Driving Design:

Creating Simple Interfaces for Complex Cars

BY CARL F. SMITH, L. RICARDO PRADA, AND MOHAMMAD T. RAHMAN

Technology designed for today's automobile interiors now includes sophisticated digital interfaces for virtually any task — including many never intended for automotive use. This wealth of information taxes the limited real estate of the average dashboard and risks losing drivers in a sea of information.

The BMW iDrive dashboard with the interface at the right and the selection knob on the center console.

To address this problem, car manufacturers have developed interfaces that combine multiple controls into one device. Combining displays, however, may have some unintended consequences. As more tasks are added to the device, drivers may find that identifying and selecting the correct operation becomes increasingly difficult. For example, task analysis of the BMW iDrive 1.0 system showed that tuning the radio to a station that didn't have a preset button required as many as six user interactions.

The dangers of complex interfaces in automobiles are clear: accidents increase whenever looking away from the road increases. Public reaction to early multi-functional devices reflected this relationship — reviewers criticized the interfaces as "making you take your eyes off the road just long enough to plow into a solid object."

One solution might be to minimize operator demand through interface redesign. However, this approach begs the question, "How do you redesign such a feature-rich application?" This article describes how we developed a methodology for redesigning a complex multi-functional interface — in this case, a digital interface embedded in the car dashboard.

Picking a Design Methodology

We started by asking which design approach would be the most useful. The number of available functions made a detailed task analysis of each one unfeasible for a project with a limited timeline. We might also miss interconnected functions if each function were analyzed individually. This approach could easily create situations in which optimizing design for one task might make another task more difficult. Take our radio station example: combining multiple functions into one display (optimizing for space) changed a simple knob turn into multiple user actions across a changing display (making the task more difficult).

Another way to address these issues would be to analyze the human and automobile as one system. By analyzing overall system constraints, we could then see the limitations in the information required by both the driver and the interface. We could then use this understanding to design methods of interface layout and operation that accounted for the entire system — human and machine.

We decided to use parts of cognitive work analysis, specifically, work domain analysis and abstraction hierarchy, to understand how high-level system goals and functions could be restructured for quick access and intuitive presentation. (Cognitive work analysis identifies system constraints that account for the interaction between the mechanical part of the system and users' behavior.)

Starting the Analysis

Some commercial systems have up to 700 separate functions. With that many functions to consider, we needed a way to categorize them into groups that could be easily understood by drivers. With the help of a sample group of experienced automobile drivers, we used a thematic card sort to distill the desired functions into five groups: safety, comfort, communications, entertainment, and performance.

These five groups became the basis for our work domain analysis (Table 1). This analysis technique describes each system group, or subsystem, at various levels, ranging from a functional level (the overall purpose of the system) to a physical level (the actual objects used). Unlike standard task analysis, work domain analysis allowed us to condense a variety of seemingly unrelated functions into a table that showed how one physical form could control multiple functions.

For example, the functions of the performance subsystem in Table 1 do not require driver input during vehicle operation. This tells the designer that performance functions do not need to be immediately accessible.

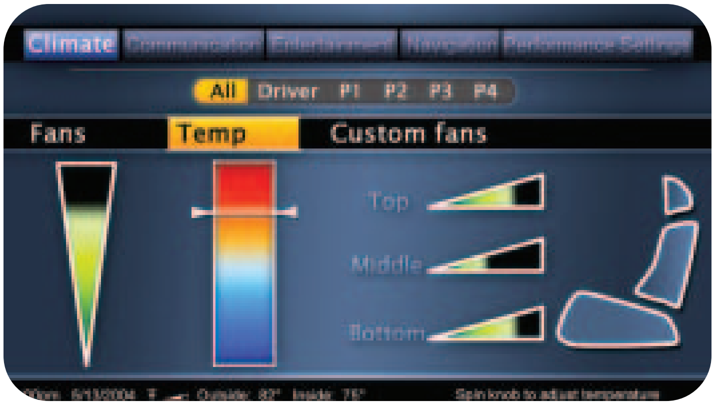
Decomposing the system into levels doesn't explicitly show the relationships between different functions and objects, however. To represent these relationships, we used an abstraction hierarchy. An abstraction hierarchy describes the whole system, but connects different levels of abstraction through their "means-end" or "why-how" relationships to each other. Items in the levels above an item show *why* the item is there, while items below an item show *how* it is used. If items on the same level are connected, the connections identify possible conflicts or shared functions among subsystems.

