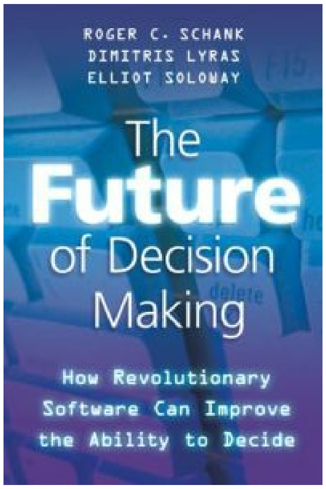

Book The Future of Decision Making

How Revolutionary Software Can Improve the Ability to Decide

Roger C. Schank, Dimitris Lyras and Elliot Soloway
Palgrave Macmillan, 2010
Listen now

- play
- pause

00:00
00:00

Recommendation

Education engineering experts Roger C. Schank, Dimitris Lyras and Elliot Soloway apply their knowledge of cognitive science, computer software and industry to a single purpose: helping you make better decisions. They directly challenge using logic and rationality as they argue for “case-based reasoning.” Their writing is exceptionally clear, which makes the book easy to follow. Its only weakness is that the authors are so positive you can improve your decision making through software that they sometimes make it sound easier than it might be. That aside, this first-rate book’s counsel applies to all fields. *BooksInShort* particularly recommends it to corporate leaders, software designers and those involved in knowledge management.

Take-Aways

- Most people make bad decisions because they use the wrong decision-making models.
- You’ve learned to make decisions logically and rationally, but decision making is an “unconscious process” based on your experiences.
- Most organizations provide rules to guide decision making, but these work against you.
- The best way to help employees make decisions is by letting them learn from experience, including failure. The second best way is by sharing expert stories.
- Interview your veteran employees, as well as former staffers, outside experts, academics and consultants, to build a storehouse of useful stories.
- The right knowledge management software can actively help you make better decisions by providing you with the right stories at the right time.
- Good decision-making software is organized like the human mind.
- You need software that is specific to your organization, industry and activities.
- Truly useful software functions like a friend who knows you well, taking your character and emotional makeup into account.
- Properly designed simulations can train people to make better decisions.

Summary

A New Model for Decision Making

Everyone in your organization wants to make good decisions, but many people make bad ones because they’ve learned the wrong model for decision making. They count on logic, data and “conventional wisdom.” That approach may work in simple situations where information is readily available, but in most instances, it works against you because “decision making is largely an unconscious process.”

> “Many organizational employees make well-intentioned but poor choices. This is true for everyone from top executives to line workers.”

To begin to improve your decision making, discard the idea that you're a rational decision maker. Like all human beings, you make your best assessments intuitively, using a feeling, a "gut instinct" or a hunch. Your best judgments are "based on accumulated experience." When presented with an issue, your subconscious mind reviews related situations from your past, the choices you made and how those worked, and then it generates new options. The finest decisions "flow from goal conflict adjudication" – juggling several conflicting factors at once – rather than "half-baked logic." For example, a captain at the helm of a freighter traveling through the Suez Canal knows it's better to push through than to stop and deal with an onboard boiler fire. Why? The captain's experience has shown that the expense and trouble of halting the ship, negotiating with local authorities to release it and missing delivery deadlines are greater than any potential fire damage. You learn to make better decisions by making decisions and learning from the results – both good and bad.

"Case-Based Reasoning"

In organizations, people may think they make decisions collaboratively, but what they're really making is "a series of small individual decisions." Even if they make choices as a team, they're unlikely to involve the right people at the right time with the right experience. Companies often give their employees decision guidelines, but "rule-based logic" is of little help. Rules are well-intentioned, but they derive from past, static conditions, within known parameters and limited variables. But business is dynamic; situations change all the time, so generalizations based on the past often don't apply to the present. Replace rigid rules with case-based reasoning, which is founded in individuals' experiences, whose emotional power has greater impact on the psyche than abstract rules do. Case-based reasoning draws on a broad variety of experiences, both successful and not, and identifies which ones apply to current situations.

> "Experience really is the best teacher. An experience is often emotional, and thus has a greater impact than a purely cognitive rule."

