Good afternoon. I am a PhD student at the University of British Columbia working with Tamara Munzner.

And over the next several minutes, I'm going to describe a *Different* way to think about visualization tasks.

Hopefully I'll convince you that our proposed *Typology of Abstract Tasks* is a useful tool.

I'll provide some examples of how you would use the typology

and I'll also comment on the similarities and differences between our work and the two papers that were just presented in this session.

The contributions of our paper include:

- a framework for analyzing *Tasks* that addresses this *Gap* between *Low* and *High* levels of abstraction.
  And this is in the form of a conceptual *Typology* of abstract tasks, rather than a *Taxonomy* of empirically observable interactions.

- And this framework integrates and extends work from several domains, including previous classifications of visualization usage as well as a number of theoretical frameworks.

- Finally, I'll present a way to describe *Sequences* of tasks, one that addresses this confusion between *Means* and *Ends*, particularly for tasks involving *Derived Data*.

My motivation for doing this project was somewhat different than my co-author's.

- In my case, I was deep into a longitudinal field study of visualization adoption and I had hit this *wall*: I was struggling to describe and compare tasks between users, visualization tools, and domains over time.
  I've also since begun work on a design study in a particular application domain.
  For both projects I really needed a practical and systematic tool for analyzing tasks abstractly, allowing me to both generate and evaluate designs that address these tasks.
  While I won't talk about *Generating* or *Evaluating* design in this talk, I will talk about a new way to *Describe* and *Analyze* tasks.

- Meanwhile, Tamara has been thinking about tasks for quite a long time and she is in the midst of writing about tasks for her forthcoming textbook, and has been in need of an *Analysis Framework* for *Describing Tasks*.

  Her motivation included this more *Blue Sky* theoretical question: *What does the visualization community know about tasks anyway*?

Of course, we started by digging up all the previous work we could find in the visualization, visual analytics, and HCI literature that proposed some classification of tasks, goals, interactions, and so on.

- We found lots of work that classified visualization usage at this *Low* level of abstraction, which allows for an easy mapping to particular visual encoding and interaction techniques.

- There's also a number of *High-Level* classification systems, which map to many domain problems but it is hard to connect these directly to any of the *Low-Level* classifications.

- At last year's VisWeek BELIV workshop, our group described this gap between *Low* and *High* levels of abstraction and a need for *Mid-Level* classification of tasks.

- We're not the only ones to notice this gap, the two papers before me in this session address it as well.
  However, even the small number of papers that do address this middle level don't explicitly connect low to high, like connecting *Retrieve Value* to *Integration of Insight* for example, which was our goal in this work.

  Connecting low to high is useful is useful because it allows us to describe and analyze the mapping from particular techniques all the way up to particular domain problems.

Another problem not addressed adequately in previous work is the confusion between *Means* and *Ends*.
Obviously Schulz and colleagues noticed the same thing.
Now consider, for instance the term *Derive*. It's found in a number of previous classification systems.
So is *Derive* a task, or the *Means* by which a task is executed?

- As an *End*, *Derive* might describe reducing the dimensionality of a dataset for its own sake.

- As a *Mean*, consider a user who is seeking to discover clusters in a *Derived* low-dimensional space

You could ask the same *Means-End* question about a lot of terms found in previous classifications.

- For instance, what about *Sort*?
  Are we *Sorting* along the way, or is *Sorting* something our goal?

- In either case, we end up with an *Output* where things are sorted differently from the *Input*.
  So we now consider these *Input* and *Output* transformations more generally, and how they might help to clarify this confusion between *Means* and *Ends*.

So we set out to integrate and extend previous work to address the *Low-High Gap* and this *Means-End Confusion*, we ventured to sort through all the vocabulary used in previous work to classify visualization usage, much like in Roth's card sorting study.

And while we didn't enlist a group of expert visualization users from a particular domain to help us sort through all this vocabulary, several of the previous classification systems we surveyed are in turn based in empirical study, though these tend to be more effective for *Low-level* classification.

- We adopted an hierarchical coding strategy akin to those used by social scientists for qualitative data analysis and theory construction.

- and we iteratively refined our codes and hierarchies as we read additional papers, initially working with sticky notes, whiteboards, and diagrams showing the associations between all these terms.

So as we read and coded, we encountered hundreds of terms and definitions characterizing visualization usage at various levels of abstraction, which we integrated.

- For example, one paper alone by Mullins and Treu had over a hundred and forty terms pertaining to tasks in their taxonomy, which was far too many for our purposes.

- So what we did is we arranged, we abstracted, we merged and split terms, we simplified, to get down to 27 terms.