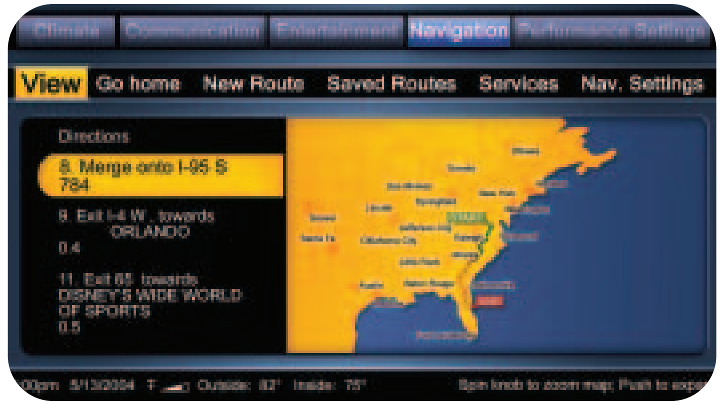
By showing the relations between system levels, we could make better decisions about

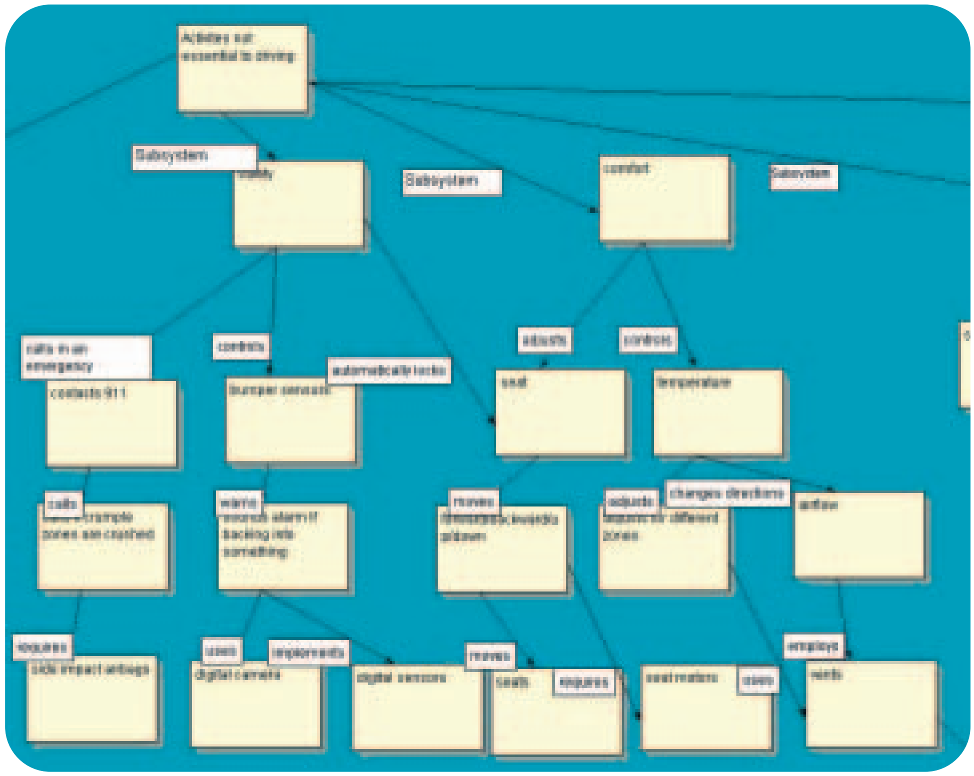
The dangers of complex interfaces in automobiles are clear: accidents increase whenever looking away from the road increases.

