Hold on to your holons: A new theory of learning and expertise

Adam Lupu
6.8.2012

Introduction

In this paper, I explore the topic of web page development in an effort to characterize the nature of knowledge and expertise within the domain. I interviewed subjects with varying amounts of experience, asking them to perform two tasks designed to make their knowledge (and how they applied it) observable. While cunducting my research, I had hoped to employ concepts from the existing literature on knowledge and expertise. Specifically, I hypothesized that concepts such as *schema*, *scripts*, and *productions systems* might be particularly useful in describing and interpreting the knowledge applications I observed. However, when it came to analyzing my interview data, I found these concepts could not fully describe what I was observing. I therefore used them to formulate and define a new unit of analysis, which I call *holons*. Once I applied the concept of holons to my data, I was able to articulate several useful descriptions of how expertise may be learned and demonstrated. I will begin with a description of web page development and the specific knowledge I was looking for in my interviews. I will then use concepts in the existing literature to define and distinguish holons. I will follow this with a description of my interview proceedures. Finally, I will apply holons as a unit of analysis to interpret my interview data and describe what we might infer about the nature of expertise and learning.

Web Page Development

To build a web page, you need to learn how to prepare the content you want to display so that a web browser (e.g. Chrome, Safari, Firefox, or Internet Explorer) can interpret how you want that content to be displayed (i.e. "rendered"). To prepare content you must use two types of files. One file type contains the specifications and labels for all the content that you want to display on the web page. The other file type denotes how the web browser should visually display each piece of content. The content container file is written using a computer code called HTML, while the style specification file is written using computer code known as CSS. In order for the browser to correctly interpret what you are intending for it to display, both files must be properly connected to each other, and the way each file refers to content must be specified correctly. This leaves web page developers with four main concerns. First, they must use proper *syntax* (symbols in sequence that together create meaning) when labeling content (HTML) and when specifying which content should be styled in which way (CSS). Second, they must adhere to certain labeling conventions and code writing strategies (e.g. what code goes where?) referred to as *semantics*. Third, they must ensure that the result of their "coding" is a web page that, when rendered, has a functional *structure* that makes sense to the intended audience. Fourth, they must consider questions of how best to achieve their desired *style*. These four considerations, syntax, semantics, structure, and style make up the bulk of a web developer's professional practice.

While investigating the domain of web page development, I looked for evidence of knowledge in each of these four sub-domains and attempted to observe how each interview subject applied that knowledge to their professional practice. By professional practice, I mean the standardized, commercialized skill of building web pages for clients. There are several other forms of web page creation that are intended for personal use (e.g. Tumbler, Facebook, etc.) that allow users to fill in a specified template with certain types of content. These are not what I'm referring to above. Instead, I mean to study the creation of unique, professional web pages, the knowledge used in the process, and the domain expertise performers of this task demonstrate.

Holons

"Schemata truly are the building blocks of cognition. They are the fundamental elements upon which all information processing depends." (Rumelhart, 1980, p. 33) I looked for evidence of schemata (or schema for short) in my interview data. Unfortunately, what I found did not seem to fit this concept of knowledge as the constructions of fundamental units. This notion of units for constructing knowledge, referred to by some as *primitives*, seemed to undermine what I was noticing. Instead of fundamental building blocks of cognition, I was detecting the application of knowledge in chunks that were simultaneously information-specific to the task at hand *and* demonstrations of rich understandings of general domain knowledge. I needed a different way to describe what I was seeing, and this lead me to explore the notion of *holons*.

The term holon was first used in *The Ghost in the Machine* (Koestler, 1967, p. 48). As Koestler defines it, a holon is something that is simultaneously a whole and a part. As I use it here, I intend it to represent a unit of knowledge applied to a part of a task which contains information about the whole task. An example of a knowledge holon for baseball would be the "count." In order to understand the count, one needs to know the difference between balls and strikes, the order in which they appear in the count, the meaning behind each, and the value at which each number of balls (4) and strikes (3) settles an at-bat. While many people have an understanding of what the game of baseball is, they may not have a knowledge holon that contains the meaning behind the count. However, an individual who understands the count is very likely to understand its context within the at-bat and within the game. He or she has their baseball knowledge bundled with their knowledge of the count, and if given a count could then extrapolate its meaning based on their knowledge of the larger topic of baseball.

While *knowledge holons* can explain schema-like concepts and the interpretation of symbols, I will use *step-wise holons* to describe the steps applied proceedurally like scripts and productions systems. For example, each step in a sequence could be considered a holon in that it contains the knowledge needed to carry out that individual step and, through its position and purpose within the sequence, knowledge about the entire proceedure. An example of a step-wise holon in getting a job might be creating a resume. While creating a resume requires its own set of applied knowledge (writing, organizing information, etc.) it also requires knowledge of its purpose and position in the process of getting a job, especially if the job seeker wants to succeed. While writing a resume, one must apply their knowledge of the position they seek, the steps they have taken to prepare for the job, and the steps they intend to take at their new job.

