Understanding Complexity: A Model for Characterizing a Sequence of Design Projects

Dr. Nicky Wolmarans, University of Cape Town

I am currently an "Academic Development Lecturer" in the Civil Engineering Department at the University of Cape Town. The position involves curriculum development aimed at improving student performance and experience in engineering. This has directed my interest in graduate preparedness and led me to focus on design both at first and final year, and also how reasoning between the concrete and abstract can be implemented in disciplinary subjects.

Dr. Jennifer M. Case, Virginia Tech

Jennifer Case is Head and Professor in the Department of Engineering Education at Virginia Tech. She holds an honorary position at the University of Cape Town. Her research on the student experience of learning, focusing mainly on science and engineering education, has been published across a range of journal articles in higher education and her recent book, Researching student learning in higher education: A social realist approach published in 2013 by Routledge. She holds an academic development post in the Department of Chemical Engineering at UCT, and teaches in the undergraduate programme there. She is a coordinating editor for the international journal Higher Education and a co-editor for the Routledge/SRHE series Research into Higher Education.
Understanding complexity: A model for characterizing a sequence of design projects

Design courses have always played an important role in the curriculum as preparation for professional practice, as far back as the Mann report in 1918 [1], and the Grinter report in 1955 [2]. Both reports call for strengthening the fundamental sciences and the need for complimentary studies (in communication, teamwork and management). But both argue that while engineering sciences are critical to engineering education they need to be taught for the purpose of better industrial performance. Both reports recommend the need for more practical projects and design projects in the curriculum in order to integrate across the sciences and to develop the link between the sciences and their application in practice.

With the shift towards competency-based accreditation of engineering degree programs the focus in design projects has shifted away from design as a link between engineering sciences and their application in practical problems. However it should be noted that while more recent reports [3, 4] recognize an improvement of the performance of graduates in generic competencies, most still struggle to use theoretical knowledge in practice:

"Although industry is generally satisfied with the current quality of graduate engineers it regards the ability to apply theoretical knowledge to real industrial problems as the single most desirable attribute in new recruits. But this ability has become rarer in recent years - a factor which is seen as impacting on business growth" [3, p7]

Therefore, although we locate our paper in the body of research in engineering design, rather than focusing on the development and assessment of generic graduate attributes, our emphasis is on knowledge in application. The paper therefore focuses on ABET outcome (a), ‘the ability to apply knowledge…’, which we argue underpins especially outcomes (c), ‘the ability to design a system, component or process to meet desired needs …’ and (e), ‘the ability to identify, formulate, and solve engineering problems’[5, p3]. We recognize that a focus on knowledge alone is inadequate to prepare students for professional practice; that viewing design as the application of science to solve problems, and design in the curriculum as equivalent to professional practice [6, 7] provides an impoverished conception of both design and practice. Nonetheless, we argue that progressive education models, while espousing education for a knowledge society have ironically tended focus on generic competencies often at the expense of specialized knowledge [8-10]. Rather we contend that relative mastery of specialized knowledge and the confidence to draw on it is the foundation of the other competencies needed in engineering practice [11]. And most significantly for this paper, we argue that learning to recruit abstract specialized disciplinary knowledge in contextually embedded scenarios is a skill that is both difficult and overlooked in professional education [12, 13]. As Abbott [14] notes, knowledge in the abstract form (as it is typically taught in engineering science courses is disassembled and organized fundamentally differently than the way in which we understand the world of ‘things’. Bridging the two is fundamental to aspects of design.

In this paper we present a conceptual model based on Legitimation Code Theory (LCT) [8] that can be used to categorize the complexity of the requirements of a design project. Based on a view of education as a progressive process of accumulating, relating, and integrating professional knowledge and skills, we suggest that learning to design should also be considered a scaffolding process from engaging in simpler projects requiring fewer ‘bits’ to more complex projects requiring more insight. By articulating a systematic understanding of complexity in engineering design projects we offer one way of thinking about constructing a trajectory of design from simpler to more complex projects. The model that we propose is presented at a level of abstraction that is based on generalizable principles rather than on
empirical examples. In this way we suggest that it can be used to construct a design brief at an appropriate level of complexity for the position of the design task in a sequence of tasks. An appropriate formulation of the brief will depend on the position of the task in the learning trajectory in relation to previous and future tasks.