TABLE 1: WORK DOMAIN ANALYSIS FOR THE CAR DASHBOARD INTERFACE PROJECT

Functional Purpose	**Control of activities not involved in driving Provide non-essential driving func-tions**					
System Level	**Overall System**	**Safety Subsystem**	**Comfort Subsystem**	**Entertainment Subsystem**	**Performance Subsystem**	**Communication Subsystem**
Abstract Function	•Minimize visual identification time •Minimize driving task interference •Limit motor use	•Protect drivers from simple threats to vehicle and person	•Provide physical comfort to driver	•Entertain passengers without distracting driver	•Allow drivers to change performance characteristics	•Receive incoming information •Send outgoing information
General Function	• Rapid shift from one subsystem to another •Early identification of subsystem being controlled •Access to subsystem states	•Communicates to airbag •Communicates to seat •Communicates with bumper sensors •Contacts emergency personnel	•Adjusts seat •Warms seat •Controls interior temperature •Controls different temperature zones of car	•Plays multiple media forms •Controls all media from single interface	•Adjusts suspension •Adjusts setting for road conditions •Provides performance statistics	•Receives and sends satellite information •Receives and sends cellular signal
Physical Function	•Allows driver to cycle through different subsystems •Control any non-essential driving tasks/situations through interface	•Sounds alarm if approaching object •Activates airbags	•Blows cooled/ heated air •Adjusts air stream for different areas of car •Adjusts Seat	•Receives radio signals •Receives satellite signals •Plays media •Allows for wireless and USB	•Lowers/raises suspension •Provides alerts for dangerous road conditions	•Information about weather conditions •Concierge service •GPS navigation system
Physical Form	•Digital interface	•Blinking lights •Sound generator •Digital interface •Digital camera •Side impact airbags	•Vents •Seat motors •Digital interface	•Bluetooth •Radio •DVD/CD player •MP3 player •Digital interface	•Digital sensors •Suspension •ABS •Adaptive cruise control	•Own satellite-based phone with phone book •Phone •Digital interface •Satellite communication







Right: An abstraction hierarchy showing the Safety and Comfort subsystems.
Above: The Climate view, with temperature selected.
Above Right: The Navigation view, showing directions and a map.

the placement of items on the interface. For instance, items that required both the digital interface and user input while driving (for example, the entertainment subsystem) would need to be easily accessible, while items that didn't (for example, the performance subsystem) would not need to be.

Links in the abstraction hierarchy showed how subsystems should relate to each other. For example, the communication subsystem shares functions and physical elements with the entertainment subsystem, so we knew we should keep the communication system within a click or two of the entertainment subsystem, but could have them both a few clicks away from the performance subsystem.

While work domain analysis and the abstraction hierarchy indicated how to do the interface layout and system logic, they didn't specify how drivers would move through the interface. Problems with earlier designs were related to the number of inputs and the time required to complete actions, as well as the possibility that similar actions across the subsystems would produce contradictory results. For example, if the driver had to press button A in entertainment mode to turn on the radio, she should be able to press the same button in communication mode to turn on the phone.

To ensure that we used the fewest number of actions (parsimony) and that actions across subsystems were consistent, we designed our tasks' input methods using the goals, operators, methods, and selection rules (GOMS) methodology. GOMS modeling showed us how to define procedural consistency across modes and parsimony in the actions required for a task.

Our hybrid approach produced an interface requiring only four modes as compared to car-industry devices with six to eight modes. Screenshots of the finished designs appear above and to the right.

Evaluating the Design

We evaluated the design using cognitive walk-throughs and user testing. The cognitive-walkthrough experts examined the interface using multiple tasks at varying levels of task dif-

Lessons Learned: Heuristics for Multifunctional Design

1 When designing a multifunctional system, analyze both the environment's constraints and the tasks before specifying an interface design.

2 A complex system's interface should reflect the underlying relationships between its functions.

3 Analysis alone cannot create an interface. You need to accompany good analysis techniques with traditional visual design principles.

4 Although traditionally used to evaluate existing systems, the goals, operators, methods, and selection rules (GOMS) methodology can be useful during design phases — it provides the design team with a common interaction language.

5 Cognitive walkthroughs, plus user testing, proved to be effective for discovering system flaws, especially when you use iterative design to fix flaws between evaluations.

ficulty. Our walkthroughs showed little or no misinterpretation when they moved between different modes or actions. Based on these results, we believed that drivers would be able to formulate consistent, correct goals and select the correct actions for tasks.

