by **Mihnea Moldoveanu**

Integration(s): The Value of Managerial Thinking in an Age of Technical Reason

In an age where the digital machine can outperform the human mind at any task that can be represented as an algorithm, Integrative Thinking is the best revenge.

DOES THINKING MATTER TO SUCCESS IN BUSINESS? If so, in what ways? Consider the null hypothesis: 'the way to success in business lies in hustle and luck – not necessarily in that order: a lucky, thoughtless hustler will outperform a deep, sophisticated, disciplined thinking doer every time.' If this were the case, the stories we delight in telling about how great business minds are causally implicated in the achievement of great business results are no more than made-up inspirational talk aimed at justifying a costly educational enterprise – which might far more efficiently devote itself might to training hustlers and hope for good luck, or, co-opting luck by training lots of hustlers and letting the chips fall where they may, than to training 'thinking doers.' Which is it?

Appeals to 'evidence' will not turn up easy answers. Most of empirical Psychology can tell us whether or not individuals obey certain rules assumed to be useful to the achievement of success in certain environments, although *that* assumption never really gets tested. And most anecdotal evidence tells us that people exhibiting certain traits have achieved certain results – without answering the obvious question that it begs: what happens to the successful individuals who *do not* exhibit these traits and the unsuccessful individuals who do exhibit them?

Disappointed by the hallways of empirical Science, the only alternative to despair is to build *new models* of the role of thinking in business – models that allow us to ask new and sharper questions which, just maybe, *are* answerable by empirical studies. One possible approach to building such new models is to *think* – to really think – about the position that thinking has in an age where the marginal cost of a calculation is $0.

It used to be the case that we would declare someone who could quickly and correctly multiply 12674 and 37993 as 'smart' and someone who could 'only' multiply numbers of fewer than *4* digits each as less smart. In the age of technical computing, these differences seem hardly worth mentioning. They certainly cannot be held up as paradigms of 'managerial intelligence.' And the fast computation of sums is just a barely significant beginning: the digital machine *will* outperform the human mind at any task that can be represented with the level of precision of an algorithm – that is, at any 'algorithmic task.' Moreover, if the marginal cost of the basic component of an algorithmic task – the operation – is $0, then, clearly, $0 is also a floor on the value that can be appropriated by a managerial mind that behaves in purely algorithmic ways.

This seemingly innocuous argument supplies a basic logic for the pursuit of connections between thinking and value creation: we must look away from purely algorithmic capabilities. OK, but, where should we turn our gaze? 'Away from' something is not exactly a valuable pointer to anything. What to do? When in doubt (and when empirical Science cannot help), we can follow the philosophers and do a thought experiment along the following lines: take a super-normally profitable business production function – one that generates at least 50 per cent gross margins on revenues of more than $100MM, and start *subtracting* managers from it, out of various teams and at various knowledge-power levels within each team, and replace those managers by algorithms that are executed by computational devices, in order to determine the value added of the respective managers' tasks and ways-of-being in executing those tasks. Once you have figured out the tasks that cannot be subcontracted to digital machines without losing the super-normal profit of the business, try to figure out if there is something that these tasks all have in common.

Let us attempt this exercise: first, focus on a super-normally profitable production function – like that of a large telecommunications equipment manufacturer (**Figure 1**); and focus, in addition, on a specific project that is part of this production function, such as the development of a next-generation cellular base station (i.e. the piece of network equipment that your Blackberry or cell phone talks *to*). Then, break up the task of designing and prototyping this monster into the specialized tasks required to complete the design, and identify the areas of specific and explicit expertise required to carry out these tasks. These will include hardware, radio frequency (RF), software and network design, and analytical expertise of various kinds (market demand, financial reporting, financial value, competition). Each individual area of expertise can be characterized by more or less well-defined algorithms – or, analysis and decision procedures for producing outputs (market demand estimates, cost structures, block diagrams, etc.) from inputs (market data, input costs, system models). Each one of these algorithms can be executed by an 'expert' – of the type we would find in a cross-disciplinary product team in firms such as **Nokia** and **Alcatel-Lucent** – or, by a collection of low-skill workers employing some central coordination mechanisms (as we increasingly find in design projects that are outsourced and off-shored).