In our paper we include this full-page lookup table that maps the *27 terms* used in out typology to previous work.
In the end we carefully considered 30 previous classification systems as well as 20 other references that made compelling or noteworthy assumptions about tasks.
The left column contains our 27 terms, and the main column on the right lists synonymous or closely associated terms from those 50 references. For each term, we also indicate references *who also explicitly use* that same term.

- For instance, we use the term *Navigate* in our typology, as did others before us. Some terms that we associate with *Navigate* would include *Panning*, *Zooming, Rotating*, and *Drilling Down*.

- Or consider *Compare*, which we associate to a number of other terms, such as *Discriminate*.

While we had this *Bottom–Up* qualitative coding approach, our thinking was also framed from the *Top-Down*, by existing theory from the cognitive sciences, HCI, communications, and visualization literatures.

- Among those considered was Norman's *Stages of Action* model, which also informed Roth. In addition we also considered Heidi Lam's proposed extension to that model set in the context of visualization usage, in which she described a higher-level *Gulf of Goal Formation*.
  In the paper, we also discuss our typology in reference to *Distributed Cognition*, *Sensemaking*, and *Play Theory*.

- Another obvious influence was Munzner's *Nested Model*, which in itself describes *Multiple **Levels*** and a mapping between data and task abstractions and visualization techniques.

- Our typology spans the *Yellow Abstraction **Level*** and the top half of the *Green Technique **Level***, and the colours used in our typology reflect this correspondence.

- Our typology is organized around these three questions: *Why is a task undertaken? What are the task's Inputs and Outputs*? and *How is the task supported* in terms of *Methods*, which we define as families of related visual encoding and interaction techniques

With this structure, our typology can be used to describe tasks at multiple levels of abstraction, while also clarifying this confusion between the *Means* and *Ends* of tasks.

- We're able to do this because we consider the *Input* and *Output* of tasks explicitly, so we can describe whole *sequences* of interdependent tasks.

So here's the whole typology. I'm not gonna walk each node in detail, as there are definitions and examples for each of these in the paper.
What I *will* do is give an overview of how you use it, and then I'll give some examples.
So to describe a task you need to include elements from all three parts of the typology.

- On the left, we have the *Why* part of the typology,
  And to describe *Why* a task is undertaken, you ultimately decide whether the task is about *Consuming* or *Producing* information.
  If it's about *Consuming* information, we can break that down further into several levels of abstraction.

- First, you choose the type of *Consume*: is it to *Present*, to *Discover*, or to *Enjoy*.

- At the next level of abstraction, you choose the type of *Search*: and our characterization of *Search* depends on whether the *Location* and *Identity* of the search target are known in advance. These include *Lookup*, *Browse*, *Locate*, and *Explore*.

- Finally, you choose the type of *Query,* which corresponds to the number of search targets under consideration: these include *Identify, Compare,* and *Summarize*.

- Including *Produce*, there are altogether 37 possible paths that traverse the *Why* part of the typology, because of the full cross product of *Consume* types, *Search* types, and *Query* types.

On the right, we have the *How* part of the typology.

- It's comprised of 3 groups of methods:
  those that *Encode* data,
  those that *Manipulate* existing visualization elements, and
  those that *Introduce* new elements.

Unlike the *Why* part of the typology, a single task can be described using *a Combination* of methods from these three groups.
I won't list all the individual methods now as I'll hit many of them in the examples to follow.

You'll also notice that all of the terms in *Why* and *How* parts of the typology are *verbs*. This is deliberate, as the *nouns* in a task description are accounted for by the *What* part of the typology.

- For each task description, you have to fill in the *Input* and *Output* nodes with *Nouns* relevant to the context of the task.

- So now I'm going give a couple of examples, but just as a recap: the *Why* part is about *Selecting from Alternatives* at each *Level of Abstraction*, the *What* part is about filling in the *Input* and *Output*, and the *How* part is about *Selecting Combinations* of *Methods*.

To start with a simple example, let's consider the task in which a user wants to *Present* a path between two nodes in a large tree to their colleague, using the *SpaceTree* visualization system.

- Using our typology, we would describe this task as *Presenting* at a *High* level of abstraction, and *Locating* and *Identifying* a path at a *Low* level of abstraction,

- This *Identification* is supported by a combination of methods working in tandem: *Navigation* through a visual *Encoding*, and *Selecting* a path to highlight it; and in *SpaceTree, Selection* implicitly *Filters* and *Aggregates* unselected nodes.

- We can then compare this to the same task as performed with another system for visualizing large trees, in this case *TreeJuxtaposer*, where *Selection* can implicitly *Arrange* nodes.

  In Section 2 of our paper we discuss how these subtle comparisons were hard to describe using previous classification systems, and yet such comparisons are essential when evaluating similar systems.