In the user testing, we asked novices to identify the functions of different icons prior to use. The users' insights proved invaluable in matching our ideas about the interface with what they expected from the interface. Users were often able to intuit the correct set of actions with the redesigned display and showed clear understanding of the system functions. When we compared our designs to the prior designs, our design seemed to alleviate confusion and facilitate comprehension.

The Experts Speak!

We had two goals: to explore a methodology for redesigning interfaces heavy in functionality; and to demonstrate its potential for producing an intuitive and usable interface. Preliminary results are promising. However, pragmatic issues remain: would designers use our approach? Is our finished product really any better for the method used?

As it turns out, the answer depends on whom you ask. We spoke to several experts in the automotive industry, including an automotive engineer, a human factors analyst, and an industrial designer.

The automotive engineer understood the

Improvement through Redesign—One Example

We were driven to the car dashboard interface project by BMW's first version of iDrive. With iDrive 1.0, users required six interactions before they could manually change a radio station. The iDrive system has a knob that lets drivers click through the various options. The steps were:

1. Press the knob to access the entertainment menu.
2. Rotate the knob clockwise two clicks to access FM.
3. Press the control to select FM.
4. Rotate the knob clockwise two more clicks to access manual mode.
5. Press the knob again to select manual mode.
6. Rotate the knob to the desired station.

These six steps use two different forms of operator input and three different displays.

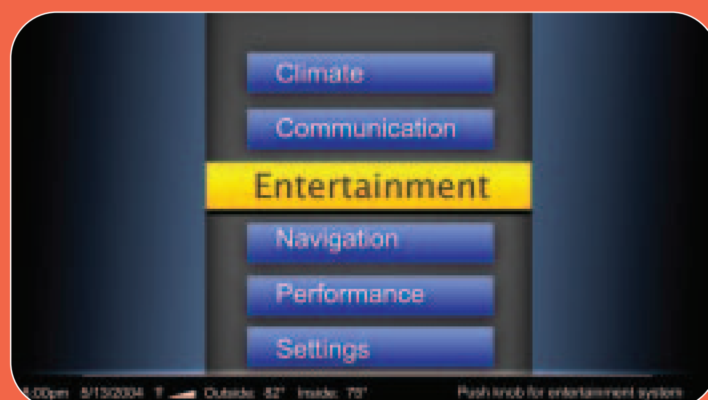
Compare this to our redesign. We also use the controller knob, but the steps are simpler:

1. Press the knob to select the mode
2. Press the knob right to access the radio.
3. Rotate the knob to a specific station.

This process takes three movements, half the movements required by the first system. We were able to define an interface that shows all of the subsystems at once even while the driver was in a given subsystem. Also, GOMS helped us standardize our input actions, so that the effects of specific operator actions are identical, regardless of the system mode.

For example, in our redesign, pressing the controller in one direction or another always moves between different modes and functions, while rotating the knob always lets drivers move within a function.

The result is an interface that reduces the number of movements, is consistent throughout, and changes the display background only once throughout the maneuver.



Above Left: The BMW iDrive entertainment menu, with the controller knob used to change displays and select items.
Top Right: Step1: Press the knob to select the entertainment mode.
Bottom Right: Steps 2 and 3: Press the controller right toward to the radio option and then rotate it to the third station.

And Now for Something Completely Different…

TEXT AND ILLUSTRATIONS BY BERNARD CHAMPOUX

Today's car technologies have extended the activity of driving beyond transportation into managing information (cell phone, Internet) while monitoring driving speed, weather, and the car's immediate environment. What used to be managed from a dash-board of dials and buttons is now being managed via a single computerized unit called a "vehicle user interface" (VUI). VUIs can help drivers manage all of this new information. However, at the University of Saskatchewan, we think it should do more than assist — it should also adapt to the driver.

Interaction with technologies, more than the technologies themselves, are distracting and a potential threat to road safety. From the driver's point of view, a VUI should not be a weak, general information system (the Swiss Army knife approach), but a strong, specific representation of the information needed at the moment.