In light of the above, although we did analyze what were called ‘design projects’ in two engineering curricula, the term ‘design project’ was used quite liberally. We identified projects labeled as ‘design’ projects within the curriculum. They tended to be formulated in terms more ‘concrete’ (embedded in the world) than those questions encountered in typical engineering science courses in which the objects of analysis tend to be simplified and idealized out of the world. The ‘design’ projects tended to require knowledge beyond a single disciplinary specialization and they tended to include a number of decision steps through the process of design. However, many did privilege specialized disciplinary knowledge over contextual knowledge. Nonetheless, because our focus was on specialized knowledge used in relation to contextually embedded problems, these sorts of problems were very useful for developing a model of complexity in relation to both knowledge and objects.

We have limited what we present in this paper to a conceptual model of complexity of ‘design’ projects and provide four example projects to illustrate aspects of the model. For the complete presentation of the development of the model and detailed descriptions and analysis of the data behind the proposed model see [15]. The intention behind this paper is to introduce a discussion on progression in design, and to reopen the discussion on specialized knowledge in design. While we acknowledge the limitations of such a study, we nonetheless believe that it is an important and overlooked subject in engineering design research.

Literature Review

With a renewed interest in design in engineering curricula since the 1990s [16] there have been a proliferation of studies on design courses in the engineering education literature [17, 18]. This should be unsurprising when one considers how many skills and competencies are needed to design [19], and how many ABET competencies are assessed in design courses [20]. However, most of this literature focuses on single courses or tasks, usually within cornerstone [for example 21], capstone [see for example the collection of papers in 22], or service learning courses [for example 23, 24]. There is less work on the idea of a trajectory of learning taken by students entering the program with limited or no design experience and very little in the way of specialized disciplinary knowledge.

Crismond and Adams [17] do present a particularly useful view on progression from ‘beginning’ to ‘informed’ designer. Based on an extensive literature review they identified nine dimensions of performance central to design, each with the characteristics of what might be expected at the beginning of a design learning-trajectory and hoped for at the end of a design learning-trajectory. They also suggest techniques and interventions that can assist students to make the shifts needed to progress from beginning to more informed designers. There are also studies that consider the introduction of a particular design thinking tool or heuristic and suggest ways in which the tools might be simplified for introduction in early design tasks, for example Kline, et al. [25] discussed the modification of a ‘design canvas’ for more and less complex use in capstone and first year courses respectively. In their discussion of CDIO, Edström and Kolmos [26] refer to progression in complexity of projects, where complexity increases in multiple dimensions including group size, duration of project, and scope of knowledge required. They suggest that the most complex projects are open ended,
ill-defined, and have contradictory objectives. However, they don't provide much detail on constructing a sequence of tasks.

At a more general level, progression can be considered in terms of various developmental 'stage-models', for example Marra, et al. [27] and Gainsburg [28] used Perry’s [29] developmental scheme, although Gainsburg argued that she did not see evidence of students following a defined sequence of development. Shay and Steyn [30] used a modified version of Dreyfus' [31] five stage model of skill acquisition (from novice, through competent performer to expert) to analyze a sequence of 17 design projects. They used Dreyfus’s stages, with modifications introduced by design thinking researchers. For example Dorst [32] introduced a level before Novice, suggesting Naïve designers employing everyday reasoning in everyday situations. In terms of design Dorst [32] and Lawson [33] describe Deyfus’ Novice stage as the accumulation of discrete design ideas or techniques; an Advanced Beginner as designers starting to recognize contextual significance; Competent designers as starting to draw on precedent and experience; Expert designers start to recognize patterns and types of problems [34] rather than individual problems; and Master designers start breaking the rules using intuition guided by embodied principles. Shay and Steyn [30] interpret this as a progression defined by the compounding of meaning, as more concepts, processes and procedures, past problems and solutions, and a vast reservoir of experience come together in a complex network of knowing.

Using Legitimation Code Theory (LCT) [8], Shay and Steyn [30] used this insight in order to develop a way of characterizing the demands of design projects in a foundation design course. They translated the principles of relative compounding of meaning to categorize their 17 projects which revealed that some of the earlier projects required far more design insight than those appearing later in the trajectory. This enabled them to redesign the sequence of tasks, allowing for a more structured introduction into design. We have also used LCT to develop our conceptual model, but with a more directed focus on engineering design and an associated interest in the specialized knowledge that underpins the profession.

