The Factory Design Pattern:

Main Points of the Paper:

• Creating objects takes significantly more time than using constructors
• People generally use the class constructors as the more natural choice. If they cannot do that then they use the available factory pattern
• Using Class clusters as an alternative pattern which can overcome some of the shortcomings of the factory pattern
• Class clusters provide the same abstraction...is it comparable to the factory pattern?
• A Factory pattern is used for creating instances of related classes

How this is relevant to move the research forward:

• A bit difficult to apply to tabletop research
• When is the Factory pattern useful?
  o The logic of determining the correct type of object can be hidden from the user
• Used to get a set of related objects and abstracting the logic of choosing the right type of object
• Most people don’t use design patterns
• 50% of all developers are below average

Discussion Points:

• There are formal ways to evaluate the usability of an API...Sending a little bit more effort to assess the usability of the API and select a group of different types of programmers. Study who the users are of the system
• If you are an API developer, do not assume that you are the typical user of the API...
• What is missing in tabletop APIs that are available?
Native PDF viewer problem...Working around the problem of using it on the tabletop with different orientations...The iconic application provides a work around.

What about using a browser application on the tabletop.

Backward compatibility issues. It would be much more helpful if the API of the table can support integrating reusable components which are not originally developed for touch sensitive screens...

The operating system capabilities are also an issue. Maybe Windows 7 can solve some of the problems. Does it support multiple concurrent interactions effectively to suit the applications on tabletops?

We cannot afford to create everything from scratch. The missing capabilities are causing a real problem when real life applications are being developed.

3 Main Points:

• How do we transition into the new paradigm of tabletops? Whether we need to rewrite everything from scratch.

• A wish list for tabletop API. What is not supported but is inherently important to be available in the API?

• The API can be evaluated formally to know if it is usable. The evaluations need highly trained professionals to participate which make these studies extremely expensive.