Reducing a whole to a smaller, more usable part without losing the information of the whole is not a new concept in intelligent thought. Similar concepts include holonomy in nueroscience (Pribram, 1991) and holographic reduced representation in artificial intelligence (Plate, 2003). However, as far as I can tell, no one has used holons (or anything similar) to describe knowledge, expertise, or learning. I will do so now.

Procedure

I am primarily interested in capturing the knowledge representations and applications that lead to demonstrable expertise in the field of web development. Given this focus, I felt it necessary to design an interview protocol and data collection strategy that might capture the subtleties of how web developers with various levels of experience and skill perform common professional tasks. Therefore, one of the tasks I chose is the common professional practice of creating a web site from a designer's mockup. I called this the *building task*. I also wanted to design a task that might challenge experts and novices alike to grapple with what they know and how they apply it. To accomplish this, I felt I needed a novel task, one that my subjects were unlikely to have attempted before, but one that was also relevant to their professional work. I chose to artificially remove the browser as a tool for viewing code and asked them to simulate the browser themselves. This task accomplished both of my goals. It was not something any of my subjects had attempted in its entirety, but each of them had experience attempting to mentally simulate how a few lines of code might look in a browser. I called this the *interpretation task*, and the results were rather intriguing. For both tasks, I wanted to capture as much of the subject's performance processes as I could. Therefore I asked them to perform both tasks while "thinking aloud." I will now describe each task in greater detail and specify the data collection strategies I used.

Building Task

In an effort to identify clearly what professional web developers know and how they access and apply that knowledge in their professional practice, I chose one task that mimics a professional front-end developer's daily responsibilities quite closely: converting a digital image of a website (referred to as a mockup) into the code a browser would need to render that image and make it interactive. For this task, I video recorded both the subject and the computer screen they used to conduct the task. Before they began, I asked them to describe their overall strategy (2-3 minutes) and then had them make as much progress on the task as they could in a reasonable amount of time (approximately 10 minutes). I also took notes on their process (including the steps they took in sequence) and asked questions when it wasn't clear what action they were taking or why they were taking it. Occasionally, I inquired as to their reasoning behind a specific process as well. The goal for this task was to see how they applied their knowledge to a standard professional practice. Two specific web development topics were critical to this part of the interview that I was careful to ask about if they did not emerge organically. These topics were *semantics* and the *box-model*. I will explain each of these topics in greater detail during my analysis of the data.

Interpretation Task

I began each interview with the interpretation task. For this task, I wanted to see how they made sense of code without being able to see it rendered in a browser. To help tease out this information, I asked them to look at the code of an existing website, then draw and describe what they thought the site would look like if they used a browser to render the code. Basically, I asked subjects to “be the browser” by simulating its output. I gave them time to work this task until they appeared satisfied (anywhere from 10-20 minutes depending on the subject). Once they had analyzed the code to their satisfaction, I had them open it in a browser, compare it to their drawn image and whatever mental image they had constructed, and describe anything they found to be inaccurate or unexpected. A specific topic that emerged from this task was what happens to the rendered code when the browser’s window is resized (referred to as a code’s “responsiveness”). I asked them to describe what might happen to the image on the screen if they were to resize the browser window. I then had them perform the resizing to confirm, disconfirm, or explain their expectations. I wanted to identify what they might know about code responsiveness and whether this was one of the features they were capable of simulating.

As with the building task, I video recorded the screen and the subject. For this task I also recorded the sketches the subjects made using a *Livescribe* recording pen to capture their pen strokes as they explained what they were writing. This allowed me to analyze both the video footage and the sketch as it was being drawn, as opposed to just the finished product. Having previously attempted to analyze subject-created artifacts and being unsure of the order in which they were constructed, I wanted access to more detailed data of what my subjects drew, when they drew it, and what they said while they drew it. Unfortunately, the technology was not without its limitations and only some of this data was clear from the recordings. Nonetheless, it offered enough data to recreate the order in which each specific element on the page was drawn.

Between the two tasks I collected 30-40 minutes of screen recordings, physical gestures, utterances, and verbalizations for each subject. Additionally, I had data on the approximate sequence in which I wrote my field notes (building task) and in which the subjects constructed their graphical representations (interpretation task). I will now describe the subjects in greater detail and briefly summarize their interviews. I will then present my analysis of this data and describe how it inspired my conception of holon theory.

The Subjects

I interviewed five subjects. Subject 1 was an expert web developer with over ten years of professional experience. Subjects 2 and 3 were practitioners with 1-2 years of experience. Subjects 3 and 4 were students who had been studying web page development for 5 weeks.