Theoretical Antecedents

The paper is rooted in Social Realism in the Sociology of Education (SRSoE), a theoretical tradition that critiques ‘progressive’ models of education for stripping knowledge from curricula in pursuit of generic outcomes [8-10, 35, 36] and consequently leaving the role of specialized knowledge as an assumed competency. SRSoE argues for a renewed attention to knowledge, the nature of knowledge and the transmission and acquisition of knowledge. SRSoE recognizes the significance of the social context in which knowledge is generated and learnt, but also holds that knowledge emerges in relation to the world, and that some knowledge is more effective than other knowledge. The importance of this underpinning philosophy is that it recognizes that some knowledge and ways of knowing have more power (both socially and in terms of explanatory power). That does not mean that the knowledge is complete or intransient, but it does mean that to build an inclusive society we need to broaden access to the knowledge that counts in society. We need to find ways to extend the knowledge that makes a difference, and the skills to use it effectively, to more people.

This study was undertaken in the spirit of broadening access to the application of specialized knowledge. In order to use the analytical power of specialized disciplinary knowledge in practice, students need to learn to identify appropriate knowledge depending on the particularities of a contextually embedded problem, simplify the complexity of the contextual
detail and translate it into a form amenable to theoretical analysis, translate the results of the analysis in contextual terms, and reintroduce the solution into the complexity of the world. We argue that this is no trivial task, and that all students should be assisted in making the transition between concrete and abstract reasoning and back, rather than relying on the expertise with which students enter the program. We argue that it is problematic to leave this critical skill to chance and assume that students will recruit and translate theoretical knowledge effectively when they encounter their capstone design project. And therefore, we argue that a sequential trajectory of projects that introduce students to knowledge in relation to artifacts and the contexts in which they are embedded is needed.

In this paper we present a conceptual model, based on LCT (Semantics) that can be used to categorize the complexity of the requirements of a design task, and consequently can be used to plan a trajectory of design projects. The model is presented at a high level of abstraction, and the projects presented in this paper are intended to illustrate examples of particular characteristics. They are not provided as an example of a sequence of actual concrete projects. As a conceptual paper we present abstract principles that can be applied to any design project in order to focus student attention on particular knowledge and skills and their relations to other knowledge and skills without overwhelming students with a complex contextually embedded design project with no form of scaffolding or structuring of expectations in preparation for that project.

Legitimation Code Theory: Semantics dimension

LCT is a conceptual toolbox [8] that is aligned with SRSoE, and is, as are we, concerned with the structure of knowledge and the requirements for cumulative knowledge building. LCT consists of a number of dimensions, each with a pair of concepts that can be used to describe the structure of different 'knowledges' [8]. One of the dimensions, LCT (Semantics), was developed in collaboration with Systemic Functional Linguistics (SFL) and looks to describe the relationship of meaning (semantics) to its 'object of knowledge' (semantic gravity), and to the 'network of concepts' that we use to describe the object (semantic density).

Semantic gravity denotes a relative dependence of meaning on the object of knowledge. Semantic gravity is defined on a cline from weaker semantic gravity (SG-) where knowledge is transferable across contexts and examples, to stronger semantic gravity (SG+) where meaning is located in a context [8]. Typically, abstract generalizable knowledge is categorized as SG- because it is intended for use in general ways across as wide range of applications. Knowledge of particular artifacts would be considered SG+, because the knowledge of the specifics of a particular artifact relate to the particularities of that artifact. In their paper on the structuring of design knowledge Dong, et al. [37] propose that design spans a wide range of semantic gravity, from design principles (SG-) to design cases (SG+). This paper has weaker semantic gravity in that our purpose is to present abstract principles that can be used to modify any design project for particular scaffolding purposes, it does not present a specific set of concrete example projects. However, we do strengthen the semantic gravity in the paper when we make reference to actual examples of projects. And we then weaken the semantic gravity again when we return to the abstract principles that we have used to categorize complexity.

Semantic density is the partner of semantic gravity and the two are usually plotted on a Cartesian plane in relation to each other. Where semantic gravity denotes the relative dependence of meaning on context, semantic density denotes relative complexity of meaning.
Semantic density can be thought of as the density (think material density) of meaning. That is, the more elements needed to make sense of an object, a theory, a procedure, or to enact a skill, and the more those elements interact and relate to one another, the more complex the object, theory, procedure or skills. Formally, semantic density has been defined as follows:

Semantic density (SD) refers to the degree of condensation of meaning within socio-cultural practices, whether these comprise symbols, terms, concepts, phrases, expressions, gestures, clothing, etc. Semantic density may be relatively stronger (+) or weaker (−) along a continuum of strengths. The stronger the semantic density (SD+), the more meanings are condensed within practices; the weaker the semantic density (SD−), the less meanings are condensed [38, p11].