Now, here is the important point: within each area of explicit expertise, tasks are more or less algorithmic in nature, and therefore those who carry them out are more or less easily replaceable by a computational device, because of the fact that each area is based on a specific discipline (see **Figure 2**) that supplies the equivalent of a 'unified programming language' (like C++ and MATLAB, only with more adjectives and adverbs than either one.) Knowledge of programming languages and environments and of operating systems, for instance, provide, jointly, a unified programming language for the execution of Software Engineering tasks. The most important feature of such a unified programming language is that it

Figure 1

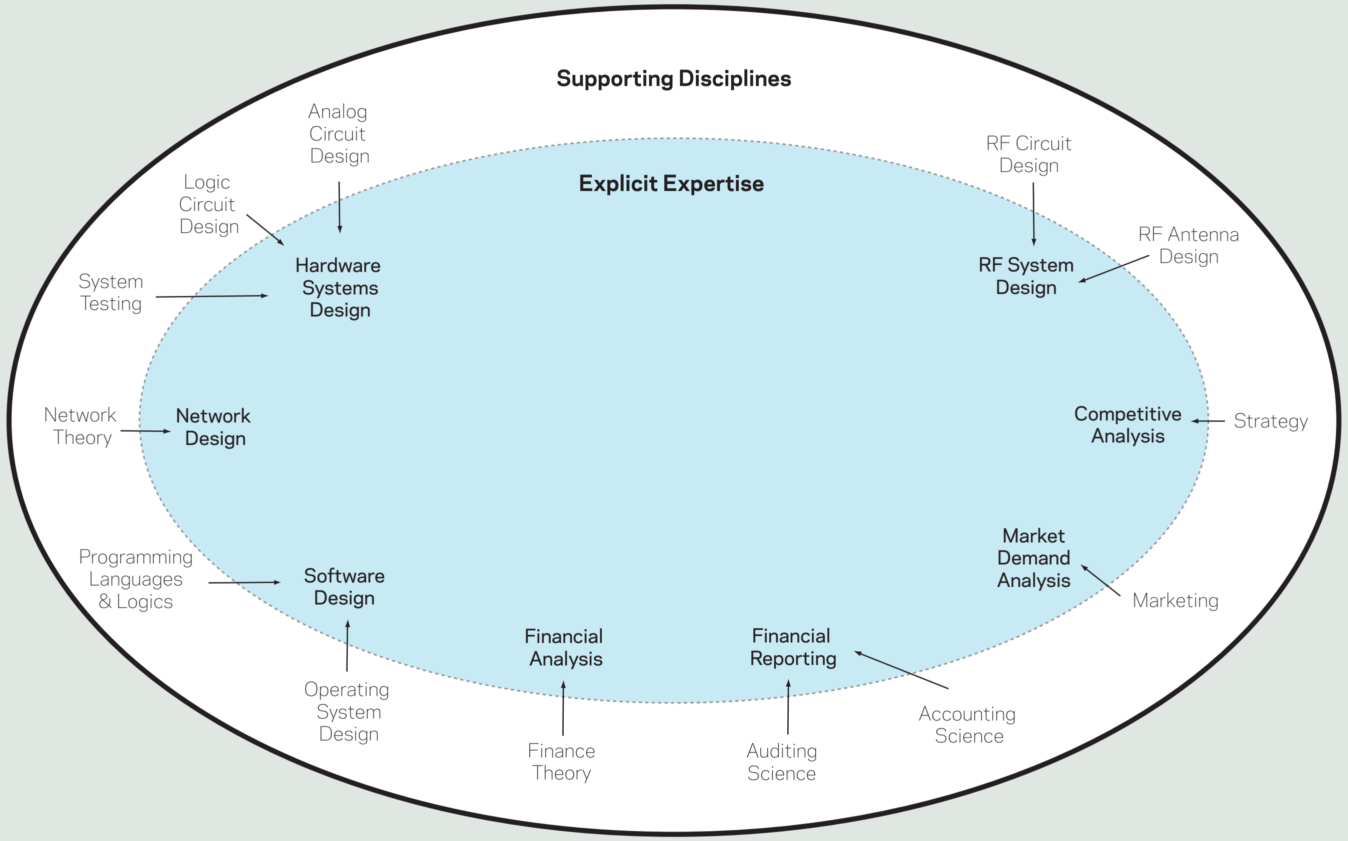
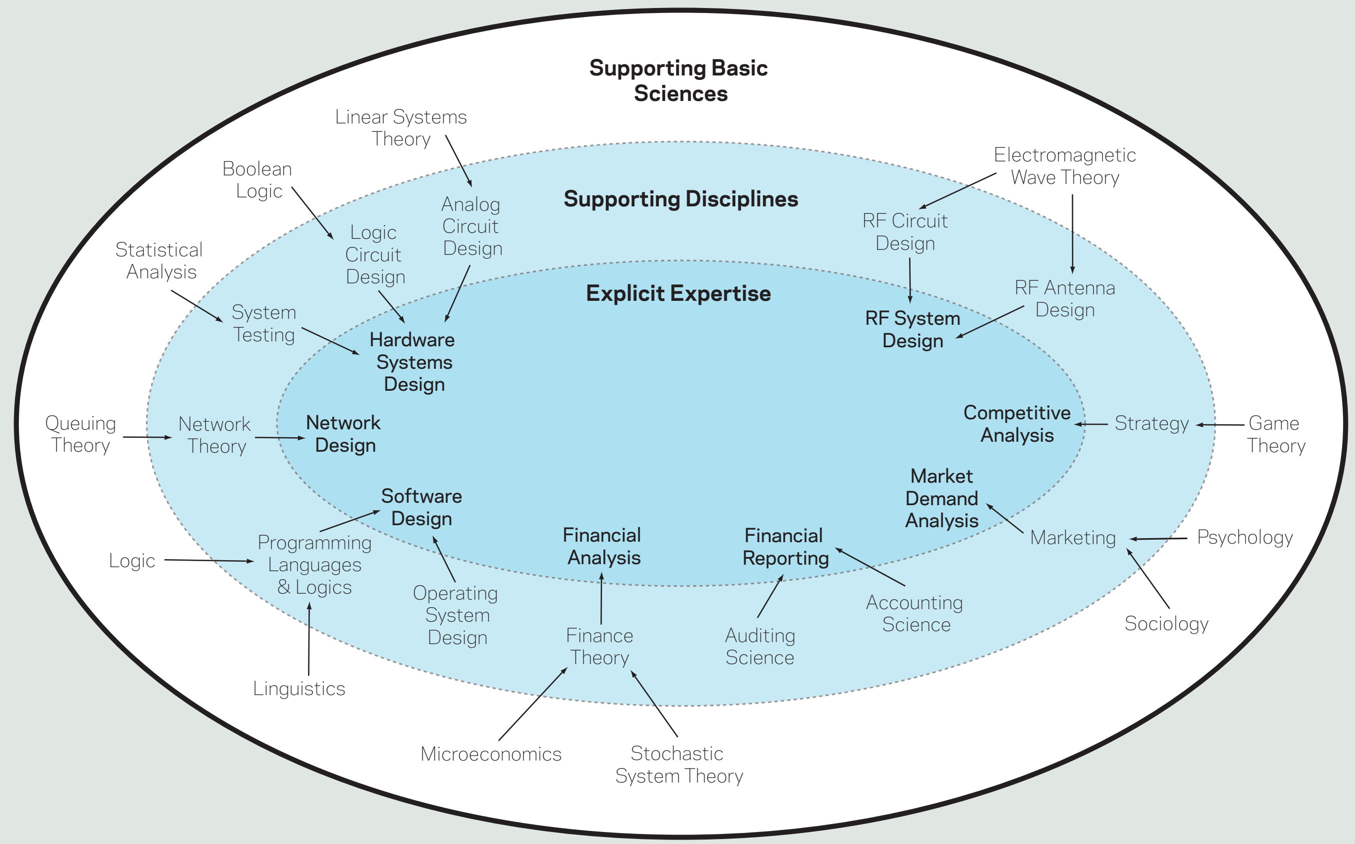


Figure 2



supplies not only a common – and, commonly agreed upon – set of concepts and rules for resolving conflict, uncertainty and ambiguity, but also a set of 'stopping rules' that define solution criteria for problems within each area of expertise.

