could surmise that haptic interaction might inhibit the manipulation of the pedals and determine that the hands and body are actually the only opportunities for haptic interaction. This constraint would have to be documented and specified in the requirements by the systems engineers to ensure that the human factors engineers would not design in "buzz" alert to be placed on the yet untouched upper arm, for example.

The final step in ensuring top down and bottom up design is the specification of communications between the now constrained components: this allows for the reuse of components between system designs, and the "swapping" in and out of different versions of the components throughout the design cycle. An example of such a specification for a driving platform would be:

Module	**Platform Instance**	**Inputs**	**Outputs**
Cognitive Sensor	EEG	Brain electrical activity	Electrical activation in μV; activation localization on common map
Cognitive Sensor	fNIR	Brain blood oxygenation	Ratio of oxygenated to non-oxygenated hemoglobin
Cognitive Sensor	Posture	Pressure	Pressure on seat in N; pressure localization on common map
Cognitive Sensor	GSR	Skin moisture content	Electrical conductance of skin in mV
Cognitive Sensor	EOG	Muscular electrical activity	Electrical state in mV

Cognitive Sensor	Pupilometry	Pupil size	Pupil diameter in mm
Cognitive Sensor	Gaze tracking	Gaze location	Gaze focus localization on common map
Environmental Sensor	IC - Themometer	Temperature	°C
Environmental Sensor	IC - Lighting	Light level	Luminance
Environmental Sensor	IC - Humidity	Air moisture content	% moisture content of air
Environmental Sensor	IC - Clock	Time of day	Hour & minute of day
Task Sensor	Status	Task completion status	Aspects of task completed; % of task completed

The best non-Augmented Cognition example of this phenomenon would be the design of seats in a car. As long as systems engineers are aware of the size, space, heft, and interface requirements of each version of the seats, they do not care what kind of seat the mechanical engineers design. The seat is not required for simulation and testing of the car, and the introduction of new seats that are within the specified requirements will not impact the rest of the car.

This is analogous to the integration of updated sensing technologies into an augmented cognition system. While the introduction of a totally new technology would require some redesign of any system, inserting merely updated sensor systems (i.e. a new EEG system) should be as easy as updating the seats in a car. As long as the required inputs (electrical activity of the brain), outputs (electrical activation in µV; localization of activation), and constraints (size, weight, location, etc…) are known, any system that fits or can adapt to the environment can be integrated with minimal impact on the rest of the system.

System Architecture and Design

As mentioned in the previous section on Augmented Cognition, these systems are highly temporally dependent, as illustrated in the following diagram.