LCT (Semantics) has been used very productively to study curriculum and pedagogy in science [39-41], engineering [42] and design [37, 43] among others. In these studies, the notion of a semantic wave has been used to describe teaching and learning paths that shift between simplified concrete examples and more complex abstracted theories. The argument made for cumulative learning is that students need to be able to see knowledge in context and knowledge in relation to other knowledge as a basis to mastery of knowledge that leads to the potential to transfer across domains and across examples [38].

The conceptual model that we present recruits semantic density as a means of describing relative complexity in design projects. As stated, the intended purpose of using a model of this nature is to assist engineering educators to design a sequence of tasks that consciously assists students to progressively acquire the skills needed to use specialized engineering knowledge productively to solve problems that are embedded in the world, problems most commonly encountered as design problems. Our model uses semantic density to categorize the relation between the complexity of the design artifact (and the context in which it operates) and the complexity of the specialized disciplinary knowledge recruited to make sense of the artifact and to predict its potential performance in context. Others have also used semantic density for other purposes.

Shay and Steyn [30] argue that in design, strengthening semantic density means compounding of meaning. As a student learns more about design they start to consider more elements simultaneously. Maton and Doran [44, 45], taking more of a linguistic approach, propose that semantic density strengthens as meanings are related to more meanings resulting in a more complex understanding.

Put another way, semantic density explores the relationality of meanings: the more meanings are related, the stronger the semantic density … The more relations established with other meanings, the stronger the semantic density [44, emphasis added]

For our purposes we used Simon’s [46] rather pragmatic definition of complexity in relation to systems design:

"I shall not undertake a formal definition of "complex systems". Roughly, by a complex system I mean one made up of a large number of parts that interact in a nonsimple way. In such systems the whole is more than the sum of its parts, not in an ultimate, metaphysical sense but in the important pragmatic sense that, given the properties of the parts and the laws of their interaction, it is not a trivial matter to infer the properties of the whole. In the face of complexity an in-principle reductionist may be at the same time a pragmatic holist."[46, p195]

In all the above applications complexity relates not only to the number of concepts/components in an assembly, but also to the relations between them. In terms of semantic density, we would describe an artifact with few interacting parts requiring a rudimentary understanding of what the artifact is and how it works as having weaker
semantic density (SD-), while a system, with multiple subsystems and interacting components all of which need to be understood in relation to each other, would have higher semantic density (SD+). Similarly, we would describe standard equations, used procedurally to size a component as SD-, because the concepts that underpin the equations link sequentially in a defined relationship. By contrast, when an analytical model needs to be developed from first principles it requires knowledge of a network of disciplinary principles linked in conceptually coherent ways, this we would code SD+. Complexity relates to the relations between parts or concepts that need to be understood in order to ‘design’.

Categorizing complexity in design projects

In this section we present our conceptual model for categorizing the complexity of design projects. The purpose of such a model is to provide principles by which engineering educators can plan design projects that allow students to progressively build the skills and knowledge needed to complete complex capstone design projects. We provide four example projects to illustrate the principles. The examples do not represent a trajectory of design projects. They are selected to illustrate the principles that we are discussing.

The model was developed as part of a PhD study and is based on the analysis of a number of different ‘design’ projects. For more detail on the development of the model and detailed descriptions and analysis of the data behind the proposed model see [15]. The publication of the development of the system of categorization is forthcoming [47]. Here we limit our discussion to the presentation of the model, with four example projects to illustrate aspects of the categorization.

Table 1 describes the coding we used to categorize the projects and the principles to which the code refers. The first column indicates the code used to categorize the complexity of each datum. The second column indicates the principle that separated stronger SD (SD++/SD+) from weaker SD (SD-/SD--). The third column describes the key features that we used to identify the code for each project. The discussion below the table will illustrate an example from the data for each code listed in table 2.

Table 1: Coding principles

<table>
<thead>
<tr>
<th>SD Code</th>
<th>Principle</th>
<th>Coding features evident in the data</th>
</tr>
</thead>
<tbody>
<tr>
<td>SD ++</td>
<td>Integration</td>
<td>Multiple interdependent constituents (components/concepts) are integrated into a coherent whole. The causal interdependencies are embedded and there is minimal attention to identification of significant aspects or exclusion of superfluous aspects.</td>
</tr>
<tr>
<td>(very strong SD)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SD +</td>
<td>Integration</td>
<td>Multiple interdependent constituents (components/concepts) are identified in the design brief, but they retain their necessary simultaneous interdependencies. Components /concepts need to be considered simultaneously.</td>
</tr>
<tr>
<td>(strong SD)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SD -</td>
<td>Separation</td>
<td>Multiple constituents (components /concepts) are identified and separated in the design brief, and the interdependencies can be treated sequentially.</td>
</tr>
<tr>
<td>(weak SD)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SD - -</td>
<td>Separation</td>
<td>A component/concept is dislocated from its relations to other parts, losing the significance of the interdependencies within the system from which it was extracted, effectively severing the real causal mechanisms or isolating concepts from their related system of disciplinary concepts.</td>
</tr>
<tr>
<td>(very weak SD)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