Master Mark

As expected, the expert I interviewed showed the deepest and most readily accessable knowledge. He was able to perform the interpretation task almost as well as a browser would, missing only two out of six colors and accurately approximating the code's layout and responsiveness. During the build task he not only progessed the farthest, he also demonstrated superior knowledge of professional practices and a perceptual ability to notice minute details that I had to literally see (i.e. he had to zoom in and show me) to believe. I gave him the name Master Mark to signify his masterful performance on both tasks and his expertise in the domain.

Rigorous Rick

The more experienced of the two practitioners was orginally a designer. He had begun learning how to code the designs he had created and had been doing so sporatically for 2 years. He was methodical and deliberate. He wanted to write all his HTML in such a way that it contained only content and would look like a "nice report" if rendered using the browser's defaults and no additional styling. He performed the interpretation task quite well, correctly approximating everything except three of the six colors and the location of one of the elements. He wrote all the HTML for the build task, and he demonstrated the ability to use CSS to style specific elements. For his attention to detail and his methodical approach I will refer to him as Rigorous Rick.

Pressumptuous Paul

The other practitioner made a lot of assumptions. Some of them demonstrated knowledge and expertise, others created discrepencies in his code that hindered his performance on the tasks. He was able to correctly approximate most of the features of the interpretation task but he made an assumption about the nature of the navigation bar that lead to an innacurate graphical representation. During the build task, he exhibited robust troubleshooting knowledge while trying to solve a problem. However, the problem would not have occured had he tested his presumptions earlier in the excercise. For his "leap before looking" approach, I will refer to him as Presumptuous Paul.

Competent Claire

The more accomplished of the two students took a long time to get started on the interpretation task. However, once she did, she was able to do as well, if not better, than Paul had (including correctly predicting the code's responsiveness). She also described a functional approach to the building task, and she was able to define semantics and apply the box model. However, we did not have enough time to have her write any code, so there was no way to ascertain what challenges she may have faced during that process. For her self described "fuzzy" but successfully applied knowledge, I will refer to her as Competent Claire.

Doubting Thomas

Despite his repeated claims of not knowing what he was doing and being "behind" in the class, the other student I interviewed was able to demonstrate functionally applied knowledge on several occasions. He didn't use the domain specific vocabulary to refer to elements and concepts, but he was able to interpret them using his own words. His performance on the interpretation task was the weakest of the participants, but he managed to correctly approximate the overall structure of the web page. His ability to complete the building task went untested do to time constraints. He accurately constructed a vague meaning of semantics and the box-model despite not remembering ever having heard either term applied to web development. For the abilities he didn't give himself credit for, I will refer to him as Doubting Thomas.

Analysis

I will now explore the data I collected and how I believe it can best be interpreted. I will begin each proceeding section by expanding the definition of holons, the units of analysis I am using to describe my observations. I will then draw from multiple interviews to illustrate the particular expertise I am using holons to describe. I will compare holons to existing cognitive units like schema, scripts, and production systems and I will demonstrate why I believe that holons are more useful for analyzing the application of knowledge and the nature of expertise. I will conclude with a summary of how I think holon theory could be used to describe learning.

Holons and the interpretation of information

Holons are like schema in that they are knowledge units that contain information about how the knowledge is to be used (Rumelhart, 1980). However, they are unlike schema in that they are self contained, they are the reduction and reshaping of all the information that took part in their coalescence. They are the resulting formation of learning as opposed to the foundation of its construction and as such, they are not fundamental units. They are increasingly compact and information filled units. Where Rumelhart suggested schemas are being constantly evaluated against new information (Rumelhart 1980), holons are being constantly formed and reformed by new information.

Therefore expertise is the application of a single, more compact, more refined, more focused knowledge unit (holon) as opposed to the proper grouping of knowledge units and sub-units (schemas). I will now discuss three examples of knowledge holons in action that I observed in expert and novice developers alike. The relative differences across expertise should help to illuminate our working definition of knowledge holons as well as explain how they are applied to specific tasks.

Structure

In web development, structure refers to the positioning of elements on a web page. These elements consist of specific types defined within the HTML lexicon (e.g. body, nav, section, header, etc.) as well as user-defined elements (e.g. divs). Structure also includes what happens to the elements on the page when someone resizes the browser window. In general, a best practice for creating the structure of a web page is to give each element its own box (rectanglular region) on the page and ensure that the widths and hights of all the boxes add up to the total width and hight of the web page itself. This is referred to as the *box-model*.

On both the building and the interpretation tasks, interview subjects applied their knowledge of web page structure. The depth of each subject’s knowledge was clearly observable across their levels of experience, and the greater the subject’s expertise, the more precisely he or she applied this knowledge. For example, in the interpretation task, Doubting Thomas missed properties of each box (e.g. borders, margins, padding), but while he had never heard of the *box-model* by name, he said that he tried to view "most of the elements on the page as boxes." His approach was not as refined as Competent Claire's. She could define the box model when prompted and could think of two ways to use it to devide the mockup in the building task, but she said she wouldn't know how to choose between the two. Her depth of understanding was surpassed by the expert, Master Mark, who was able to articulate two ways of using the box model to divide the mockup *and* choose the more appropriate one based on what he called his "sense of structure."

