By Dov Dori

WORDS FROM PICTURES FOR DUAL-CHANNEL PROCESSING

Combining graphics and text representations of complex systems, Object-Process Methodology makes it easier to understand technical ideas, whether or not one is technically oriented by nature or training.

Text and graphics are complementary modalities our brains process interchangeably. Conceptual modeling, recognized as a critical step in architecting and designing systems, is an intellectual activity that would greatly benefit from the concurrent utilization of the verbal and visual channels of human cognition. A conceptual-modeling framework that employs graphics and text would help alleviate cognitive loads. Object-Process Methodology (OPM) is a bimodal graphics/text conceptual-modeling framework catering to these needs. Here, I argue on behalf of the OPM holistic approach in addressing assumptions about the dual channel, as well as limited-channel capacity and active processing. To help make the case, using a running example of a car's emergency braking system, I demonstrate bimodality and complexity management via hierarchical decomposition and animated simulation to address these cognitive needs. Meanwhile, work is under way to employ some of these ideas in a future version of the Systems Modeling Language (SysML) (www.sysml.org).

*When the user expresses a piece of knowledge in one modality—graphics or text—*the complementary one is automatically updated so the two remain coherent at all times.

Humans assimilate data and information, converting it simultaneously into meaningful knowledge and understanding of systems through words and pictures. During eons of human evolution, the human brain has been trained to capture and analyze images, enabling us to escape predators and capture food. In contrast, processing spoken words, let alone text, is a product of a relatively recent stage in that evolution. As our brains are hardwired to process imagery, graphics naturally appeal to the brain more immediately than words. However, words can express ideas and assertions that are way too complex or even impossible to express graphically; as an example, just try graphically representing this sentence to sense the validity of this claim. While a picture may be worth a thousand words, a word or sentence is indeed sometimes worth a thousand pictures. A problem with the richness of natural language is the potential ambiguity that arises from its use. This does not imply that pictures cannot be ambiguous as well, but graphic ambiguity is greatly reduced, even eliminated, by assigning formal semantics to pictorial symbols of things and to the relationships among them.

Diagrams aid cognitive processing due to their specificity [11], a theory proposing that graphical representations limit abstraction and thereby aid "processibility." That is, diagrams, because they usually involve fewer interpretations than free text, are more tractable than unconstrained textual notation. When corresponding words and pictures are presented near each other, learners are better able to hold corresponding words and pictures in working memory at the same time, enabling the integration of visual and verbal models [8]. A contribution of diagrams may be that they reduce the cognitive load of assigning abstract data to appropriate spatial and temporal dimensions; for example, whereas information about temporal ordering is only implicit in text, a flow diagram reduces errors in answering questions about that ordering [6].

A theory called "multimedia learning" proposed in [8, 9] is based on three main research-supported cognitive assumptions:

Dual channel. Humans have separate systems for processing visual and verbal representations [1, 3];

Limited capacity. The amount of processing that can take place within each information-processing channel is limited [1, 2, 10]; and

Active processing. Meaningful learning occurs during active cognitive processing, paying attention to words and pictures and mentally integrating them into coherent representations. The active-processing assumption is a manifestation of the constructivist theory in education, which focuses the construction of knowledge by one's mind as the centerpiece of the educational effort [12]. That is, in order for learning to be meaningful, learners must engage physically, intellectually, and emotionally in constructing their own knowledge.

As the literature suggests, there is great value in designing a modeling approach and supporting tool to meet the challenges posed by these assumptions. While [9] used them to suggest ways to reduce cognitive overload while designing multimedia instruction, they can also be a basis for designing an effective conceptual-modeling framework. Indeed, conceptual modeling is the active cognitive effort of concurrent diagramming and verbalization of one's thoughts. The resulting diagrams and text together constitute the system's conceptual model. A model based on a set of the most primitive and generic elements is general enough to be applicable to a host of domains yet simple enough to express the most complex systems. A sufficiently expressive model can help detect design-level errors, be reasoned about, make predictions, be communicated to other stakeholders, and evolve throughout a system's life cycle.