Simply put, the hardware designer 'just knows when to stop' working on a problem, and go from design work to testing or prototyping work. How does he know this – within some reasonable tolerance? Well (see **Figure 3**) he knows this because his expertise (hardware design) is embedded in a supporting set of disciplines (analog and digital circuit design and testing) that are themselves embedded in a supporting set of basic sciences (linear system theory, Boolean logic, statistical analysis of experimental results) whose main task is to supply a common and internally consistent set of patterns of representing and reasoning about the world, which, together, tell the expert in question how to regulate his thinking processes in a way that yields reliably 'valid' results. Of course, these basic sciences are themselves inhabited by individuals whose main task is to turn 'reality' into tractable mental objects and knowledge structures that support algorithmic reasoning of one kind or another.

So, where does this bit of cognitive Archaeology leave us, relative to our task? It leaves us in a rather informative place, in that we have identified in a precise way the parts of the super-normally profitable production function that builds cellular base stations that is algorithmic in nature – i.e. that admits of replacing experts with advanced computational devices, or, with lower-skilled workers coordinated by a central information sharing system. It also lets us make a good guess about the locus of 'really valuable thinking' in our map: it lies at the intersection of radically different areas of expertise – at the very centre of our map (see **Figure 4**).

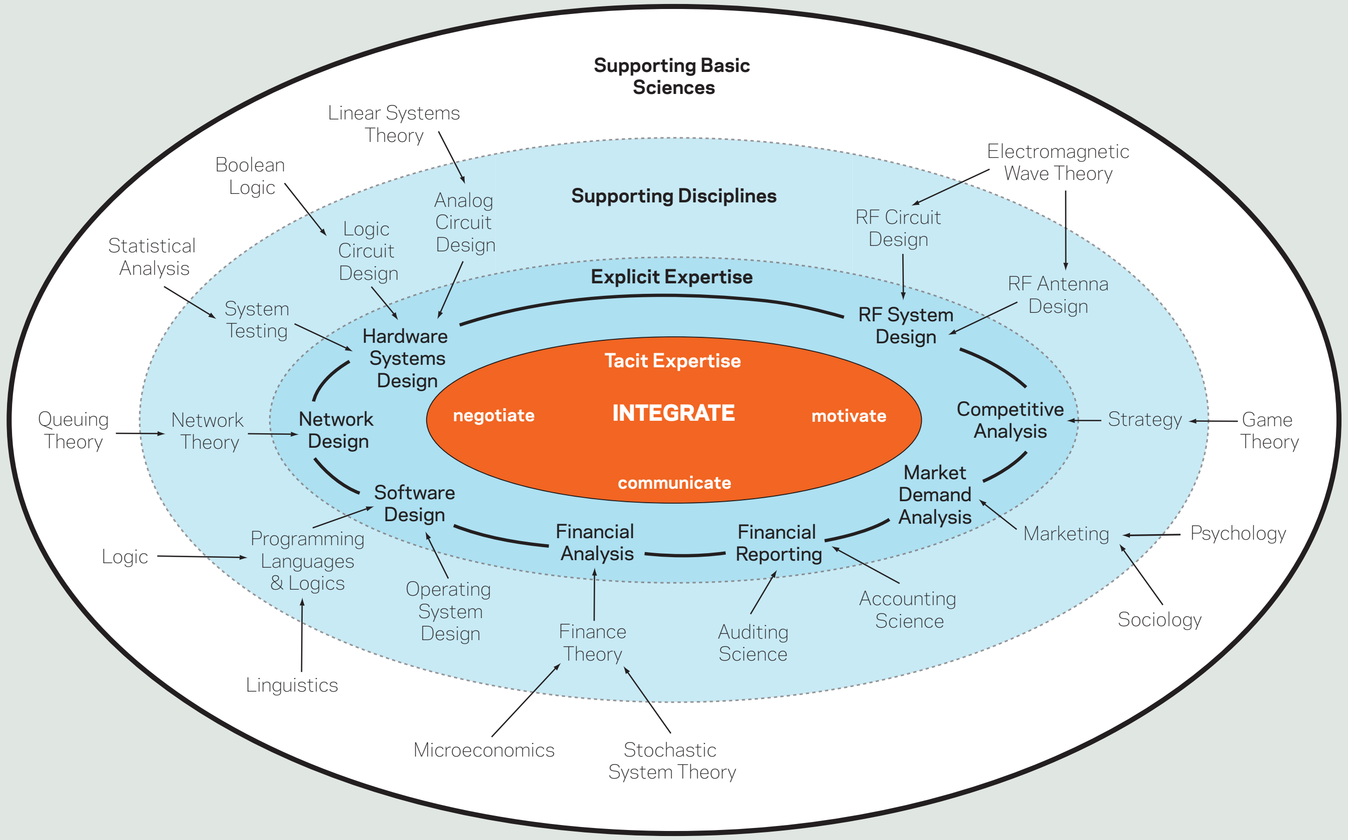
Such 'really valuable thinking' is precisely what **Roger Martin** and I call Integrative Thinking – that part of thinking which *cannot* be captured by an algorithm because it has to do precisely with the successful integration of knowledge structures, mental models and patterns of reasoning and communicating that serve as the very foundations for radically different *kinds* of algorithms.