Incorporated into Master Mark's application of his knowledge of the box-model, is his knowledge of other aspects of structure (including positioning, resizing, and professional conventions). This entire "bundle" of knowledge is what he applies to his decisions about how to divide up and position the elements of a web page. Whereas Competent Claire had less resolved knowledge of structure to draw on when applying the box-model, and was therefore unable to construct a rationale for deciding between two separate applications. While this variation may be explainable using schema, I find it more effective to discuss this difference in expertise using *knowledge holons*. Holons allow us to talk about the nature of knowledge as being broad in scope but condensed (without much loss) in application. For instance, Claire may have a less refined understanding of general structural principles, but she is able to apply them to divide a web page in multiple ways. Her knowledge holon pertaining to the box model possesses enough information about structure to be useful. But Mark's knowledge holon has enough resolution (clarity) about the nature of web page structure that he can use it to judge potential options and pick the one he believes is best.

Symbolic Representations for Color

There are several methods for specifying the color of an element when creating a web page. Each of these methods requires the use of a specific set of symbols that get interpreted by the browser and displayed as colors. Each method has its own set of symbols and a specific syntax that must be used in order for the browser to correctly interpret the symbols. One of the methods is to simply use a predefined name (e.g. red, blue, grey, etc.). While this is certainly the easiest method to apply (due to the fact that the browser has a built in color interpretation for certain words), it doesn't allow web developers to specify what kind of red or how light a shade of grey. Therefore, web developers commonly rely on hexadecimal representations for specifying the color of an element on a web page.

When using hexadecimal representations, a base-16 alphanumeric (0-9, A-F) refers to the amount of red, green, or blue to be added by the browser. For example, 3AC could roughly be transcribed into 3 units of red, 10 units of green, and 12 units of blue. Since the primary colors (red, green, blue) are additive, 000 would mean no color added (black), and fff would mean the maximum value of all three colors added (white). To get even more specific, web developers can use two hexadecimal values for each of the primary colors (i.e. 3A B7 5F) for a total of 256^3 or over 16 million colors.

In the interpretation task, every color was listed as a hexidecimal representation using three characters (not six). Again, the depth of knowledge was proportional to the level of expertise. Both students, Thomas and Claire, were able to articulate that different groupings of symbols represented different colors. Claire was able to identify white, while Paul was able to identify white and grey. Rick was able to identify white, black, and grey and he also said he didn't "totally get the science" behind color representations but that he knew that the hexadecimal values somewhat corresponded to RGB values and that he knew those values were measured on a scale of 0-255. Master Mark not only recognized white and grey, he was also able to correctly identify three different shades of grey based only on the hexadecimal values. He also expanded a three digit hexadecimal code into a six digit code in attempting to guess its color, and during the building task, Mark created a color using both hexadecimals and another representational method (RGBa).

This data indicates a clear progression of expertise, with each participant applying their own bundle of knowledge to the interpretation of a single symbolic representation. The less experienced subjects are not misinterpreting these symbols, they are interpreting them using the knowledge they currently possess. As expertise increases across the subject pool, so to does the robustness of knowledge each subject applies to interpreting the given symbols. Their ability to make sense of the symbols in context is not dependent on a fundamental understanding of hexadecimal representations. After all, even Master Mark missed two hexadecimal color representations and didn't indicate a viable method for figuring them out. This is not an example of understanding from the "bottom up." Instead, the subject's ability to interpret the symbols is a product of the amount and depth of context specific knowledge each of them have compressed into a readily available interpretation mechanism, a mechanism I would therefore classify as a knowledge holon.

Semantics

There are several conventions in web development, some are strictly obeyed, others have emerged over time and are subject to changes in technology and professional practices. Since the collection of conventions is both continuously reforming and open to debate, one's ability to write semantically legible code is dependent on their experience in the domain and the sources of semantic knowledge they have drawn from. Regardless of these sources, web developers must consider several key questions when making semantic choices. These include: What browser-defined element should be used to prepare content for interpretation? What names should be assigned to developer-created elements? Where should a particular instruction to the browser be placed? (e.g. in the HTML file? In the CSS file?) How should the code be written and indented so that other developers can read and make sense of it? When assignging style properties (i.e. color, typeface, etc.) to multiple elements, how should this be done to reduce the amount of code that needs to be written? How should code be writen so that it's optimized for modern search engines?

With all these semantic considerations inherent to the professional practice of web development, experts will obviously demonstrate more expertise than novices. For instance some of the questions above may not even be considerations in the novice's mind. Indeed this was evident in my interview data. There was a clear division between the students and the practitioners, and an even sharper division between the practitioners and the expert. In both cases the division was evident in which questions were being considered and in the subject's specificity of the answers to those questions. However, it was the moment at which each subject applied their semantic considerations that set this data apart as a representative example of knowledge holons.