Such an environment would help us take advantage of the verbal and visual channels and relieve cognitive loads while designing, modeling, and communicating complex systems to stakeholders. These were key motivations some 15 years ago in my design of OPM [4]. The OPM modeling environment implementation by OPCAT[1] [5] embodies the assumptions. Stateful objects (things existing in some state) and processes (things that transform objects by creating or destroying them or by changing their states) are the building blocks of OPM. Structural and procedural links express static and dynamic relations among entities—objects, object states, and processes—in the system, and a number of refinement/abstraction mechanisms are built into OPM for complexity management.

DUAL-CHANNEL PROCESSING

Following the dual-channel assumption, the brain simultaneously engages the visual and verbal channels (likely the two brain hemispheres) for conveying ideas regarding the system's architecture. Indeed, OPM represents knowledge about the system's structure and behavior—pictorially and verbally—in a single unifying model. When the user expresses a piece of knowledge in one modality—graphics or text—the complementary one is automatically updated so the two remain coherent at all times.

To illustrate how to account for the cognitive assumptions, I follow a stepwise example of the modeling of a car's anti-lock brake system (ABS). Figure 1(a) outlines OPCAT's graphical user interface, simultaneously displaying the graphic (top) and text (bottom) modalities needed to exploit human dual-channel processing. The top-right pane presents the model graphically in an Object-Process Diagram (OPD); the one below it in Figure 1(a) lists the same model textually in Object-Process Language (OPL). OPCAT recognizes OPD constructs (symbol patterns) and generates their OPL textual counterparts. OPL is a subset of natural English, and each OPD gives rise to a textual OPL equivalent sentence or phrase.