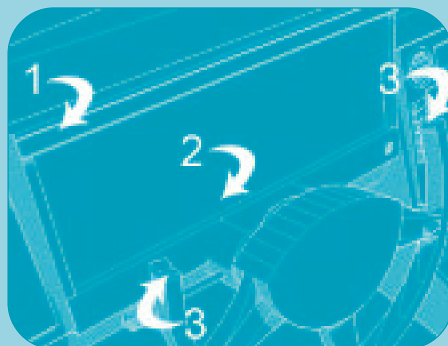
When drivers perform demanding cognitive tasks while driving, they spend more time looking straight ahead and less time looking at the periphery as their cognitive workload increases. For this reason, in our design, we put the VUI in front of the driver.

Integrated computer unit display behind the wheel.

The display can be customized by the driver and saved as settings. The left and the right preset buttons, located on the steering wheel, let the driver switch from one set-ting to another.

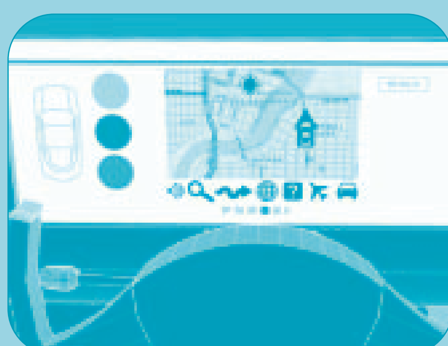
The advantage of having drivers customize their own representations is that they can jump to the right information quickly and safely when a demanding situation occurs.

In our designs, we assumed two basic types of settings, city and highway, and that the driver is holding a demanding conversation on a hands-free phone.



The display screen with the steering wheel. 1. Upper interface. 2. Main screen. 3. The switches on the steering wheel.

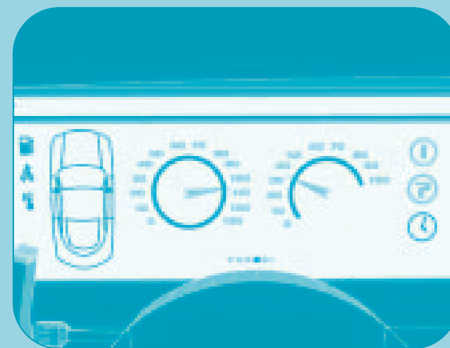
When the driver is on city streets, he or she generally focuses on the traffic around the car and the GPS map rather than on the car's speed.



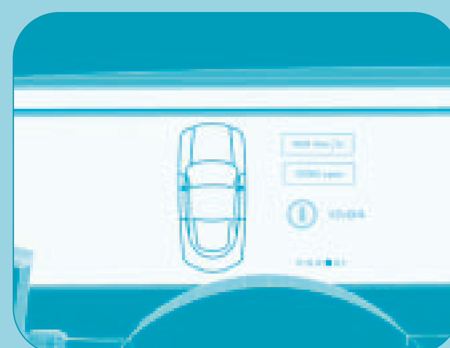
City driving display. Note, from left to right, the collision avoidance system view (the car image), the global positioning map with an arrow showing the car's location, and the speed. The three colored dots—red, yellow, and green—are there to warn the driver about road conditions or traffic. If conditions change, one of them will blink and thereby warn the driver to pay closer attention.

On a highway, driving is less demanding and the representation can be more conventional or even minimalist.

We have not yet done any user testing on the designs, but we hope to get a grant that will fund usability tests.



Highway driving display. From left to right, Gas, seat belt, and temperature icons; collision avoidance system view; speed; and RPM.



Minimalist highway setting: Only the car view; the speed, RPM, and the time shown digitally; and the current gear.

Minimalist highway setting with a video playing at the right (a common feature in Japanese cars but not in North American ones). If another car comes too close, the primary display will warn the driver.

logic in the analysis, and thought it would be useful for guiding design. However, he cautioned that even a well-designed visual device might be too complex for use while driving.

Christopher Monk, a human-factors analyst from SAIC (www.saic.com), said our analytic techniques compared well to the hierarchal task analyses and functional analyses he would normally use to understand the levels and functions available in a system. However, he said it wasn't clear to him how we would turn the analyses into an actual interface design. And although he liked the use of GOMS, he was concerned that, since many designers don't know about GOMS, they might not be able to apply the technique.