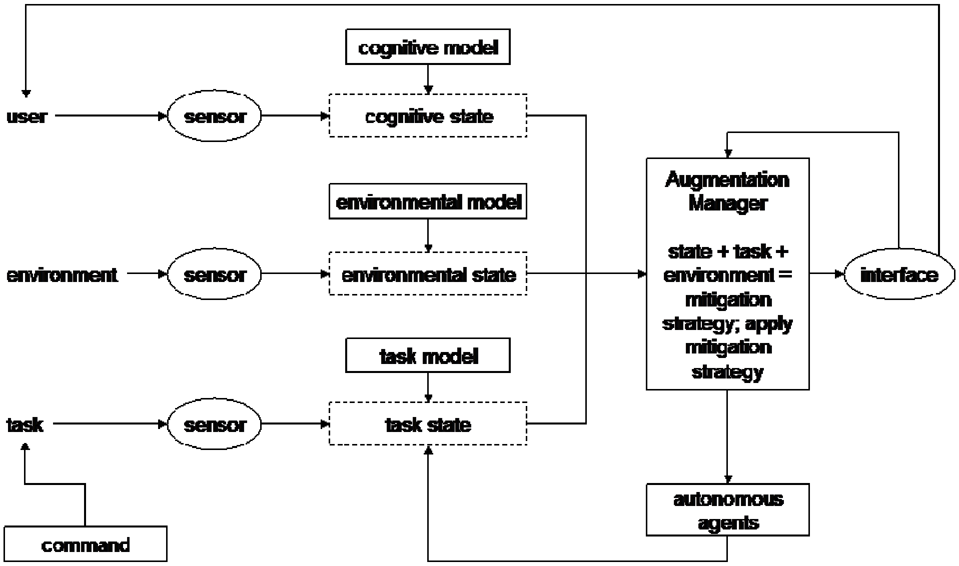


Figure 2: System Level Augmented Cognition Architecture

The arrows in Figure 2 indicate flows of information. The general flow is as follows: commands influence a task; sensors detect activity in the user, environment, and task; sensed and modeled information is combined to create real-time models of the user, environment, and task; state information and interface information is used by the Augmentation Manager to determine an appropriate information-bottleneck mitigation strategy; the Augmentation Manager impacts the interface, which communicates with the user, and autonomous agents if necessary, which complete tasks that the user is to overloaded to complete.

Several things about this architecture illustrate the complexity of Augmented Cognition systems, but the most obvious is the element of time. Everything in the system is responding in real time, all sensing, modeling, and mitigation in the system is continuous – but a certain flow of information is required in order for the Augmentation Manager to have appropriate information on which to take action. The basic solution for this enigma is that cognitive, environmental, and task state are all continuously updated in real time, but are independent of each other. This way, the Augmentation Manager can use only the current states to evaluate what mitigation strategies are needed to improve information flow. A simplified version of the architecture might look like:

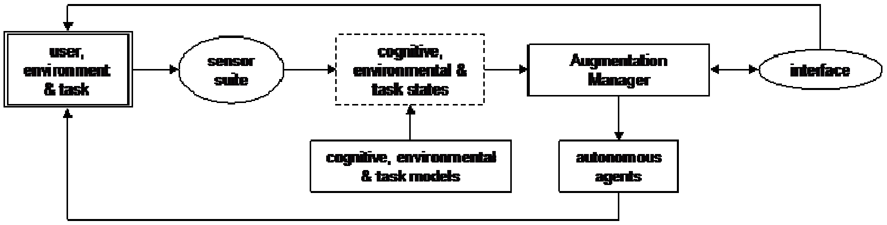


Figure 3: Simplified System Level Augmented Cognition Architecture

The most important aspect of Augmented Cognition systems is their closed-loop nature – this is how humans are able to be part of the system. As shown in the general system architecture, the only aspects of the system that bring the human in the loop are the real-time sensing and the interface. In this way, the human completes the loop.

By comparing the system-level architecture with instantiations of platform-specific architectures, it is easy to see the importance of including both top-down and bottom-up design methodologies. In Figure 4, an architecture for a driving system is specified. It has the same general structure as the system-level architecture, but the sensors and interfaces are specified, potential mitigation strategies are identified, and the links to command or autonomous agents are not available – a representation of an actual instantiation of an Augmented Cognition system.