Doubting Thomas didn't know what *semantics* meant and often struggled to refer to elements by name. However, he did show one instance of remarkable depth of semantic knowledge by questioning the use of IDs instead of classes in the code I used for the interpretation task. He correctly pointed out that semantics (he called them "best practices") favored the use of the reusable categorization known as "classes" as opposed to the unique and non-reusable "IDs." Even without knowing the domain specific label for the concept of naming conventions, he was able to identify where one of those conventions had been broken. He was not "evaluating" his schema against new information as Rumelhart suggests. He was challenging the new information based on the knowledge holon that crystalized when he saw the improperly named elements. This was the only evidence of Thomas articulating some form of best practices, which could indicate that his knowledge of semantics is sharply refined for this specific consideration, but may be less clearly resolved or even completely void of other semantic knowledge.

Competent Claire also stuggled with appropriately naming several elements. However, she was able to generate a working definition of semantics ("giving things meaning outside their [specification] block"). While her verbal definition was quite inclusive, she could only think of one consideration for which semantics were used. Search engine optimization (SEO) was tied directly to her knowledge of web development semantics. This meant that any application of semantics depended on that single consideration. Clearly Claire's knowledge was also compactly applied, but failed to similarly contain the robustness of multiple semantic considerations. Whereas Rigorous Rick's considerations included SEO naming conventions, and a strong dedication to the proper placement of instructions. Master Mark demonstrated all of the above considerations and several more. For example, during the building task, while creating the logo, Mark indicated that it was "common practice" to combine a company's logo graphic with the text of the company's name and tagline indented out of view. This is done to make the company name "visible" to search engines. But it is also done so that blind website visitors who use text to speech software will get the name read to them. Bundling these considerations together demonstrates a greater volume of knowledge being simultaneously applied through the creation of a single instruction. This is how an expert applies knowledge holons.

Holons and strategies

While knowledge holons map nicely to concepts, symbolic interpretations, and bundles of applied knowledge, they do not characterize the entirety of cognition. We must also consider proceedural accounts of intelligent thought. Schank and Abelson referred to one such process as "a standard event sequence" and used the term *script*. Anderson and Koedinger explored *production systems* as proceedures for accessing and performing operations on declarative knowledge based on given conditions. These units of analysis seem to share a quality which distinguishes them from Rumelhart's definition of schema and, in my estimation, serves to group them together. If we consider scripts, production systems, and other proceedures to be their own set of knowledge representations, we might call them *strategies* and define them as the application of knowledge in a particular sequence of steps to achieve a particular pattern or resolution.

If a strategy contains multiple steps, then each step must contain information about the entire strategy. For instance, knowledge of the step's position in the sequence, its purpose for being a part of the strategy, and how its implementation affects other steps within the strategy would all be characteristics of expertise and may indeed be necessary for the successful application of the step. The greater the expertise, the more refined and robustly integrated the step within the strategy. In this case, a holon would be an individual step within the overall proceedure. It contains information about the strategy and the knowledge necessary to be applied successfully. I will now explore three strategies used by my interview subjects and describe how the individual steps they used required the application of their existing step-wise holons.

Setup

When creating a web page from a wireframe or mockup, a developer must decide how to begin. There are certain requirements to consider including the HTML and CSS files must be correctly linked together, the folder structure used to house those files must be properly created and named, and the document type must be specified so that older browsers can understand newer web pages. There are also decisions to be made including whether or not to use a *template* that offers prewritten code, a *framework* that offers prewritten code and specifications for writing future code, and/or a *reset* which changes how the browser will interpret coded styles by default.

Interview subjects attended to each of the requirements and decisions mentioned above while seting up or discussing the setup of the building task. For instance, Doubting Thomas said that he would use the Twitter Bootstrap framework because it was what he was "most comfortable with." He demonstrated knowledge of browser defaults during the interpretation task by remarking that when the CSS stylesheet did not target the paragraph tags in the HTML element, it would lead to "regular old paragraphs." When I asked him what he meant by this he described (without using specific terminology) the concept of browser defaults and resets. However, he expressed that he wasn't clear what defaults and resets would do so he preferred to stick with what he knew (i.e. Twitter Bootstrap).

Meanwhile Presumptuous Paul liked to take shortcuts. He said he would usually start with a template, but since he didn't have one readily available, he copied and pasted the code for specifying the HTML document and linking files from the interpretation task to begin the building task. This was a rather ingenious time-saver and he had clearly used this step in other setup proceedures before. However, it did lead to a problem later on when the code linking files in the interpretation task didn't match the folder structure he had created for the building task. The code in the interpretation task conformed to semantics in the way its folders were setup, whereas he had not set his folder up in the same way.