In the maritime world, veteran sailors are called "old salts" because they've been through everything on the water and they know just what to do in an emergency. Call on your organization's old salts to help in decision making. However, you can't afford to hire people with only this level of expertise. Instead, capture their proficiency in story format through interviews. Stories are "the next best thing" to direct experience. Stories bypass your relatively inefficient logical mind and activate the wisdom of your unconscious. Your mind continually indexes your experiences, synthesizing events into patterns; case-based reasoning will allow you to tap these insights in ways that rules do not.

> "Organizations loathe the idea of failure, but failure is integral to on-the-job learning."

Applying case-based reasoning to corporate decision making "brings objectivity" to the process, discarding the "pet theories" that frequently cloud judgment. Case-based reasoning improves your hiring and retention process: In recognizing the value of experience, you'll position your old salts where you can tap their know-how. Too often leaders have a lot of experience, but not the right kind for making "high-level decisions"; case-based reasoning can supplement this dimension. It will also help people master their responsibilities faster and take smarter risks. And making decisions this way has intangible benefits: People will feel better about their decisions.

Automating Decisions

Companies should transition gradually to case-based reasoning. The right decision-making software (which, in this context, can mean anything from information systems like email to structured applications like those for databases and purchasing) can help. But many times, generic software is complicated and not user-friendly. It can also be "object-based," good at tracking budgets and files, but not at registering the activity-based, dynamic, decision-making variables specific to your field or company.

> "Companies must develop industry-, organization- and process-specific software as well as capitalize on the common sense of the people with situation-specific experience."

Software that incorporates the principles of cognitive science uses "indexing language" to organize expertise into useful, accessible forms. The indexing system that software uses should align with the four groups into which the human mind classifies information: "goals, plans, goal conflicts and conditions."

> "Expert stories are only worth as much as their ability to be delivered in a timely manner to the people who most need to hear them."

As an example, consider the differences between the musical *West Side Story* and the play *Romeo and Juliet*. They take place in different centuries, places and situations. Yet they are indexed in your mind by "the same label: 'Warring factions destroy lovers' plans and lovers themselves'." Similarly, one colleague's complaint that his wife overcooked his steak last night triggered his friend's recollection that he could never find a barber to cut his hair properly 30 years ago. The common label? "Extreme request."

Stories Feed the System

Software can remember stories forever and objectively, without emotional bias. In addition, your software should remind you of the most appropriate stories at the most useful times, filter out excess data and connect your "decision makers with experts."

> "Some of the most interesting and instructive stories an expert can tell are related to mistakes he has made."

Populate your decision-making software with stories from your old salts. Which veteran employees have been through it all? Ask them to discuss extreme experiences: the best and worst decisions they ever made, deals that looked great on paper but didn't work out, big risks that had great payoffs, and so on. Request that they spell out the logic behind their decisions, what the negative results of their actions were, what they learned, and more. In all stories, push for specific details, and help arrange the material into a solid structure with a beginning, a middle and an end.

> "Most of the email systems used in organizations are not as smart as even the slowest-witted, most inexperienced secretary."

Then, collect expertise in specific areas: Who is the most proficient at cutting the budget, at fixing kinks in the supply chain, at innovation? These old salts won't have "an unbroken string of successes" on their records; they've failed at times, but they learned from their mistakes. If you don't have these expert employees in your company right now, fill the gaps with former staffers, academic experts and outside consultants.

> "The difference between traditional and cognitive training is the difference between telling employees how to make good decisions and letting them practice making them."

Once you've captured such knowledge, your software can begin to recognize it as similar to current challenges and link them together. The link might happen at a basic level, such as reminding you of past decisions related to the same subject matter, but software could also link saved stories and current situations at a more abstract level. For example, an HR officer relates her children's desire to stay up late to the issue of her company laying off older workers. Underneath each is a hidden need – the kids want to watch a particular night's TV program, while the employees want the respect their jobs afford them. This "cross-domain" thinking leads her to the right solution: The children can stay up late one night per week, and the older workers will be hired as part-time external consultants. In the same way, artificial intelligence can bring up seemingly disparate occurrences and find the commonalities in them to get to the right answer.

