From your reading of chapter 6 and your own knowledge from other sources, discuss the following:
What is the significance of Lorenz and Kidd.s specialization index? What are the implications of a high specialization index? A low one? A major change in the index as the product evolves?

One of the key advantages of Object Oriented design is reusability. Classes are built to model a generalized set of properties and methods, which are then implemented in object instances. Subclasses inherit properties and methods from parent classes, and then extend them with additional properties and methods.

With this in mind, according to author Mark Schroeder, the specialization index metric:

> …measures the extent to which subclasses override (replace) the behavior of their ancestor classes. Numerous overridden methods indicate that the abstraction may not have been appropriate, since (if much behavior needs to be replaced) the child and ancestors may not have much in common. A more appropriate for of subclassing involves extending class behavior with new methods rather than replacing or deleting behavior through overrides (Schroeder, 1999).

This is a less-concrete explanation of SI than our textbook gives, where SI combines the overridden methods plus times the level (which I'm assuming is the level of inheritance), divided by the total number of methods, but the result is the same, the higher the index, the more methods being overridden, suggesting the subclass is not as related to the ancestor as it should be.

If subclasses are regularly overriding their ancestor classes, then this suggests there is a problem with the object model. The objects are not generic enough, or improperly defined. The lower the SI, the better, as the more generic and reusable the parent class.

If the SI increases dramatically during development, this could be the result of pressure on the developers, who may be implementing less-elegant solutions due to time constraints. A dramatic drop in SI could come from a code-review session or a period of time spent refactoring.

References:

Schroeder, Mark (1999). *A Practical Guide to Object-Oriented Metrics.* IT Pro. IEEE. Retrieved from Centro de Informatica on May 4, 2009 at:
http://www.cin.ufpe.br/~inspector/relacionados/metricsbymark.pdf

 Lee,

Great response, I wasn't familiar with the Specialization Ratio, but it is a very important metric. You are correct that we do not want a one for one relationship of parent classes to

subclasses, that would indicate the parent classes were too specialized and not sufficiently reusable. I like the concept of a "wider tree" of object relationships, with a parent class being the trunk, and the subclasses the branches expanding out from it, if the Object design matches this pattern, it demonstrates a high degree of reusability for the parent class.

With this in mind, then applying the Specialization Index to the subclasses to determine how many of the properties and methods they are inheriting from their parent classes would inform us as to whether the parent classes are sufficiently generalized enough or if they were too specialized for the subclasses to inherit from.

Thank you for a challenging reply. I had to do some research to properly reply to it. :)

Miscellaneous object-oriented metrics
http://www.aivosto.com/project/help/pm-oo-misc.html