For example, you *can* find an algorithm for optimizing your inventory, given a set of cost and demand conditions; you can also find an algorithm for optimizing your design given a set of client-driven feature requests; but you cannot find an algorithm for optimizing both together, because the two algorithms are written in different languages, with different sets of optimization criteria and stopping rules. Accomplishing that entails integrating across domains of knowledge and experience that were not designed to 'talk to each other'; so, the integrator at the centre of our map of the base-station production function has to re-design and re-engineer – in real-time – the basic interaction patterns that will enable the successful cross-domain synthesis required to deliver the super-normally profitable output of the business unit.

In particular, three types of Integrative Thinking stand out for the value they create:

1. Substantive integration: the integration of different specialized languages and mental models of the business. First, the integrator will be called upon to produce constructive resolutions of clashes among the different mental models that organize the perceptions, thoughts and actions of experts in the different functional areas (**Figure 5**). To the CFO or the controller, a development project will look like a series of discounted cash inflows and outflows with precise time-lines and penalties associated with risk, uncertainty, ambiguity around deliverables and time lines. To a development engineer, the same project will take on the representation of a potentially non-linear optimization problem, with constraints supplied by the target cost of goods sold and technological limits and objectives supplied by product feature sets. To the software engineer, the project will look like a large scale search problem, with the search space constrained by the syntactical properties of the programming language, the functional constraints of the operating system and the structural constraints of the hardware platform. Thinking integratively in this case relates to thinking that bridges between these different mental models of the task in a way that heeds the concerns and constraints of all of the specialized experts that are called upon for the successful execution of the design and development task, by building and legitimating new representations of the task that are agreeable to all of those whose contribution is needed for its success. This is not merely equivalent to the task of successfully translating the buzzwords and concepts of one area of

Figure 3



expertise into those of another, because in many cases the very criteria of what counts as a successful translation will vary from one discipline to another, and, because each discipline is reductive in its own idiosyncratic way, there is no guarantee that experts from two or more disciplines will recognize a single translation as legitimate. Moreover, what counts as a valid problem statement and as a test of 'valid knowledge' will vary from one discipline to another: engineers may accept nothing but uniqueness or optimality proofs relying on deductive logic alone as 'clinchers' of difficult argument and valid members of solution sets for tough problems, whereas a marketing professional will often count stories, anecdotes and partial data as 'evidence' in arguments that rely on inductive and abductive logics to make their points. For this reason, the task of integration additionally involves bridging across disciplinary divides (**Figure 6**) that relate to much more than differences in words, but, rather, cut through to differences in ways of formulating problem statements and solving problems.