"Decision-Making Software"

Organizations often become buried in information, so your knowledge-management software should bring you exactly what you need, even things for which you didn't think to search. This increased capacity should extend even into your email system. For example, your software could sort the organization's messages by imposing a basic structure on the emails employees compose. Rather than writing emails as "free-form text," structure them into "tasks, plans and goals" to make indexing easier. Just as personal assistants know their bosses' likes and dislikes, so, too, can email act as a "prioritizing, reminding and coordinating" resource.

> "Emotion can serve decision makers well."

This greater functionality can apply to search engines as well. Right now, even the best programs search linguistically, generating many false hits (such as information on President Gerald Ford when you're searching for Ford cars). Search engines, like other software, should index material based on how the mind works. Search functions should be "computer-intuitive," too and generate searches based on whatever you're doing on your computer.

> "Computer users sense when software feels right. Something inside of them says that the software is asking them questions, reminding them of stories or providing other knowledge that is exactly what is required."

Tough decisions stymie executives because choices often involve "underlying goal conflicts." For example, a CEO must choose between cutting staff to save costs and to please investors, or retaining employees and investing in "the future of the company." Well-designed software can help negotiate these clashes by providing information about similar divergences in the past that can aid present-day decision making. The software must allow for "modification" to incorporate new experiences as they happen. It must also track failures, so that you can gauge what went wrong and store those evaluations to help your corporate leaders in the future.

Implementation

A lot of software is functional: The screen is attractive, the icons are clear and the applications open smoothly. But that's "level 1 usability," where software functions as a tool, as in, for example, Microsoft Word. The goal is to produce "level 2 usability," where software plays an active role in "facilitating the process of making a good decision." For this sort of operability, the "visual metaphor" of the "desktop" takes software in the wrong direction: It organizes information like an office does, rather than like a mind does.

> "Cognitive-based software is well-equipped to adjudicate through its communication capabilities. This is true whether or not decision makers are aware of these conflicts."

Your decision-making software must customize itself to each user. It should guide you by offering "contextual knowledge," examples, suggestions, guidelines and the like. Your software can function like a friend who knows you well, taking your character and emotional makeup into account. It should know your short- and long-term goals, how they conflict, what decisions you've made, and what obstacles you're likely to encounter.

> "Decision making isn't about right and wrong. Training that attempts to convince people to follow decision-making rules won't work."

Truly useful software should provide interfaces in which graphics reflect how the mind actually functions. An interface should use "cognitive abstractions" for each activity mapped – that is, breaking down jobs that have different titles to expose underlying similarities. Likewise, it should demonstrate an "understanding of enterprise processes." And it needs to winnow the results so it doesn't bury users in data. Make your software respond to each "user's role within the larger process" so that it interacts with experienced engineers differently than it does with new marketing hires. Software should put all interactions in context, yet be able to shift quickly from topic to topic. It should link all "related processes" and reflect how your organization functions.

Training in Decision Making

You can use the same software-design principles to help people learn to make better decisions. Recognize that you cannot teach people to decide better, however, you can give them practice so they can learn by doing. To do this, use software to provide "realistic decision-making simulations" in which learners can practice all aspects of decision making, including failing.

> "Designing usable software for decision makers isn't rocket science; it's brain science."

These simulations should include "expectation failure," in which people make decisions that appear good but turn out badly, and "ambiguity," to allow learners to practice realistically complicated decisions. These simulations should be "goal-based" for participants, yet fun enough to engage them.

As you develop software to train your employees, ask them which decisions frequently give them trouble. Solicit stories about bad decisions and their outcomes, about circumstances that have led employees to the wrong choices, and about the different kinds of judgments that most often cause problems. Determine what "employees need to know how to do to make good decisions."

Expect obstacles and objections as you implement this training. You can't design good training courses on making specific decisions, so those who want such classes will be frustrated. Instead, offer practice in decision making, feedback on the process, time to reflect and guidance. Set a brief series of "performance objectives," and evaluate whether the training meets those goals. Finally, recognize that training is beneficial, especially in helping "employees change certain perceptions or habits," but it can't be a cure-all.

About the Authors

Roger C. Schank is a professor emeritus at Northwestern University. **Dimitris Lyras** founded Ulysses-Systems, a software consultancy. **Elliot Soloway** is a professor of engineering, information and education at the University of Michigan.