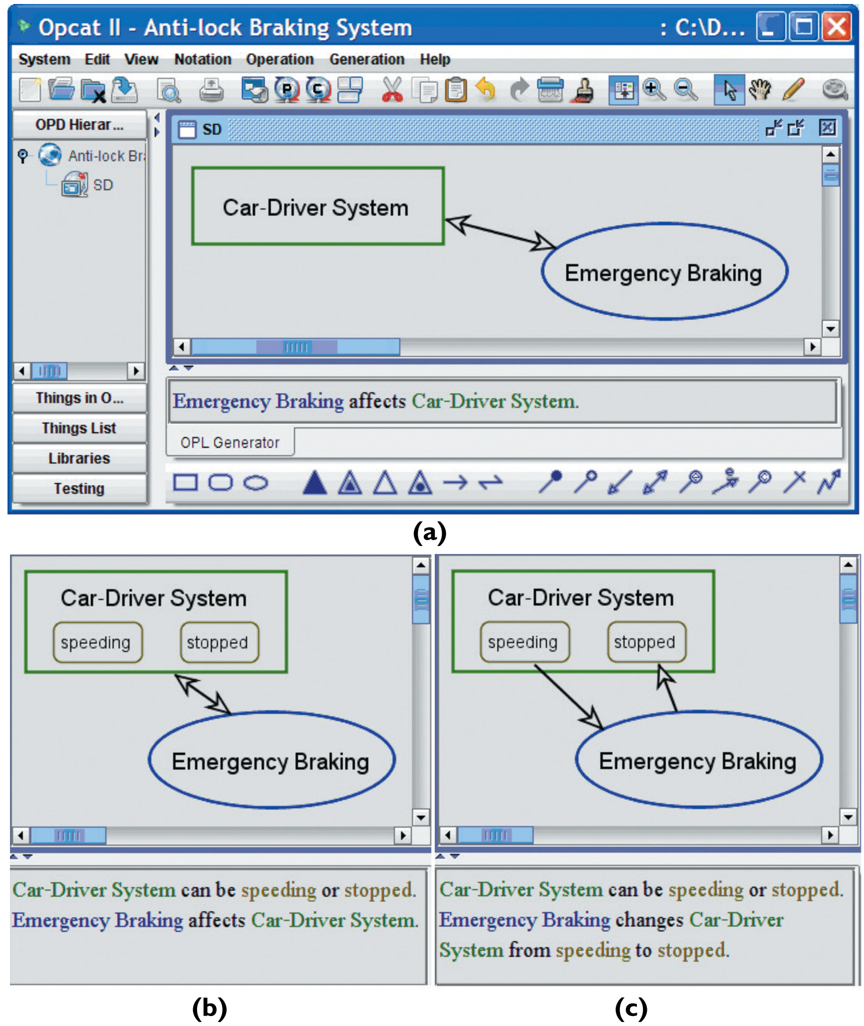

Figure 1. Top-level OPD built in stages. (a) OPCAT user interface, showing the initial system diagram of the anti-lock braking system (top) and its OPL textual specification (bottom) in which the object Car-Driver System is affected by the process Emergency Braking; (b) the states speeding and stopped are added to the Car-Driver System; (c) the input/output link pair is added from the input state to the process and from the process to the output state.

For example, `Emergency Braking`, the central system's process, is the blue ellipse in Figure 1(a), and `Car-Driver System` is an object (green box) affected by and benefiting from `Emergency Braking`. This object-process connection is expressed by linking `Car-Driver System` to `Emergency Braking` via an effect link—a bidirectional arrow indicating that the process affects the object by changing its state from unspecified input state to unspecified output state. As soon as the modeler joins the object with the process through the link, the first OPL sentence, "`Emergency Braking` affects `Car-Driver System`," shows up in the OPL pane of Figure 1(a).

As the example shows, the OPL syntax is designed to generate sentences in plain, natural (albeit restricted) English. Unlike programming languages, OPL names can be phrases (such as `Emergency Braking`). As a subset of English, OPL is accessible to nontechnical stakeholders, and other languages can serve as the target OPL. To enhance the text-graphics

[1] A research version of OPCAT is available for free download at www.opcat.com/downloads/restricted.

link, the text colors of the process and the object names in OPL match their colors in the OPD. Since graphics is more amenable to cognitive processing than text, modelers favor modeling the system graphically in the OPD pane, while the textual interpretation is continuously updated in the OPL pane and can also be continuously referenced to verify that the modeler's intent is captured.

The OPL sentences constructed or modified automatically in response to linking graphical symbols on the screen provide immediate feedback to a modeler, as well as to his/her audience. This real-time human-like response "tells" the modeler what the modeling environment "thinks" he/she meant to express in the most recent graphic-editing operation. When the text does not match the modeler's intention, the modeler can take corrective action. Such feedback is indispensable for spotting and correcting errors at an early stage in any system's life cycle, before they have a chance to propagate and cause costly downstream damage. Any correction of the graphics changes the OPL script; changes can be applied iteratively until a result satisfactory to all stakeholders is obtained. While generating text from graphics is the prevalent working mode, OPCAT also generates graphics from text.

The System Diagram is constructed such that it contains a central process, which in this case carries out the system's main function and delivers its main value to the beneficiary for whom the system is built. In the case of anti-lock braking, Emergency Braking is the process that provides value to the Car-Driver System, the beneficiary.

Having established this basic conceptual design, we can now be more specific. In Figure 1(b), I specify the two states—speeding and stopped—of Car-Driver System. This specification triggers generation of the OPL sentence "Car-Driver System can be speeding and stopped."