What follows are four examples from the projects analyzed in the study in order to illustrate our coding principles. The first example is the first project in the mechanical engineering
design trajectory (M1) in which students focused on selecting and mounting a bearing pair (M1: Bearing mounting). This project illustrates the principle of dislocation, coded SD--. The second example was the second project in the structural design trajectory (S2), where students were presented a conceptual layout of a steel shed and were required to size each of the key structural elements (S2: Steel Structure). This project illustrates the principle of sequential relations, coded SD-. The third example was the eighth and final project in the mechanical engineering design trajectory (M8). Students were taken through a sequence of systems thinking tools in the development of a user requirement specification for a simplified coal fired power plant (M8: Power plant specification). The reasoning required students to consider multiple elements simultaneously, but each element was identified for the students. It illustrates the principle of simultaneous interdependencies, coded SD+. The final example was the final project in the civil engineering design trajectory (C5), in which students were referred to a real precinct and were required to develop a plan to upgrade the precinct including the detailed technical design of a number of infrastructural elements (C5: Future Foreshore). Students were left to identify significant elements themselves, and set up the relations between elements. The problem represents an emergent problem, coded SD++.

**SD--: Bearing mounting (M1)**
The first example serves to illustrate the principle of dislocation, coded SD--. Students were required to select two bearings and design the mounting for them. The artifact was prescribed in the brief as shown in figure 1. The assembly was dislocated from any machine in order for students to focus on only the bearings. The power transmission elements are absent, simplifying the geometry and interaction between parts. The loading is defined as fixed forces, shaft speed and power, eliminating the need to interpret real, variable and uncertain operating parameters. The purpose of the task is only to focus on bearing sizing and then on mounting the bearings for operation and assembly. This project shows how a complex artifact can be stripped down and idealized so that students can focus on discrete aspects of a ‘design’ The bearing selection process is a procedural process. The mounting design is not trivial, but the simplification of the assembly does function to dislocate the problem, allowing students to focus on a single task without being overwhelmed by a very complex artifact.

![Figure 1 M1: Bearing mounting](image)

We coded the problem SD-- because of the dislocation of the task in order for students to focus on single aspects of a power transmission assembly.

**SD--: A component/concept is dislocated from its relations** to other parts, losing the significance of the interdependencies within the system from which it was extracted, effectively severing the real causal mechanisms or isolating concepts from their related system of disciplinary concepts.

Although this shows the process of simplification through simplifying the artifact by dislocating it from its complex interactions, there are other ways to simplify a design project through the brief very differently but using the same principle of dislocation. For example, a
single stage of the design process might be dislocated from the rest of the process by focusing only on ideation or only on identifying performance requirements. In fact, it is usually for the introductory stages in typically engineering science to use this principle to introduce discrete, but dislocated concepts. However, we argue that in order not to leave a task dislocated it is important to make the dislocation and reason for the dislocation apparent to students.

SD-: Steel structure (S2)
One of the four structural engineering projects provides an example of a project that is simplified by prescribing a linear sequence of steps. Students were required to design various elements of a steel structure. Each element was designed under guidance and was intended to integrate a conceptual understanding of the theoretical knowledge and the codes of practice used to design typical structural elements. The conceptual sketch provided in the design brief (shown in figure 2) helps to hold the full design together, allowing students to get a sense of the interdependencies between parts, but by prescribing the sequence of design the interdependencies can be treated sequentially rather than simultaneously (this is not inevitable). The way in which the design brief was constructed means that students can gain an adequate understanding of the structure and the knowledge needed to size each element by considering each element sequentially, not necessarily appreciating the interdependencies between parts and design decisions. They may not choose to engage with the interactions between design decisions.

Figure 2 S2: Steel structure

SD-: Multiple constituents (components /concepts) are identified and separated in the design brief, and the interdependencies can be treated sequentially.