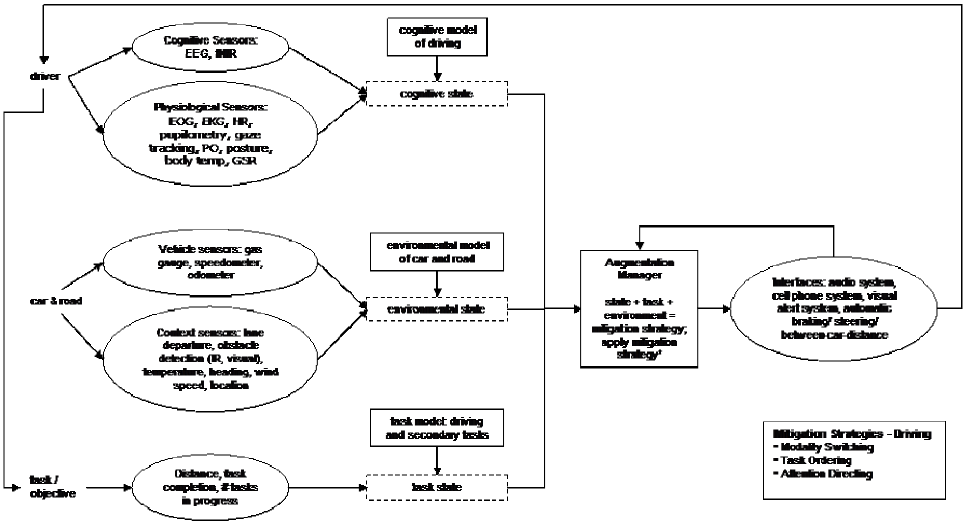

driver
Cognitive Sensors: EEG, fNIR
Physiological Sensors: EOG, EKG, HR, pupilometry, gaze tracking, PO, posture, body temp, GSR
cognitive model of driving
cognitive state
car & road
Vehicle sensors: gas gauge, speedometer, odometer
Context sensors: lane departure, obstacle detection (IR, visual), temperature, heading, wind speed, location
environmental model of car and road
environmental state
task / objective
Distance, task completion, # tasks in progress
task model: driving and secondary tasks
task state
Augmentation Manager
state + task + environment = mitigation strategy; apply mitigation strategy
Interfaces: audio system, cell phone system, visual alert system, automatic braking/ steering/ between-car-distance
Mitigation Strategies – Driving
- Modality Switching
- Task Ordering
- Attention Directing

Figure 4: Augmented Cognition Architecture for a Driving System

In Figure 5, an architecture for a learning system is specified. It also has the same general structure as the system-level architecture, but the same specifications are made as in Figure 4's driving system.

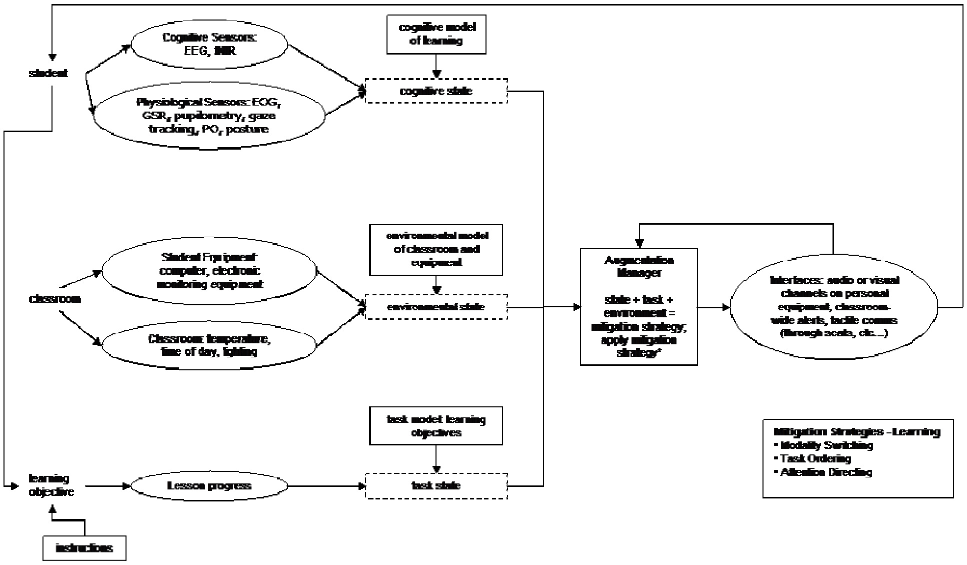

student
Cognitive Sensors: EEG, fNIR
Physiological Sensors: EOG, GSR, pupilometry, gaze tracking, PO, posture
cognitive model of learning
cognitive state
classroom
Student Equipment: computer, electronic monitoring equipment
Classroom: temperature, time of day, lighting
environmental model of classroom and equipment
environmental state
learning objective
instructions
Lesson progress
task model: learning objectives
task state
Augmentation Manager
state + task + environment = mitigation strategy; apply mitigation strategy*
Interfaces: audio or visual channels on personal equipment, classroom-wide alerts, tactile comms (through seats, etc…)
Mitigation Strategies - Learning
• Modality Switching
• Task Ordering
• Attention Directing

Figure 5: Augmented Cognition Architecture for a Learning System

Both system architectures are illustrative of the platform-based design principles discussed thus far in the paper. The realization of the system architectures is the aspect of the design process where using these principles is most evident – it allows for faster development and design of the required systems. However, platform-based design becomes truly invaluable in requirements generation and the validation and verification processes, as it allows these steps to be accomplished early in the design, rather than at the end, when changes and corrections are extremely expensive to implement.