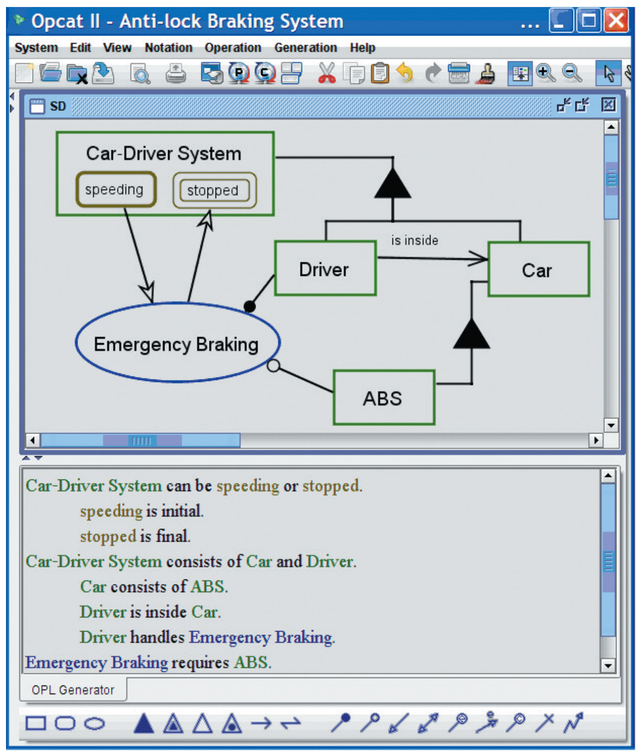

Figure 2. Top-level OPD (system diagram) resulting from adding Car, which consists of an ABS, and characterized by Speed and the effect of Emergency Braking on changing Speed from high to zero.

By replacing the effect link with an input-output link pair consisting of an input link (from speeding to Emergency Braking) and an output link (from Emergency Braking to stopped), Figure 1(c) explicitly shows that "Emergency Braking changes Car-Driver System from speeding to stopped," as specified (equivalently) in the OPL sentence at the bottom of the figure.

The System Diagram is elaborated further in Figure 2. First, the modeler "unfolds," or specifies, the parts of the whole Car-Driver System. The black triangle is the aggregation-participation symbol, specifying that "Car-Driver System consists of Car and Driver." Driver is linked to Emergency Braking via an agent (human enabler) link (the line ending with a black circle), and ABS, a part of Car, via an instrument (nonhuman enabler) link (the line ending with a blank circle). The relationship between Driver and Car is expressed by the "is inside" tagged structural relationship between Driver and Car, and the states speeding and stopped are marked respectively as initial and final.

To aid active processing, humans use pencil and paper to flesh out their thoughts. OPCAT accepts pencil strokes as graphic input for objects, processes, states, and links. Pattern-recognition techniques help it recognize these graphical elements, even for sketchy pencil strokes.

LIMITED CAPACITY

Figure 2 is the final System Diagram, the bird's-eye-view model of the system. This OPD contains about seven entities and seven links, pushing the limit of our cognitive capacity, as determined by the "magic number seven plus or minus two" concept [10]. However, we have not yet specified the subprocesses comprising the Emergency Braking process or the parts of ABS. Addressing our limited human capacity, OPM advocates keeping each OPD simple enough to enable the diagram reader to quickly grasp the essence of the system by inspecting the

OPDs without being overwhelmed by an overly complicated layout. Overloading the System Diagram or any other OPD with more artifacts would put the viewer's comprehension at risk, so showing additional detail is deferred to lower-level OPDs.

When an OPD approaches the limit of human comprehension, the model must be refined to manage the system's inherent complexity. Figure 3 outlines the newly generated OPD, labeled "SD1—Emergency Braking in-zoomed." In it, in-zooming Emergency Braking reveals five subprocesses and an interim object. This view is expressed in the OPL sentence "Emergency Braking zooms into Braking, Signal Detecting, Boosting, Anti-locking, and Actuating, as well as Signal Set." The modeler is now able to specify that Driver is the agent (in charge) of the Braking subprocess, and the Actuating subprocess is the one that actually changes Car-Driver System from speeding to stopped. The time within a zoomed-into process flows from top to bottom, so Braking happens first, Boosting and Signal Detecting are executed in parallel, and Actuating is last. ABS is unfolded to reveal its constituent parts (such as Brake Assembly and Mechanical Subsystem), making it possible to express procedural relations between the subprocess Emergency Braking and the parts of the ABS.