Rick skipped the document specification (saying that he knew it "pretty well" and could add it in later), and went straight to building the folder structure and linking the files, which he did so correctly. Then he built all the HTML before moving on to the CSS. He claimed he knew HTML very well and CSS not as well so he wanted to start with the "stuff" he understood the best.

For each of these subjects, their choices of steps involved in setting up the web page reflected the knowledge they were most comfortable with. Thomas wanted to use Twitter Bootstrap, Paul wanted to copy and paste, Rick wanted to stick with writing the HTML himself before moving on. While the steps they chose reflected a bias for comfort with knowledge application, each step required an understanding of where it could be placed in the sequence of steps and what other steps it allowed the developer to skip or alter. The choices each subject made fulfilled his or her need to create a setup strategy and carry out that stategy based on what he or she knew. The steps themselves reflected the knowledge holons each subject felt they possessed.

Master Mark demonstrated his superior expertise by stating that he had "hand coded" a combination of templates, resets and frameworks (which he called a "frameset"), but since it wasn't available during the task, he just created his own folder structure, worked through the document specification, then used a powerful targeting mechanism (which he called a "wildcard" selector) to create a CSS reset in a few seconds. He had a step by step process that he could access, and within each step, he possessed the depth of knowledge needed to recreate a partial version of the pieces he had already built for his professional practice. This was an example of his application of robust knowledge in discrete chunks used to achieve individual steps that fit together in a successful setup strategy. I call these *step-wise holons* and suggest that they describe the nature of strategic expertise.

Workflow

In addition to required proceedural strategies like setting up documents and files, professional web developers have their own personal strategies for building web pages. These strategies manifest in the specific sequence of actions they take to perform sub-tasks relevant to the overall task. For example, Rick's desire to code all of the HTML then move on to the CSS or Mark's desire to begin at the top of the web page and build downward element by element, are examples of personal strategies. When it came to these chosen *workflows* one of the sharpest contrasts emerged between the students with five weeks of experience and the other subjects (who all had 1 year or more). Basically, the students appeared to flounder without any discernable workflow strategy. While the practitioners and the expert each began applying their strategies immediately to both tasks.

When Doubting Thomas was attempting the interpretation task, he said he wasn't sure "where to start" and decided to "just pick" a specific div (labeled "container") because it "seemed to make sense" to start there. When he switched to the CSS to see what kind of styling had been applied to it, he tried to use a *find by name* function even though the name "container" was immediately visible on the screen. He was further hampered in using the *find* function by his inexperience with using a macintosh computer. It became obvious that the way I set up the task prompted the use of several step-wise holons that he did not have as much understanding of as did the other interview subjects. While Claire also scrolled through the code looking for somewhere to begin the interpretation task, Rick, Paul, and Mark were all able to visually scan a list of CSS selectors, and each one of them had a method for choosing which element to work on interpreting next. But even without much practice at applying these step-wise holons in this particular strategy, Thomas still created a strategy to attempt the task I had given him, and his strategy used the steps that he knew. In other words, even though he was far surpassed in efficiency and accuracy by the more expert subjects, he applied his knowledge in the same basic way, by using the step-wise holons he could access to create and follow a strategy. I will further discuss the implications of this uniformity in knowledge application across levels of expertise in the following section on holons and learning.

While Mark, Paul, and Rick all used workflows that involved the strategy of analyzing elements from the top of the web page down, Rick preferred to do this first in HTML and then CSS, while Mark and Paul both preferred to preface this approach by collecting information about styles that might apply to multiple elements on the page. These styles are called "global" styles and would include things like common typefaces and the background and button colors, all of which would be applied across multiple pages or multiple times within a page. Taking the step of setting the global styles before dealing with individual elements implies at least some knowledge about the semantic consideration that prompts developers to write non-repetitive code. By writing the global styles first, Paul and Mark could then focus on writing just the necessary style adjustments on individual elements (referred to as "local" styles). This further demonstrates how a step in the process requires knowledge of the entire strategy and how the other steps are then affected by the incorporation of this new step. Indeed, the entire strategy is reshaped by the accretion of this one step-wise holon at the beginning of the proceedure. Again, I will discuss this relationship between step-wise holon accretion and the reshaping of overall strategy in the next section.

Debugging

During my interviews, I witnessed two debugging proceedures. One was performed by Presumptuous Paul and the other was performed by Master Mark. Debugging is a troubleshooting proceedure that occurs when a web developer encounters an unexpected event while performing an individual step in their strategy. This event halts their progress toward task completion and usually involves the browser failing to interpret their code as they had intended. While this unexpected event is the catalyst for the debugging process, the strategy each developer uses to fix this "bug" in the proceedure depends on their knowledge of the situation and how the browser interprets what they have written.