2. Interactional integration: the integration of different modes of interacting and communicating. As if that is not difficult enough, the integrator's task extends to a very different set of models, which require her to consider a large class of other 'differences that make a difference.' These are differences among different ways of interacting that members of different areas of expertise and disciplines bring with them as part and parcel of their professional identities. Teams, groups, divisions and organizations may be organized according to principles of authority ('who's the boss?'), fairness, or equity ('is everyone getting his/her fair share of the payoffs or opportunities?'), efficiency ('is everyone getting the most out of the task given their outside market opportunities?') or welfare ('is the entire group, team or organization better off as the result of each of the actions of all of its members?') – which, collectively, supply different *logics of interaction*, which often remain implicit and shape dialogue and interactions 'from the darkness of the barely conscious.' Whereas financial professionals may interact according to logics of efficiency or authority, scientific experts will bring along with them interactional logics based on maximizing the knowledge or expertise of the group as a whole ('maximal welfare') or, alternatively, based on minimizing the perceived differences in payoffs to individual members of the group ('equal opportunity'). These interactional models – as specific to various teams and organizations as they are to different professional groups and sub-groups – will often shape expectations of different group members about the pragmatics of the activities ('meetings', 'design reviews', new product introduction processes, market requirements document production) that constitute the very fabric of the overall production function of the organization, via such questions as: who gets to speak first and why? What arguments are openly discussible and which must remain unstated – and, why? What is the role of organizational rank in discussions about technical merits – and why? The net result is that the integrator must formulate and solve – locally and in a way that algorithm can *ex ante* specify – a host of 'wicked' problems, so called because the criteria for what counts as a solution evolves along with the process of searching for that very solution.

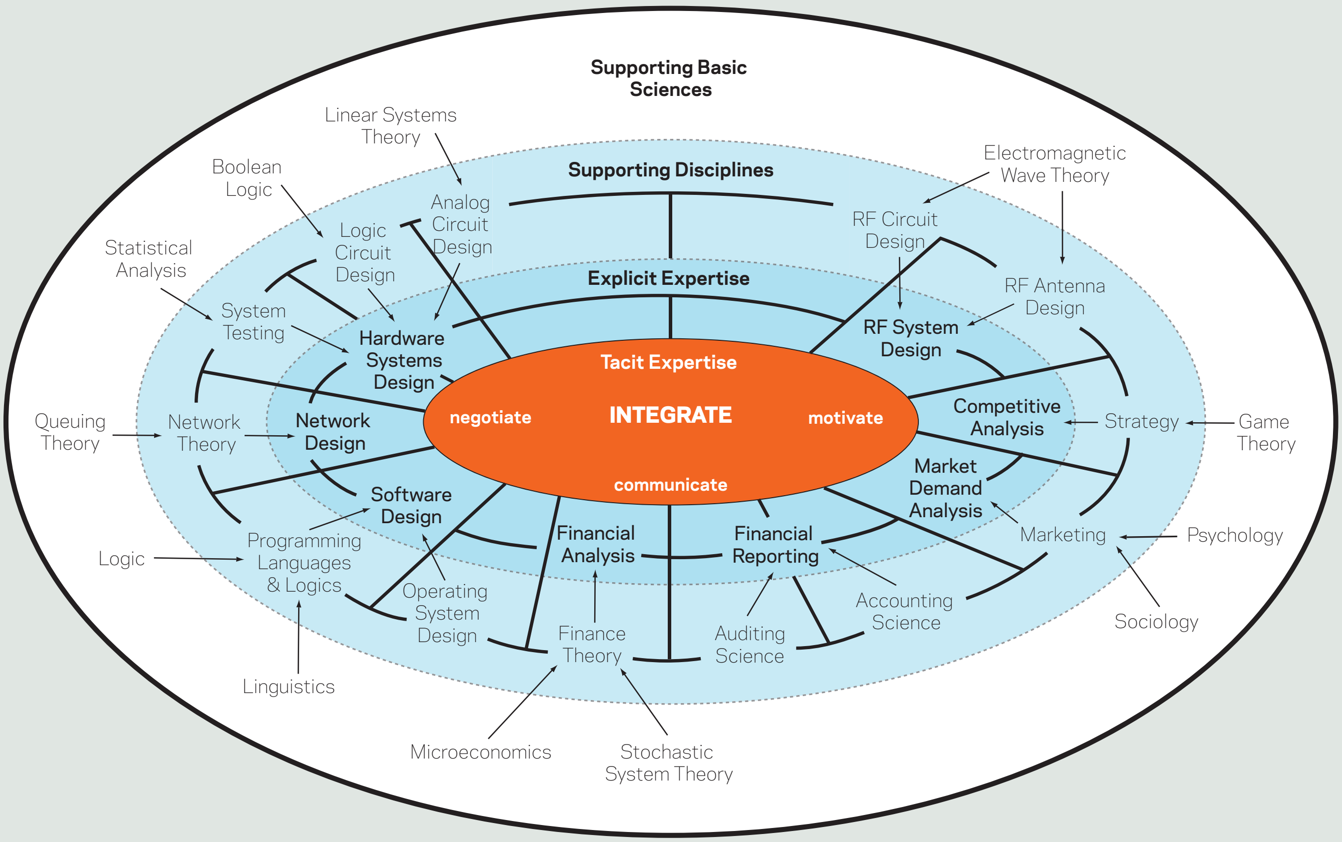
There is no single set of codes or rules or single hierarchical structure that the integrator can simply defer to.