Frazier McKimm, an industrial design consultant and founder of Studio McKimm (www.mckimm.com), felt that our analysis could be useful in representing the abstract and/or spatial relationships that often escape strict task analysis. However, he felt that hierarchical functions, task flows, and linear explanations of design logic could be of similar or greater value. One of his suggestions was to show task progression within the interface. He also suggested several interface changes — for example, using static displays across pages to make it easier for drivers to orient themselves.

Final Thoughts

While multiple analysis techniques helped simplify our multifunctional interface, the impact of our redesign remains in question. The automotive engineer noted that drivers still prefer separate controls over "all in one" systems. Also, one expert hinted that the current complexity and negative publicity of multifunctional devices has many automakers focusing on voice activation, haptics (touch), and other methods that minimize visual attention.

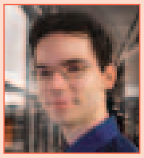
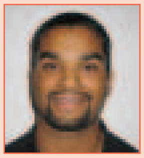
Regardless of the multifunctional automotive interface's future, using analysis techniques to decompose complex systems into parts and constraints can be applied to many devices that are more functional than usable. As systems become more complex, the designer will have to use multiple analysis and evaluation methods to assess the complex interworkings of today's systems, whether they are in the home, the office...or the dashboard. UX

For more information, see http://www.usabilityprofessionals.org

Acknowledgments:

Thanks to Nicholas Prada for editing our visual designs.

ABOUT THE AUTHORS

Carl, Ricardo, and Mohammad are members of George Mason University's Arch Lab, and are obtaining graduate degrees in Human Factors and Applied Cognition. Carl's research interests include alternative aviation displays and visual design principles for complex systems. Ricardo's main interests focus on human-computer interaction and aviation psychology. Mohammad's primary interests lay in automotive design, specifically automotive telematic systems. For more information about the authors and the Arch Lab, see http://archlab.gmu.edu.

ABOUT THE AUTHOR

Bernard Champoux is an associate researcher and graphic designer at the Computer Science Department Interaction Lab at the University of Saskatchewan in Saskatoon, Canada. Mr. Champoux was a researcher at ATR (Advanced Telecommunication Research Institute) in Kyoto, Japan, where he designed a computer-assisted iconic language system (CAILS) and regularly lectured on visual communication at Doshisha University, Kyoto. He also sketched, designed, and developed prototypes for a 3D browsing device at the Eindhoven University of Technologies in Eindhoven, Netherlands, where he taught in-dustrial design. For the past year, Mr. Champoux has been working on mechanisms to design computer-enhanced objects, and is currently developing vehicle user interfaces for intelligent transport systems (ITS). Bernard Champoux can be reached at bjc489@mail.usask.ca.

Reality Check—Industry Observations

Christopher Monk, SAIC (www.saic.com)

When I worked on these types of interfaces, we definitely used task or function hierarchies to understand the levels and layers of functions available with the system. Abstraction hierarchies seem similar and are likely useful.

Cognitive walkthrough is particularly useful. User testing is indispensable. I used, and I hope other designers and evaluators do as well, both of these techniques when working on an interface.

Frazier McKimm, Studio McKimm (www.mckimm.com)

Your methodology seems very close to what we do in our lab. However, we often couple these environments with logic functions, which can be difficult to do without.

I had some concerns about the logic flow: I couldn't discern a hierarchical framework related to the solution of the problem [in the analysis]. It would be constructive to have a logic flow with a route to given objectives. It's important to be able to show the progression and orientation.

When working through steps on the interface, the progression of steps should be visible to the user. For example, you could change background color as users progress through a task to reflect progress towards a goal.

Cognitive walkthrough is essential for rapid redesign, as it allows you to see the effects of rapid progressions. Test and test again!

Anonymous automotive engineer, USA

Your approach will probably result in better designed systems, but the question is, will they still be too complex for people to use while driving? Anything that forces the driver to take his eyes off the road, no matter how elegantly designed, will be a turn-off.

It would be interesting if you applied your methods to optimizing the interface with a voice activation system; perhaps your approach could reduce the menu complexity of these systems.