Now to an example involving a sequence of interdependent tasks, one in which the user wants to *"verify a hypothesis regarding the existence of clusters of items in a scatterplot of dimensionally reduced data, then labelling those clusters of points."*

This involves several steps:

- First, dimensionality reduction from a high-dimensional space to a 2-dimensional space, an *Encoding* of the 2-dimensional data and the *Exploration* of the scatterplot, the *Identification* of clusters

- and their *Annotation* with Coloured labels.

Here's how we would describe this sequence of tasks:

- We begin with a task that's about *Producing* new information rather than *Consuming* existing information; here we're *Producing* 2D data by *Deriving* it from high-dimensional data, in this case using dimensionality reduction.

- Second, we have a *Discover* task involving *Exploring* and *Identifying* clusters by visually *Encoding* that data as a scatter plot, *Navigating* that visualization, and *Selecting* clusters and points.

- Finally, here's another instance of *Produce,* in which the *Output* is the result of *Annotating* these clusters and points with *new* coloured labels.

  Again this sequence of interdependent tasks would have difficult to describe using previous classification systems.

With the time that remains, I'll discuss some similarities and differences between our typology and the tools presented in the last two talks: these being Schulz and colleagues' *Design Space of Visualization Tasks*, as well as Roth's *Taxonomy of Cartographic Interaction Primitives*.

Here the blue highlights ideas that are common to all three systems, while the red highlights some of the notable differences, these being elements not accounted for in those two classification systems, *OR* elements that we organize in a different way.

- One difference, for example, is that neither Schulz and colleagues nor Roth explicitly accommodate the *Casual* use of visualization, this being visualization usage motivated *Not* by a need to *Present* or *Discover* information, whereas we use the term *Enjoy* to describe such tasks.

Another prominent difference from our typology and that of the previous two talks is in our characterization of *What*?
In early drafts, we tried to characterize *What* comprises a visualization in more detail, but the story got far too complicated: and we decided that you really need a whole separate paper to talk about data as opposed to tasks.

- We simplified to an agnostic *Bring-Your-Own-What* mentality. We realized that as long as you specify the *Input* and *Output* of a task, you can use any previous classification of elements comprising a visualization, some of these including those that I've listed here might be specific to a particular domain or data type, such as for graphs or time-oriented data.

- You could just as well use the *Characteristics*, *Target*, or *Operand* classifications from the other two papers presented in this session.

I'll also speak about the use of our typology to describe sequences of tasks.

- Schulz and colleagues do provide a textual notation for describing what they refer to as *workflows*, such as their example from section 3 of their paper in which they use their notation to describe Shneiderman's well-known *Visual Information Seeking Mantra*

- Our corresponding visual notation captures the same three steps

- while also explicitly shows the links between *Inputs* and *Outputs* of subsequent tasks in the sequence.

- In the *Overview First* step, we *Explore* and *Summarize* an Overview of the data, which is supported by the visual Encoding.

- In the *Zoom* and *Filter* step, we *Browse* the *Overview* to *Identify* a *Subset of Items* that interests us, supported by *Navigation* and *Filtering.*

- Finally, in the *Details on Demand* step, we *Browse* that *Subset* and *Identify* a particular item that interests us, *Navigating* and *Selecting* it to learn more.

- Also note that each part of the *Mantra* is about *Consuming* information, and it could apply equally to *Present*, *Discover*, or *Enjoy* contexts.

To summarize what I've talked about, I'll take the perspective that asks, *what is this typology good for?* And I'll make reference to the three criteria proposed by Beaudouin-Lafon in his 2004 AVI paper:

- I've shown that we can **describe** the use of existing systems, allowing you to analyze and compare *sequences* of visualization tasks between tools at multiple levels of abstraction, from *Low* to *High*, clarifying *Means* from *Ends*.

- You can also use the typology when **generating** new designs, mapping domain problems into techniques, and this is particularly useful at the early stages of *Design Studies*. I'm currently working on a project making use of the typology in this capacity.

- Finally, you can use the typology to **evaluate** as I have been doing in an ongoing study of visualization adoption, as a tool for coding domain tasks at multiple levels, allowing for rigorous and systematic qualitative analysis. Another potential application is in providing higher-level external validity for prescribed tasks in controlled lab study experiments.

I'd like to thank our funding source, NSERC, and the many people we discussed these ideas with.

I'll be happy to chat with you about our typology and how I'm currently using it, and I welcome your questions and comments.

Thank you very much.

Applied visualization design is a process of…

…abstracting a domain task and considering appropriate visual encoding and interaction techniques.

So how do we describe an task in an abstract way?
And what's the right level of granularity?

We propose a way to think about abstract tasks that distinguishes why a task is undertaken…

…from how the task is executed…

…while acknowledging what dependencies the task may have

These descriptions allow us to make comparisons across domains and between visualization tools incorporating different techniques.