This example the principle of managing the learning through identifying relevant elements (concepts, components or skills) and defining the sequence in which they should be linked. It is useful for beginning to introduce interdependencies between elements, but in an ordered way. This example shows how identifying and sequencing parts of the artifact in the design brief can be used to simplify the design task, reducing the need to consider multiple interdependencies simultaneously, but retaining at least a sequential interdependence. A typical example of simplification through the same principle is to prescribe a linear sequential design process for students. While this risks students losing the reality of iterative design, it is useful early in a design trajectory to introduce structure to design without overwhelming students new to design with the complexity of it. Again, in order to emphasize that sequential reasoning does not define design it is important to be clear about the pedagogic objectives of the project, not leaving implicit any assumption that this sort of project defines design. This
same principle is evident in teaching engineering sciences where students learn to link concepts, initially through linear sequences of equations. Again, while a useful introductory technique, it does risk students seeing engineering sciences as linear procedural operations.

SD+ : Capstone Mechanical Engineering project (M8: Power Plant Specification)
A senior mechanical design project in the mechanical engineering program required students to produce a ‘User Requirement Specification (URS)’ for a simplified coal fired power plant. The project required a group of students to do a high level approximate analysis of the system to determine various technical performance characteristics required for each of the main components in the power plant. Each of the students was then allocated a subsystem in the power plant and was required to do a more detailed analysis on their allocated subsystem. Students were ‘walked through’ a sequence of eight tasks involving different systems thinking tools, such as functional analyses, risk analyses, interface control etc. The required technical analyses were accompanied by extensive notes, articulating the development of each technical analysis from the actual system.

The brief was presented as follows:

The Republic of Rainbows plans an extensive expansion of its electricity supply network. It has decided to construct a number of single unit coal fired power plants at locations close to where the power is required. The generation capacity of each unit depends on its location, but all of them will be supplied by the same coal mine and will have similar site specifics. … All power plants must be based on a simple Rankine water-steam cycle, using coal as fuel. The maximum pressure and temperature must be comparable to typical subcritical steam plants. Other equipment such as coal mills, boilers, steam parts etc. must be of similar technology as employed elsewhere in the republic.

Students were provided with a process diagram of a simplified single unit power plant based on the Rankine cycle, with each subsystem and its relations to other subsystems shown, and clearly showing the path of the three material (fuel, air, and water/steam) streams. The context of operation was defined in thermodynamic terms, with no account given to how they were established, or any concern for seasonal or daily variation:

The following information is valid for the proposed sites where the power plant units will be constructed.

- Ambient air dry-bulb temperature $T_{db}=25^\circ C$
- Ambient average relative humidity $RH=10\%$
- Atmospheric pressure $P_{atm}=101.3kPa$

See the next page for Psychrometric information

The saturation enthalpy vs temperature of moist air can be approximated by the following relationship:

$$h_s(T)=C_3T^3+C_2T^2+C_1T+C_0 \ldots$$

The notes appended to the design brief provided comprehensive details for the technical analyses, showing how various mathematical models were derived from theoretical assumptions in relation to the actual artifact.

Although there is no question that the design of a power plant is extremely complex (far beyond the expectation of 4th year students we would suggest), and the analytical models needed to predict the performance of the artifact involve advanced and complex science, the lecturer managed to reduce the complexity of the task by identifying all relevant components and concepts and their relations to each other for the students. But the reasoning required students to make sense of the multiple interdependencies simultaneously. For example, the
process diagram provided identified material streams and subsystems, and showed their relation to one another, but in order to make sense of the power plant students needed to make sense of all the components and materials in relation to each other as a complex coherent whole. By producing notes defining the theoretical principles of analysis specialized for each sub-system the lecturer did the same with the theoretical knowledge. However, by presenting a completed ‘equation’ that students could use for the analysis, the theoretical knowledge was effectively dislocated from its relations to other theoretical concepts (SD--). Note that this is not an inevitable consequence, some students will choose to engage deeply with the notes and build the network of relations between concepts and disciplines themselves (SD+), while others will choose to treat the equations as discrete objects (SD--).

SD+: Multiple interdependent constituents (components /concepts) are identified in the design brief, but they retain their necessary simultaneous interdependencies. Components /concepts need to be considered simultaneously.

As with the previous example (S2: Steel Structure) students are taken through a sequence of mini design tasks, which function to reduce the complexity of the task substantially. What differs in this example is that the artifact is far more complex. Where the sequence of tasks introduced in S2 focused on individual components, the systems thinking tools introduced in this example helped to retain simultaneous interdependencies between components and between specialized 'knowledges'. In S2 the knowledge was constrained to structures, in M8 the project was more multidisciplinary, adding to the necessary consideration of simultaneous interdependencies.

