PRODUCT PROBLEM: For social applications to gain compelling critical mass, users on the supporting social graph must be made aware of other users' interactions with these applications. This suggests deeper integration of the application *into* the social site.

This problem has existed since the dawn of software: the difficulty of getting our data, product, or system out into general use. The lack of users becomes a particularly notable difficulty in the space of Web 2.0 because without users to consume and (especially) *generate* our content, how useful can our system ever become?

Facebook supports a large number of users who are interested in sharing information along social connections, and it can feature content from applications just as well as its own content. Giving external applications a presence on the Facebook site would make applications built by both large and small developers more discoverable, helping them gain the critical mass needed to support good social functionality.

Whatever solution we create, the applications need a distinct display presence on the Facebook site. The Facebook Platform makes this available to our application, reserving the URL path *http://apps.facebook.com/fettermansbooks/...* for that application's content rendered on Facebook. We'll see how the platform integrates the application's data, logic, and display shortly.

The second problem is another outgrowth of our data services built into the "Data: Creating an XML Web Service" and "FQL" sections, and is just as tricky.

PRODUCT PROBLEM: External applications cannot use certain core data elements that Facebook does not expose through its web services.

Facebook makes a great deal of data available to its users when producing the content of its website (*http://facebook.com*), but it chooses not to make every bit of this available through the external data services. Privacy information itself (the can_see mapping from "Some Facebook Core Data") is a good example—not explicitly visible to users on the Facebook site, the can_see mapping remains invisible to the data services as well. Yet enforcing the use of the privacy preferences users maintain on Facebook is the hallmark of a well-integrated application, and one that upholds the expectations of users on the social system. How are developers able to harness this data, which Facebook, to maintain the privacy of its users, has not released through its data services?

The most elegant solution to these problems will incorporate Facebook data with the external application's data, logic, and display, while still operating under a trusted environment for the user.

DATA SOLUTION: Developers create application content for execution and display on the social site itself through a data-driven markup language, interpreted by Facebook.

Applications using only the Facebook Platform elements of the "Data: Creating an XML Web Service" and "FQL" sections create a social experience *external* to Facebook, augmented by the use of Facebook's social data services. With the data and web architecture described in this