ACTIVE PROCESSING

The active-processing assumption is tacitly accounted for during the conceptual modeling process in that each and every modeling step requires the complete engagement of the user—the system architect carrying out the conceptual modeling activity. When modeling, the architect places the conceived elements on the screen (possibly through the pencil tool), linking them and inspecting the OPL textual interpretation that is continuously created in response to new graphic inputs. The architect must from time to time rearrange the graphic layout to make it more comprehensible through such actions as grouping entities and moving links to avoid crossings. If the current OPD is too busy, that means it is approaching our limited channel capacity, in which case a new OPD must be created via in-zooming or unfolding.

Animated simulation is another aspect of active processing. Humans have been observed to mentally animate mechanical diagrams to aid comprehension.

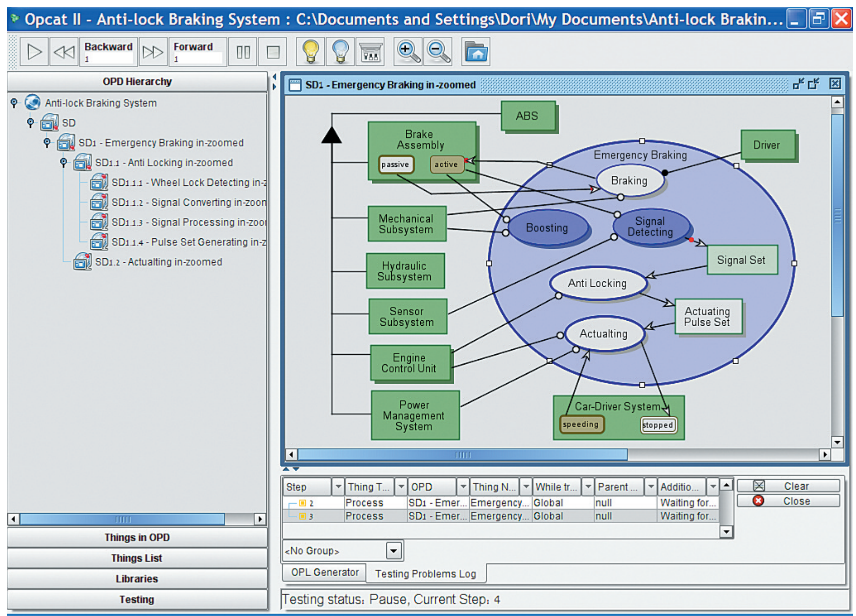

Figure 3. The Emergency Braking process, revealing five subprocesses and two interim objects. This snapshot of the system's animated simulation serves as a design-level visual debugging tool. Shown by red dots (indicating the flow of control), Braking has just changed the state of Brake Assembly from passive to active. Boosting and Signal Detecting are executed in parallel, while Signal Set is generated by Signal Detecting.

Using a gaze-tracking procedure, [7] found that inferences were made about a diagram of ropes and pulleys by imagining the motion of the rope along a causal chain. Similarly, an active-processing aspect of OPCAT is its ability to simulate the system by animating it. This animation enables the modeler to simulate the system and see it "in action" at each point in time during its design. Like a program debugger, the modeler carries out "design-time debugging" by running the animation stepwise or continuously (back and forth), inserting breakpoints where necessary.

Figure 3 is a snapshot of the animated simulation. Objects in green exist at this point; the white ones (such as Actuating Pulse Set) were either consumed already or are not yet created. Blue processes (such as Anti-Locking with Boosting and Signal Detecting within it) are now taking place. The active participation of the modeler (as system behavior is inspected) has proved highly valuable in communicating action and pinpointing logical design errors (corrected early on), saving precious time and avoiding costly trouble downstream.