Requirements

Requirements generation is an essential aspect for any system effort, including a process using platform-based design to generate a family of products. In this section, we will examine general high-level design and constraint based requirements for a proposed augmented cognition system.

General High Level Requirements

General high-level requirements are used to communicate the basic goals and objectives of the system to managers, developers, users, and/or other contributors. An example of high level requirements for Aug Cog are listed below.

General High Level Requirements
Level 1
The system must be able to assist the user in completing mission in real time, taking into account mission critical data.
Level 2
The system must be able to sense user cognitive state.
The system must be able to sense environmental state
The system must be able to sense task state.

The system shall be able to communicate with the user.
The system shall be able to communicate with platform interfaces.
The system shall be able to alter interfaces.
Level 3
The system shall contain devices that measure the user's cognitive state.
The system shall be able to analyze the data from sensing devices to determine user's cognitive state.
The system shall contain devices that measure the environmental state.
The system shall be able to analyze the data from sensing devices to determine environmental state.
The system shall contain devices that measure the task state.
The system shall be able to analyze the data from sensing devices to determine task state.
Level 4
Cognitive State
The system shall be able interoperate with multiple instances of the same form of cognitive measuring devices.
Cognitive measuring devices should have an accuracy of 80% or greater.
The system shall be able to receive data from cognitive measuring devices.
The system shall be able to receive data from cognitive models.
The system must be able to determine cognitive bottlenecks.
Environmental State
The system shall be interoperable with multiple instances of environmental measuring devices.
Environmental measuring devices shall have an accuracy of 80% or greater.
The system shall be able to receive data from cognitive environmental devices.
The system shall be able to receive data from environmental models.
Task State
The system shall be interoperable with multiple instances of task state measuring devices.
Task measuring devices shall have an accuracy of 80% or greater.
The system shall be able to receive data from cognitive task devices.
The system shall be able to receive data from task models.

Upon examination, one can determine that these high level requirements do not detail specifications or include detailed information about the system. Since augmented cognition systems are extremely complex, high level requirement generation is especially important to ensure that all designers and engineers have the same concepts and goals to move forward. However, high-level requirement generation is not extremely useful in determining how the system can achieve these goals.

Design Requirements

Design requirements are needed to help creators of the different system components communicate how system components are to work together and how objectives from high level requirements can be met. Design objective levels of abstraction can vary from nearly-high-level requirements all the way to extreme levels of detail (i.e. bolt size). For the purposes of this paper, we have created simple design requirements.

Design Requirements
System shall not inhibit platform's original capabilities
System shall be compatible with platform interface
System shall be able to alter platform interfaces - Driving: seat, steering wheel, dashboard, exterior of car - Learning: Desk, Chair, computer

The platform shall contain equipment to sense environmental state - Driving: Location, internal conditions, fuel, weapons, external weather, prescence of chemical &/or biological agents, obstacles, hostile conditions - Learning: internal conditions
The platform shall contain equipment to sense user state - Driving: EEG, fNiR, THz, HR, Posture, Pulse Ox, GSR, Temp, EOG, Pupilometry, Gaze tracking - Learning: EEG, fNiR, Posture, Pulse Ox, GSR, EOG, Pupilometry, Gaze tracking
The platform shall contain equipment to sense task state - Driving: mission status - Learning: mission status
Algorithms shall be created to model platform environment state - Driving: Driving path - Learning: Classroom
Algorithms shall be created to model platform user state - Driving: Driver - Learning: Student
Algorithms shall be created to model platform task state - Driving: Get from point A to point B - Learning: Learn concept X
The environmental state shall be determined through the input of environmental models and data from environmental platform sensors
The user's cognitive state shall be determined through the input of cognitive models and data from user platform sensors
The task state shall be determined through the input of task models and data from task platform sensors
Augmentation Manager shall be able to form a mitigation strategy based upon data from platform interfaces, user cognitive state, environmental state, and task state.

The difference between platforms is clear here, as well as the fact that there are more detailed directions for the inclusion of different equipment to be used with specific platforms. This works well for higher level design requirements but once designers lower levels of abstraction, the platforms need to be discussed individually and at some great deal. This is achieved using constraint-based requirements.

Constraint Based Requirements

Constraint-based requirements are dictated by platform constraints and restraints from design requirements. A few constraint based requirements for each platform is listed below.