3. Agency: the integration of seeing, feeling, thinking and action. Last but certainly not least, the high-value-added thinker provides a solution to what is perhaps the ultimate integrative problem: the problem of agency (**Figure 7**). The best way to get a feel for this problem is to ask yourself: why – given the great proclivity of the human intellect to commit fallacies of reasoning and perception *vis-à-vis* the 'normative' workings of a digital computation device – do we rely on human experts for problems that 'cut really close to the flesh' – such as medical, legal and managerial or financial ones? Why do we not sub-contract that thorny problem of making the right diagnosis that distinguishes between a benign atrial arrhythmia and a potentially fatal one to a large computational device that can *really* perform all of the correlations required for a statistically-relevant finding? Why trust to the so-called mind of a physician who depends so much on his own sleep patterns and a large host of other visceral factors for making the right call?

I am not excluding the possibility that this *may* be the way of the future, but, for now, it is clear that, when results really matter, agency – and the responsibility assignment that it entails – also really matters. The task of claiming and establishing agency is ultimately one that a digital device cannot perform and that the execution of no algorithm can safeguard. It is also a task that is critical to he who must stand outside of any disciplinary language, safeguard, or pillar in order to effectively integrate across multiple ways-of-being-and-knowing.

Our integrator must, then, function as an *agent* – as one that shoulders the responsibility of his or her own perceptions, feelings, thoughts and actions and thus foregoes an appeal to the therapeutic cocoon of any one set of professional 'codes of conduct' ('I'm covered!'), interaction rules ('I did the right thing, outcomes notwithstanding!'), or hierarchical rank ('I was told to do it!'). The reason for this is simple enough to state: there *is* no single set of codes or rules or single hierarchical structure that the integrator can simply defer to – that is precisely why he is an integrator. *He has no 'air cover'* – to put the matter in military terms: ground work is where his job is at. And, to come to grips with it, he can draw inspiration from no less than the intellectual forefather of all integrative thinkers – **Aristotle** – who, besides positing that the faculty of reason is the queen of human faculties in virtue of being their *integrator* (!), also cautioned about the difficulties of establishing real agency when he admonished that 'to feel angry is easy; but, to feel angry at

Figure 4



the right person in the right moment for the right reason – that is not so easy.' Yet that is precisely what the establishment of agency requires of the integrative thinker.

In closing

To sum up: there is – I posit – a particular kind of thinking that matters to success in business, but, figuring out what it is and how to do more of it requires a new conceptualization of valuable thinking in terms of its *integrative* functions – the functions that produce successful integration across knowledge and interactional domains and confer agency upon decision makers.

Can these forms of thinking be taught? The answer is, 'stay tuned', but, here is a hint: transferring thinking skills across domains of experience (for instance, getting professional logicians to think logically about their own medical problems and their diagnoses) is an endeavor that is fraught with failures, which should tell you something about trying to 'teach thinking' in a vacuum. However, appropriating new ways of thinking by experiential immersion in new domains of expertise is not, which opens up a vast untapped, realm of pedagogical possibilities. ***R***

Mihnea Moldoveanu is director of the Desautels Centre for Integrative Thinking and an Associate Professor of Strategic Management at the Rotman School of Management. He is the co-author of *Master Passions: Emotion, Narrative and the Development of Culture* (MIT Press, 2002) and the forthcoming *Designing the Thinker of the Future: The Future of the MBA* (Oxford University Press) as well as founder, past CEO and CTO of Redline Communications, Inc/, the world's leading WiMax equipment manufacturer and a Wireless Fast50.