As I mentioned previously, Paul's unexpected event occured because of his incorrect linking of his HTML and CSS files. He noticed that the browser was not rendering one of the style changes he made, but he didn't know the reason why. He guessed that it might be because he didn't target the individual element correctly when he set up the style selector. He then launched into a debugging proceedure using a strategy that employed both knowledge and step-wise holons. The strategy included refining the selectors he used to alternative (and more specific) ones, searching for and choosing a site on the web that would provide him with the correct syntax to use when making the style adjustment he was attempting, and finally, checking to see if other styles he had previously written were being rendered by the browser. This last step, led him to the solution. Once he discovered that none of his styles were being rendered, he knew to check the code that specified how the files should be linked. He noticed the error and corrected it. He successfully, albeit inefficiently, applied a debugging strategy to overcome the unexpected event and continue the task.

Mark had a very similar event occur while he was performing the building task. He too noticed that the browser was not rendering one of the styles he had just written. However, he had already applied a step-wise holon during his setup strategy to check whether the files had been properly linked. He wrote the linking code, then wrote his first style, then checked to see if the style had taken effect (he even remarked that this was an indication that the style sheet was linked successfully and he was "good to go"). So what might seem like prescient foreknowledge of what could go wrong in the future, had already been incorporated into his setup strategy in the form of a step-wise holon containing this knowledge (probably from several experiences similar to the one Paul had).

Instead of a broken file link, Mark was faced with an even more obscure bug. He had just created a search bar (one of the features of the mockup) and was attempting to give it a particular styling (in this case a *border-radius*, also known as rounded corners). The styling adjustment he made wasn't having any effect on what the browser displayed. So he began his own debugging strategy. His strategy included holons that demonstrated greater expertise and more extensive knowledge than Paul's. First, he thought it might be a browser-specific issue so he applied prefixes to the code he had written to specify that the particular browser he was using (Chrome) should read that code in a certain way. This didn't work, so he opened the browser's code inspector (which allowed him to see what the browser was interpreting and how it was interpreting it), he located the styling effect and "cranked it up." There was no change. Then he tested the same code in a different browser and noticed that his styling *did* take effect. The corners were just as round as he intended them to be. This prompted the following (excerpted from my field notes):

At this point Mark guessed that Chrome has its own way of interpreting a "search" form with its own uneditable default styling that was overriding the styling he was trying to apply. He checked this assumption by changing the labeling of the input from “search” to “text” and then rechecking the styling in the Chrome browser. This solved the issue and now his styling was taking effect. He then recalled having seen this before with Chrome and identified it as a significant conflict. He wanted the “semantic value” of having a search input type, but Chrome has its own styling and therefore the display would not look exactly like the designer had intended. He said he would take this dilemma to the client and would say, “Look it's going to look a little bit differently in Chrome” and urge them to let him define the input as “search” in order to "take the semantic win over the stylistic difference." He went on to say, "if this is a fortune 500 company, sometimes that's just not an option." In which case he would have to change the label to "text."

The depth of knowledge Mark displays in the above passage difinitively characterizes him as an expert in his field. He bundles knowledge of semantics, style, and client preferrences into a single choice of what label to give a specific element on a web page. However, his process is basically the same as Paul’s. He applys what he knows in discrete bundles encapsulating related knowledge and effectively stepping through a problem solving strategy. He uses the holons he has to perform the task he is given.

Two key difference emerge in Mark’s expertise which I believe offer insight into how holons can be used to both describe and improve learning outcomes. The first difference is in the level of specificity and clarity observable in Mark’s holons. The second difference is in his ability to extrapolate (or expand on) his knowledge and clearly articulate each holon’s relationship to a larger body of knowledge. In other words, he can reason from part to whole. I will now describe these differences in greater detail and use them to promote a more thorough investigations into holons and learning.

Holons and Learning

When examining expertise in web page development, it has been helpful to characterize knowledge as existing and being applied in the form holons. These bundles of domain specific knowledge that enable experts to complete discrete tasks must be aquired somehow. Surely, experience helps. Specifically though, what about experience facilitates the creation of holons? I will first describe what I believe to be three mechanisms of holon creation. I will then illustrate each using examples from my interview data. Finally, I will conclude with suggestions for how these mechanisms may be applied to create learning opportunities that help novices become experts.

Accretion

Let me begin with how holons form. I wish to introduce a term Rumelhart used to describe the creation of what he called factual knowledge, namely *accretion*. I do not intend to use his definition, but instead offer one that is more analogous to the concept of accretion as defined in planetary science. According to theories of planetary evolution, the Earth formed from the dust, gas, and rock that orbited a simultaneously forming Sun. This proccess of material coming together to form a more discrete whole is referred to as accretion. I believe this is similar to how holons form, through the accretion of groupings or patterns of information, that are combined and compacted into a functional, task-specific form. The feeling of holon accretion could be described as that sense we get when our understanding is “taking shape” or becoming more “defined.”