SD++: Capstone Civil Engineering project (C5: Future Foreshore)
The senior design project in the civil engineering program required students to propose upgrades to the Cape Town Foreshore precinct. The project was undertaken in conjunction with the City of Cape Town. Students were required to develop a high-level precinct plan in a group and then each group member was required to develop the technical design, including technical drawings and report, of one of the infrastructural elements in the team proposed precinct. The brief was presented as follows:

What should we do with the Foreshore?

The north Foreshore precinct is a derelict part of the city characterized by neglected and unused open spaces and remnants of an older freeway-building era. Yet there is great potential to make use of this precinct to create a vibrant, mixed use area, open and attractive to all Capetonians, demonstrating principles of integration and sustainability, and possibly re-establishing the historical link between central Cape Town and the sea. Major new planned projects in this area and in relation to the Port make this task an urgent one.

In conjunction with site visits, photographs of the precinct and a number of presentations by various city officials, students were encouraged to explore the precinct and gather relevant background information. It is clear that making sense of the task and developing both a precinct plan and technical solution requires the consideration of multiple interdependent constituent parts (integrated). But what the students need to consider about the precinct is left for them to identify. There are a great deal of social, cultural, historic and economic considerations to take into account. There are also disciplinary engineering aspects to consider – the hydrology and rainfall pattern, the terrain, the geotechnical conditions (the precinct is located on land reclaimed from the sea), the transportation patterns and existing infrastructure, among other physical and disciplinary considerations. How students make sense of the problem and develop a solution becomes a dialectical relation between the
material world and the theory they choose to describe it. But these elements (material and theoretical) are embedded in the specific problem and are only surfaced by students if they recognize them as significant. This requires both mastery of technical disciplines as well as an appreciation of the impact of engineering in context.

SD++: Multiple *interdependent* constituents (components/concepts) are *integrated* into a *coherent* whole. The causal interdependencies are *embedded* and there is minimal attention to identification of significant aspects or exclusion of superfluous aspects from the brief (or associated materials provided).

What distinguishes this project from all the others is that all the elements required for the design had to be identified by the students themselves and related to each other in meaningful ways, without guidance from the design brief. The reasoning is emergent from the problem.

Discussion

The key point that we are making in this paper is that, just as in engineering science courses where concepts are progressively introduced, and the relations between concepts become more complex, building design competence can be viewed in a similar way. Only, in design the elements are not restricted to concepts in predetermined conceptually coherent relations to each other. In design students need multiple skills, exposure to multiple artifacts in addition to the theoretical concepts they may draw on to model any design proposals. The challenge we offer in this paper is, why do we present students with complex, contextually embedded problems expecting them to identify and relate multiple elements without building up to it? If this challenge has merit, then we have to consider what needs to progress, and how might be develop a progression.

In this paper we have presented a potential logic of progression based on building networks of meaning based on a principle of strengthening semantic density – increasing complexity. We have presented a number of principles that can be used to manage the complexity of any project. Designing any sequence of learning based on a logic of increasing complexity, requires us to find a means of defining the complexity in a design project. It is not as simple as saying how complex is the engineering science that will be drawn on, nor how ‘big’ is the artifact. Rather, the design brief can be used to direct what students need to focus on. Dislocation of a task in order to focus on particular aspects of design can be a very effective way to formulate simpler projects at the beginning of a design trajectory (SD--: M1). Nor is a very complex task (C5: Future Foreshore) necessarily at an appropriate level of complexity. Unless students have already learned to identify relevant aspects of the problem (both the important aspects of the context and artifact – what they are and how they work; and the appropriate disciplinary specializations), underprepared students are left to flounder, and the task becomes a measure of students’ personal exposure to complexity and the process of simplification.

We have presented examples of how design briefs might be formulated in order to simplify design tasks to a level of complexity appropriate to the position of the task in a trajectory of design tasks. When students have little or no experience of design it is useful to strategically dislocate aspects of design from the bigger system (SD--). This can be done by extracting a smaller part of the artifact to focus on particular aspects of design visualization, decision making and/or basic sizing procedures. It could be done by dislocating a part of the design process in order to focus only on ideation or only on identifying appropriate aspects of the
problem and associated knowledge requirements as part of defining the problem. It represents what most engineering science courses focus on, dislocating the theoretical knowledge from the complexity of the world, and from other disciplines, for example stress analysis from thermodynamics, each used in the analysis of highly idealized objects. These are all critical to learning the skills needed to design but are not enough for students to deal with complex, embedded design projects.