Constraint Based Requirements
Driving
System must be compatible with automobile standard functions.
System shall not inhibit driver’s vision of the road and/or surroundings.
No system equipment/procedure shall require driver to migrate attention behind self.
No system equipment/procedure shall require the driver to move beyond driver’s seat.
No system equipment/procedure shall require the driver to have both hands off of the steering wheel.
Learning
System shall be compatible with other instructional instruments in classroom.
System shall be adjustable to fit a variety of young users.
System shall be contained to student desk and chair area.
System shall not inhibit students from hearing and seeing teacher and classmates.
System shall allow easy application and removal from student.

Requirements are an integral part of system engineering for large complex systems especially those using platform-based design. General high level requirements are necessary to communicate big picture ideas while design and constraint based requirements go to lower levels of abstraction. Each is necessary and can assist in early system verification and validation procedures.

System Verification and Validation

System Verification and Validation is typically done at the end of the design phase of product development. However, when using platform-based design, system verification and validation (V & V) can be completed earlier in the design stage. Tools that assist in doing system V & V are requirements tracing, analysis of communication flow, spatial logic, and temporal logic.

Requirements Tracing

Requirements traces are useful in ensuring that all requirements have a purpose and all lower level requirements are based upon and congruent with high level requirements. Its purpose is to ensure that the designer takes into account all the general high level requirements when creating his design requirements. When looking at a requirements trace, one needs to make sure that all general high level requirements map to at least one design requirement and vice versa. If this is not achieved, more requirements need to be created so that this criteria is met. Since platform-based design is a combination of bottom up and top down approaches, this sort of requirements trace is necessary to make sure that the two methods align properly with one another.

Another requirements trace that may be useful is tracing requirements to system components. This form of trace ensures that requirements are written to govern the development and interaction with other system components. Again, it is important that every requirement maps to at least one component and vice versa.

Analysis of Communication Flow

Another useful system V & V tool is analysis of communication flow. This method requires and examination of the system architecture and a determination of what information can flow to and from components. Analysis of communication flow can be done at high and low levels of abstraction. This technique identifies that the information leaving one unit (A) and going into another (B) is compatible with the unit it is entering (B). For example, if unit B is looking for a numeral, and unit A's output is an alpha value, then one can determine that there is already going to be a problem before anything is constructed. For this project, one must look at the communication flow table to ensure that the expected outputs and inputs are appropriate according to the system architecture.

Spatial Logic

Spatial logic looks at the design area in three dimensions and determines any limitations of the system due to its platform environment. It can also determine if the designer created requirements that take into account the unique features of the platform that may be overlooked when discussing multiple platforms. Constraint-based requirements play a major role in this section. The advantage of using spatial logic is that it answers key questions such as:

- Are platform and interface requirements suitable for each platform?
- Are system components creating an uncomfortable user environment?
- Are there any systems components placed in an unsafe position that inhibit the functions of the platform?
- Did designers adhere to constraint based and design requirements?

Human factor engineers will most likely be involved in this aspect of design and analysis for any augmented cognition system.

Temporal Logic

Finally, temporal logic takes into account time sequences of the system. Questions that can be answered through the utilization of temporal logic are:

- Is this receiving the data at the correct time?
- What is the sequence of data that needs to happen before the next event can take place?
- What is the critical path of data?
- What are timing constraints for this component to perform its specific task?
- What is the projected reaction time of the system?

Augmented Cognition systems face some unique challenges because they:

- Are primarily controlled be feedback loops (i.e. user-user sensors-user state-Aug Manager-interfaces-user)
- Feature three sub loops running in parallel (i.e. user, environment, and task)
- Feature continuously-streaming data (i.e. user, environment, task, and interfaces are always emitting data)
- Must account for sensing and modeling delays, user reactivity time, and the timing of outside factors

Temporal logic identifies key time issues in complex systems that if otherwise not considered could inhibit system functions.

System V & V for products created by platform-based design can be done earlier than conventional systems to potential save a great deal of money, time, and effort. Key tools for system V &V on complex systems such as augmented cognition platforms would be requirements traces, analysis of communication flow, spatial logic and temporal logic.