CONCLUSION

Technological limitations can no longer be cited as an excuse for a lack of human-centered design. A

A conceptual modeling environment should not be merely usable but fun to use, so users want to use it rather than feel compelled to do so because that's what they've been told.

conceptual modeling environment should not be merely usable but fun to use, so users want to use it rather than feel compelled to do so because that's what they've been told. The OPM modeling approach has adapted this philosophy into the OPCAT environment to cater to our cognitive abilities (dual processing), limitations (limited capacity), and needs (active processing). Dual-channel processing is addressed through the model's bimodal representation. Since technically oriented people usually prefer diagrams while others might favor text, individuals with both preference types are able to benefit from consulting one modality while inspecting the other. Domain experts and executives on the customer side should participate in eliciting and analyzing their system requirements, but programmer-oriented modeling approaches and environments bar such involvement.

Devoid of the cryptic syntax normally found in programming languages, the OPM model is understandable to customer-side stakeholders, allowing them to inspect the model, understand the system they should expect, and verify that the model meets their requirements. Switching between graphics and text, OPM system modelers are less likely to make costly design errors, while model readers are more likely to comprehend the system and detect design mistakes or omissions that might otherwise slip by. OPM addresses our limited cognitive capacity by providing abstraction/refinement mechanisms that enable complexity management. Active processing is facilitated by animated simulation that helps detect costly design errors. Further, OPL's formality is a basis for generating the designed application automatically. This capability reduces the modeler's manual translations of the modeled requirements, narrowing the gap between requirements and implementation.

All professions and organizations today demand lifelong learning, so designers of modeling environments must be able to account for the variety of human preferences and learning styles. The holistic OPM paradigm, with its intuitive implementation, is an example of a forward-thinking approach that could be adopted in future modeling and learning environments. Indeed, work is under way for the Object Management Group's scheduled release of SysML 2.0 in 2009 or 2010 to utilize OPM elements, including the addition of textual representations to SysML diagram types in order to achieve bimodal representation for dual-channel processing. **C**

References

1. Baddeley, A. Working memory. *Science 255*, 5044 (1992), 556–559.
2. Chandler, P. and Sweller, J. Cognitive load theory and the format of instruction. *Cognition and Instruction 8,* 4 (1991), 293–332.
3. Clark, J. and Paivio, A. Dual coding theory and education. *Educational Psychology Review 3,* 3 (Sept. 1991), 149–210.
4. Dori, D. *Object-Process Methodology: A Holistic Systems Paradigm.* Springer Verlag, Berlin, 2002.
5. Dori, D., Reinhartz-Berger, I., and Sturm, A. Developing complex systems with object-process methodology using OPCAT. In *Proceedings of Conceptual Modeling—ER 2003, Lecture Notes in Computer Science 2813* (Chicago, Oct. 13–16). Springer Verlag, Berlin, 2003, 570–572, 2003; software available at www.opcat.com.
6. Glenberg, A. and Langston, W. Comprehension of illustrated text: Pictures help to build mental models. *Journal of Memory and Language 31,* 2 (Apr. 1992), 129–151.
7. Hegarty, M. Mental animation: Inferring motion from static displays of mechanical systems. *Journal of Experimental Psychology: Learning, Memory and Cognition 18,* 15 (Sept. 1992), 1084–1102.
8. Mayer, R. The promise of multimedia learning: Using the same instructional design methods across different media. *Learning and Instruction 13,* 2 (Apr. 2003), 125–139.
9. Mayer, R. and Moreno, R. Nine ways to reduce cognitive load in multimedia learning. *Educational Psychologist 38,* 1 (2003), 43–52.
10. Miller, G. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review 63* (1956), 81–97.
11. Stenning, K. and Oberlander, J. A cognitive theory of graphical and linguistic reasoning: Logic and implementation. *Cognitive Science 19,* 1 (Jan.–Mar. 1995), 97–140.
12. von Glaserfeld, E. *The Construction of Knowledge: Contributions to Conceptual Semantics.* Intersystems Publications, Seaside, CA, 1987.

Dov Dori (dori@ie.technion.ac.il) is an associate professor in the William Davidson Faculty of Industrial Engineering and Management at the Technion, Israel Institute of Technology, Haifa, Israel.

DOI: 10.1145/1342327.1342337