Typical capstone design projects are very complex, SD++, where students are required to make sense of real contexts, understand and formulate their own design goals, identify useful disciplinary knowledge as appropriate and translate the physical environments into symbolic form in order to conduct appropriate technical analyses, all embedded in the complex social context of teams and multiple skills-based outcomes. Learning to identify relevant and discard irrelevant aspects from a real design scenario, learning to identify what disciplinary knowledge is necessary and what analyses are not, and learning to translate physical reality into fixed symbolic form is an important aspect of learning engineering judgement, a central skill needed to design. It is this shift from projects where what is relevant (in terms of making sense of the object, and of using appropriate theoretical knowledge) has been identified or cued in the design brief (SD+) to projects where students have to do this simplification themselves (SD++) that is too often overlooked when students land unprepared in a capstone project.

What we offer in this study is a way to think about progression. Our conceptual model helps to provide a language to describe complexity and consequently a guide to constructing design briefs at an appropriate level of complexity, and with a view to building from dislocated introductory concepts/skills/objects to embedded capstone designs in a progressive manner. Although we focused on knowledge in our study, we suggest that our model is adequately abstract so as to be applied also to the development of generic skills.

In table 3, we offer some techniques that were evident in our data that can be used to manage complexity. We have clustered them into each of the principles used to categorize complexity discussed above. While we have not included the data itself, these can be viewed as generalized principles consistent with each category. They are neither prescriptive nor comprehensive but should be viewed in the light of conceptually generalizable principles.

Table 3: Principles for managing complexity in design

<table>
<thead>
<tr>
<th>Codes</th>
<th>Techniques for managing complexity in design</th>
</tr>
</thead>
<tbody>
<tr>
<td>SD--</td>
<td>Simplify the artifact and design only one element</td>
</tr>
<tr>
<td>SD-</td>
<td>Prescribe a sequence of tasks within the design to direct students to relevant physical attributes and theoretical procedures.</td>
</tr>
<tr>
<td>SD+</td>
<td>Prescribe particular design thinking or systems thinking tools to help direct students attention to particular components of the artifact while retaining the interdependencies between components.</td>
</tr>
<tr>
<td>SD++</td>
<td>Provide open ended design projects requiring integration of making sense of a real artifact in an embedded context that requires the use of some theoretical modelling to integrate the physical and theoretical.</td>
</tr>
</tbody>
</table>
Conclusions

What we have presented in this paper is a conceptual tool that can be used to categorize the complexity of design projects. The purpose of the model is to highlight a view on scaffolding learning into the complex capstone design projects that confront students at the end of their academic program. The details of the research project from which the model derives, a rigorous dialectical analysis between empirical data and established theoretical models of sociology of knowledge is not presented here, but can be found elsewhere [15, 47]. Instead we have focused on how the results of the study can inform curriculum design.

One of the fundamental assumptions that our proposal is founded on is the hierarchical construction of knowledge through a progression of increasing complexity. A trajectory of increasing complexity (the progressive strengthening of semantic density) means the construction of networks of meaning, the addition of concepts and more importantly, making sense of the relations between concepts. The fundamental educational principle we base our work on is that cumulatively knowledge building (and by extension the skills to use knowledge effectively for some purpose) is a process of adding ‘things’ (concepts, skills, professional procedures, etc.) into a coherent network. Designing a curriculum to build knowledge and skills cumulatively suggests finding ways to simplify early tasks, allowing students to deal only with a smaller basket of concepts (components or skills, etc.) until they have mastered them before adding new ones which need to be constructed into this network of professional knowledge.

We then argue that any artifact can be simple or complex depending on how many components need to be considered in order to understand what the artifact is and how it works sufficiently to complete the design task. As the number of interacting components that need to be considered increases the more theoretical concepts are needed to adequately model the potential performance of an artifact proposal. The art of curriculum design is to find ways to manage the complexity of the task. Based on a systematic model of complexity, we propose that curriculum designers can identify various techniques that can be used to simplify tasks as is appropriate to a trajectory of cumulative knowledge building. We have attempted to suggest some of these potential techniques in Table 3. Of course, this does not predetermine the way in which all students will engage with the task. It is still important for design coaches to guide students in the class through various interventions. But it does help to set up the intentions of a task.

Acknowledgements

The authors would like to acknowledge the NRF Thuthuka PhD Track Grant for financial support for the study and the UCT URC Conference Travel Grant for making attendance at this conference possible.

References


