In this example, you're creating button fields with a label and an icon. You can use an `Image` object for the icon (1), or a `PdfTemplate` (3) (in this case, you're using an imported page). I've used this functionality in a real-world project to create online examinations. Every question had a button that allowed the student to get a hint. If that button was clicked, an annotation was made visible and a hidden field was set. The value of this hidden field was posted together with the answers, so that the tutor could see for which questions a hint was used.

Some very simple JavaScript is used to hide (or reveal) the fields (or annotations) (2). You get a field instance with the `getField()` JavaScript method for interactive fields, or with `getAnnot()` for ordinary annotations. Then you change the properties of these objects as explained in the JavaScript reference. In this example, clicking the upper button (named `click`) hides both buttons. Clicking the lower button (named `advertisement`) opens the web page dedicated to this book at Manning.com.

~~Pushbuttons aren't always meant to be pushed (or clicked). In the next example, we'll use pushbuttons as "hot areas" that trigger an action when the mouse moves over them.~~

7.4.3 A popup triggered by a button that doesn't need to be pushed

A popup annotation has no appearance stream or associated actions of its own. It's always associated with a parent annotation. Figure 7.14 shows a text annotation as a popup. If you take a close look at the image, you'll also see a widget annotation on top of the *Donnie Darko* poster. If you move the mouse inside the borders of this widget annotation, the popup with the text annotation will appear; if you move the mouse pointer outside the widget annotation, the popup will disappear.

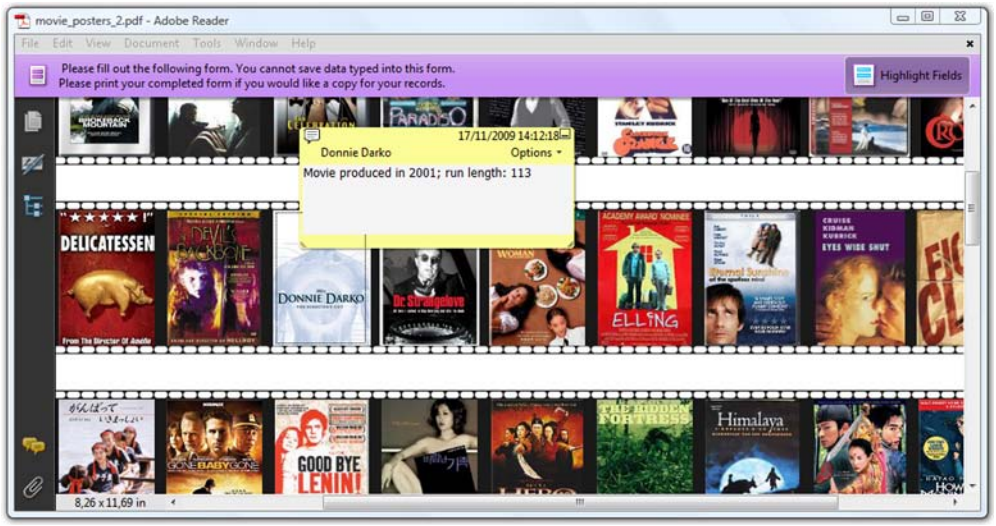

Figure 7.15 **Text annotation in a popup using a button and its events**