Conclusion

Augmented Cognition systems will improve human information processing capabilities by providing easily-assimilated information by knowing the human cognitive state and the current tasks being executed. They will enable a huge advance in operator-computer effectiveness by creating cognitive closed-feedback loops between operators and adaptive computer based systems. Additionally, they will have the capability to change information modalities (providing information through a different medium such as aural, spatial, or verbal) or offload tasks – whatever is required to enable and optimally functioning user.

Current Augmented Cognition systems are in the prototype phase at their most advanced, and are totally platform- and task-specific. The components that have been created are not extensible from environment to environment, and the next generation of systems will be largely 'built from scratch.' In envisioning the design cycle for future systems, it was first assumed that a merely modular design would enable the reuse of these sophisticated components. However, it was later realized that aspects of the specific platforms would have to be part of the design process in order for systems to actually be realized: modular design made implementation of augmented cognition designs possible, but platform-based design makes it feasible.

Using the principles of platform-based design will allow Augmented Cognition technologies to be extended from primarily military-based platforms to educational, office-work, and even personal-use environments. This technology has the potential to revolutionize the way and amount of information that humans are able to process, and will become only more important as the amount of and scope of information available continues to increase.

References & Further Reading

Austin, M. (2004, December). Toward Platform Architectures for Modular Cognitive Cockpits. Abstract submitted to Augmented Cognition International, Las Vegas, NV. In review.

Austin, M (2004, September). Class Notes for ENSE 621-623. Not published.

Browning, T (2001, August). Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Directions. IEEE Transactions on Engineering Management, Vol. 48, No. 3. pp 292-306.

Ericsson, A. and G. Erixon (1999). Controlling Design Variables: Modular Product Platforms. ASME Press, New York, NY.

Holmqvist, T. and M. Persson (2003). Analysis and Improvement of Product Modularization Methods: Their Ability to Deal with Complex Products. Systems Engineering, Vol. 6, No. 3. pp 195-209.)

Horowitz, B., J. Liebman, C. Ma, T. Koo, A. Sangiovanni-Vencenteli, S. Sastry (2003, January). Platform-Based Embedded Software Design and System Integration for Autonomous Vehicles, Proceedings of the IEEE, Vol. 91, No. 1. pp. 198-211.

Huang, C. and A. Kusiak (1998). Modularity in Design of Products and Systems. IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans, Vol. 28, No. 1. pp. 66-77.

Kusiak, A. and C. Huang (1997). Design of Modular Electrical Circuits for Testability. IEEE Transactions on Components, Packaging, and Manufacturing Technology – Part C, Vol. 20, No. 1. pp. 48-57.

Raley, C., Stripling, R., Schmorrow, D., Patrey, J., & Kruse, A. (2004, September). Augmented Cognition Overview: Improving Information Intake Under Stress. Proceedings of the Human Factors and Ergonomics Society 48th Annual Meeting, New Orleans, LA. In press.

Rasmussen, J., A. Pejtersen, and L. Goodstein (1994). Cognitive Systems Engineering. John Wiley & Sons, New York, NY. 1994.

Sangiovanni-Vincentelli, A (2002, February). Defining Platform-based Design. EEDesign of EETimes.

Sangiovanni-Vincentelli, A. (2003, May-June). Electronic-System Design in the Automobile Industry. IEEE Micro. pp. 8-18.

Schmorrow, D., Raley, C., Marshall, L. (2004, March). Toward a Cognitive Cockpit. Poster presented at Second Annual Human Performance, Situation Awareness and Automation Technology Conference, Daytona Beach, FL.

Schmorrow, D., Raley, C., et. al.. (2004, March). Toward improved situational awareness through knowledge engineering. Poster presented at Second Annual Human Performance, Situation Awareness and Automation Technology Conference, Daytona Beach, FL.

Schneider, L. (2004). Verification of Reactive Systems: Formal Methods and Algorithms. Springer-Verlag, Berlin, Germany. pp. 2-3.

Ulrich, K. and S. Eppinger (2000). Product design and development. McGraw-Hill, New York.

Zakarian, A. and G. Rushton (2001). Development of Modular Electrical Systems. IEEE/ASME Transactions on Mechatronics, Vol. 6, No. 4. pp. 507-520.