Returning to the analogy, the Earth does not neccesarily take the same form as the material from which it accreted, but it does contain representative elements of all of that original material. Holons also contain representations of the patterns of information that solidified to form them. And just as the elements that compacted to form the Earth reformed and changed shape, so too do our perceptions of the groupings and patterns of information that accrete into our more refined holons. Gravity was the single, dominant force responsible for the formation of the Earth. I believe there is an equally dominant force responsible for the formation of holons. Because I do not yet have a more refined or specific term for this force, I will simply refer to it as the *learning force*. Without exageration, we can safely say that the learning force is observable over the course of every human lifetime as they aquire new skills and apply knowledge to perform tasks. It was also observable in Presumptuous Paul's debugging strategy. After applying several step-wise holons to determine what was not working and how to fix it, he finally arrived at a solution that prompted him to recall a lesson he was still in the process of learning. He summarized this lesson when he concluded his debugging session by exclaiming, "That's why you don't copy other people's code!" Paul could have diagnosed the problem, fixed it, and moved on. But he felt some compulsion to relate his debugging experience to previous information. It is this compulsion, this force of learning, that I believe drives the next stage of holon evolution: continuous formation.

Continuous Formation

Rick and Mark had two distinct workflows when it came to performing the building task. Rick wrote all the HTML first, then moved on to the CSS. Mark wrote each element in HTML, then he wrote the styling in CSS. Both of them exhibited superior expertise and both workflows were successful applications of knowledge to the given task. Each strategy emerged independently through the accretion of unique step-wise holons. As these strategies are applied and reapplied to similar tasks, Rick and Mark likely adjusted and readjusted each step in their strategy's sequence. The possibiliy that holons undergo this iterative style of continuous formation would account for the emergence of distinct but equally potent performance strategies. If holons are non-transferable (i.e. they cannot be given or exchanged), then each individual performer must accrete their own task-specific holons. Applying this framework to learning implies that it would be more advantageous to focus on developing one's own problem solving or task performance strategies then to rely on mimicking other people's strategies. In other words creating one's own semantics, or personal conventions, on top of existing professional (cultural) semantics would be a form of learning optimization.

An example of this quality of continous formation occurred during Doubting Thomas's interpretation task. He noticed that the code in the task used *ID*s as labels, but he thought (correctly) that *Classes* should have been used instead. He said he was "confused" and asked for clarification. He made an effort to clarify his understanding against new information. This could be described as a form of evaluating one's schema, but I believe we get more traction in a conversation about learning, if we interpret this as a form of focusing on contradictions as a way of further clarifying what one knows. In other words, I suggest describing learning not as the adition of knowledge to existing constructs but as the refinement of our knowledge forms, or the continuous formation of our holons.

Extrapolation

As I mentioned in the previous section, Master Mark repeatedly demonstrated an ability to reason from part to whole. He was able to verbally extrapolate how the discrete task he was performing at that moment was related to the overall task. He was also able to articulate other information that was relevant but not necessary to the task at hand. If we consider holons as the application of knowledge to a given task, Mark's holons certainly contained the most information, applied with the greatest clarity, when compared to all other subjects.

I assert that Mark's ability to extrapolate is not just a manifestation of his expertise, but also contributes to his continued development as an expert in his field. Mark's ability to re-present his knowledge in multiple forms and at multiple levels of abstraction and specificity was not unique to him. Every subject exhibited some attempt to re-present (i.e. extrapolate) from part to whole. These attempts occurred most often when the subject encountered a difficulty or some insufficiency in their knowledge. In each of these cases, the subject eventually articulated a novel form of information grouping that they had not previously stated or used. For example, when I asked Doubting Thomas what web page development semantics were, he claimed to have no idea (despite having drawn on several semantic considerations to perform his interpretation task). Instead, he attempted to understand semantics by analogizing them to his wife's use of English. He said that his wife was a non-native speaker and while her meaning was usually clear, the way she said something was often "a little off." After drawing on this analogy he said, "so maybe semantics is something like giving your code the right meaning in the right way… like best practices or something?" To me, his statement suggests that some form of accretion and/or continuous formation was occuring, prompted by his extrapolation from part (web semantics) to whole (conventions of meaning and syntax in spoken language). This interpretation could explain why reasoning by analogy can be a successful strategy.

Conclusion

I have promoted a new conception of knowledge and expertise. I claim that knowledge is applied in discrete bundles containing both general domain and task-specific information. I call these bundles holons and assert that expertise emerges from their accretion, continuous formation, and extrapolation. From this perspective, learning is not the assembly of knowledge, but the clarification of one's interpreations of patterns of information. To draw on language commonly used when expressing what one knows and how well one knows it: through "clarification," what seems "fuzzy" becomes "crystal clear." Rather than expertise being the result of knowledge built "brick by brick" it is the result of condensing "clouds of information." Therefore, I offer holon theory as a prototype theory of expterise and suggest that the central function of holons is in the application of knowledge to the performance of a task. Through this lens, I gained valuable insight into the knowledge and expertise of web page developers. I invite further investigation and research into how learning can be described and promoted using holon theory.