



Applied Design Tools & Interfaces

Designing for the Activity - Framework for Application Design

Designing for the Activity

Qualities of a great application

Is there a framework for good design?

“*you bet there is!*”

Focuses on the activity instead of the audience.

The feature set consists of what is absolutely necessary.

It supports the users mental model of what it does.

It helps users get started quickly.

Streamlines workflow so mistakes are kept to a minimum.

Reinforces spatial memory.

Does not present a cluttered UI.

The screenshot shows a web application interface for managing article publishing. It features a top navigation bar with tabs: Publishing, Images, Parameters, and Meta Info. Below the navigation bar is a 'Link to Menu' button. The main content area is divided into two columns. The left column contains a 'Publishing Info' section with fields for State (Published), Published (checked), Access Level (Public), Author Alias, Change Creator (Administrator), Override Created Date (2004-08-19 19:11:07), Start Publishing (2004-08-18 23:00:00), and Finish Publishing (Never). Below this is a 'Content ID' section with fields for Content ID (5), State (Published), Hits (10), Version (1 times), Created (Thursday, 19 August 2004 19:11), Last Modified (Thursday, 19 August 2004 19:14), and Expires (Never). The right column contains an 'Article ID' section with fields for Article ID (1), State (Published), Hits (33), Revised (20 times), Created (Thursday, 12 October 2006 13:00), and Modified (Thursday, 03 May 2007 10:41). Below this is an 'Article Parameters' section with fields for Author (Administrator), Author Alias, Access Level (Public), Created Date (2006-10-12 13:00:00), and Start Publishing (2006-01-02 18:00:00). At the bottom right, there are two expandable sections: 'Advanced Parameters' and 'Metadata Information'. The 'Metadata Information' section is currently expanded, showing 'Left: 1.0' and 'Above: 1.5'.

Applied Design Tools & Interfaces

Designing for the Activity

Basic Framework for Application Design

Define the solution completely...

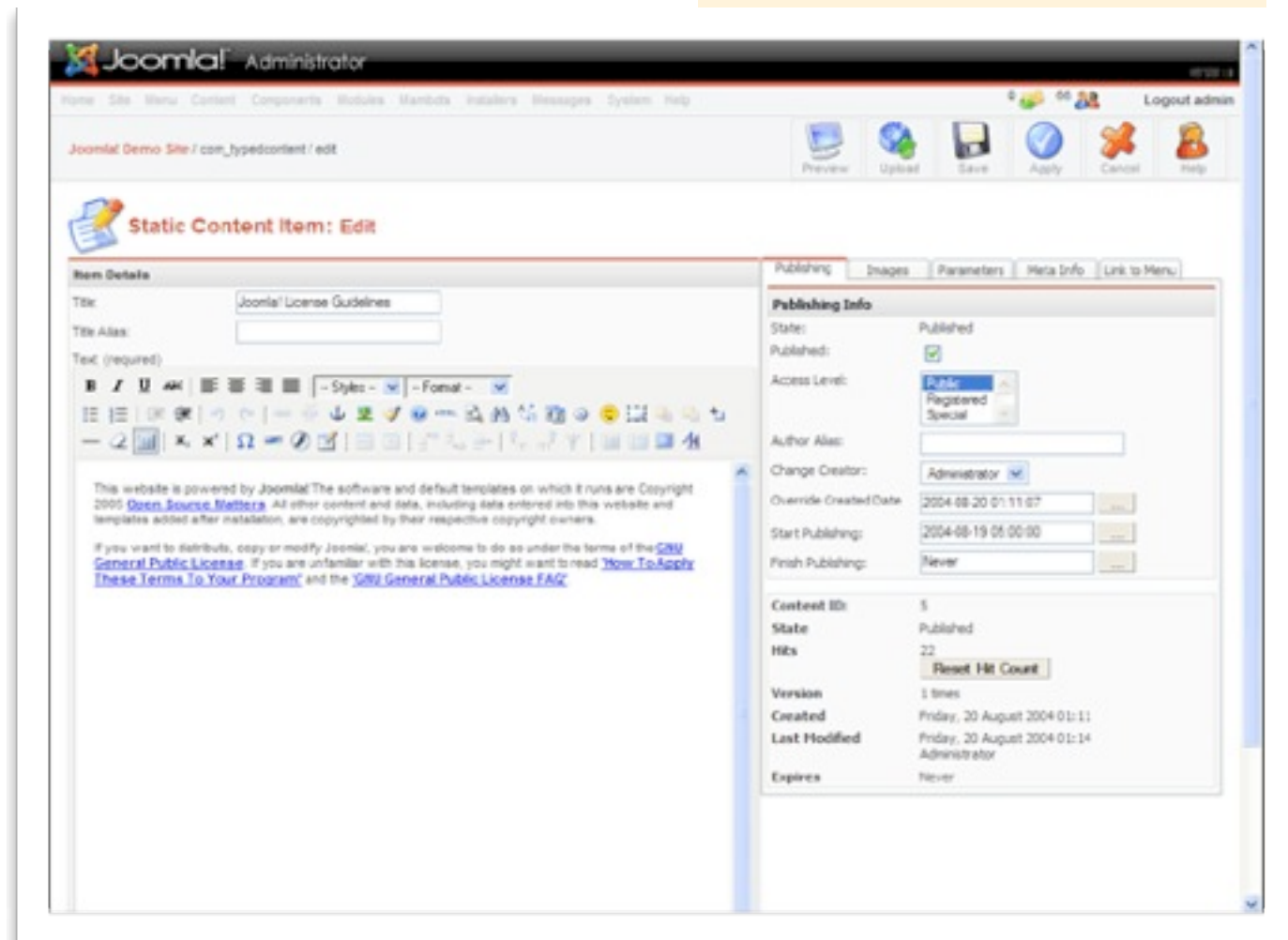
Know what to build

Requires complete understanding of the what application will support.

Conceptual Element: consists of the knowledge of what to build, what not to build and the underlying intent of the application.

Knowing the essential components will allow for clarity of purpose.

Clarity of purpose helps create a sense of desirability around the WBA.



Applied Design Tools & Interfaces

Designing for the Activity

Basic Framework for Application Design

Define the solution completely...

Know what makes it great

Technology alone does not make an application great.

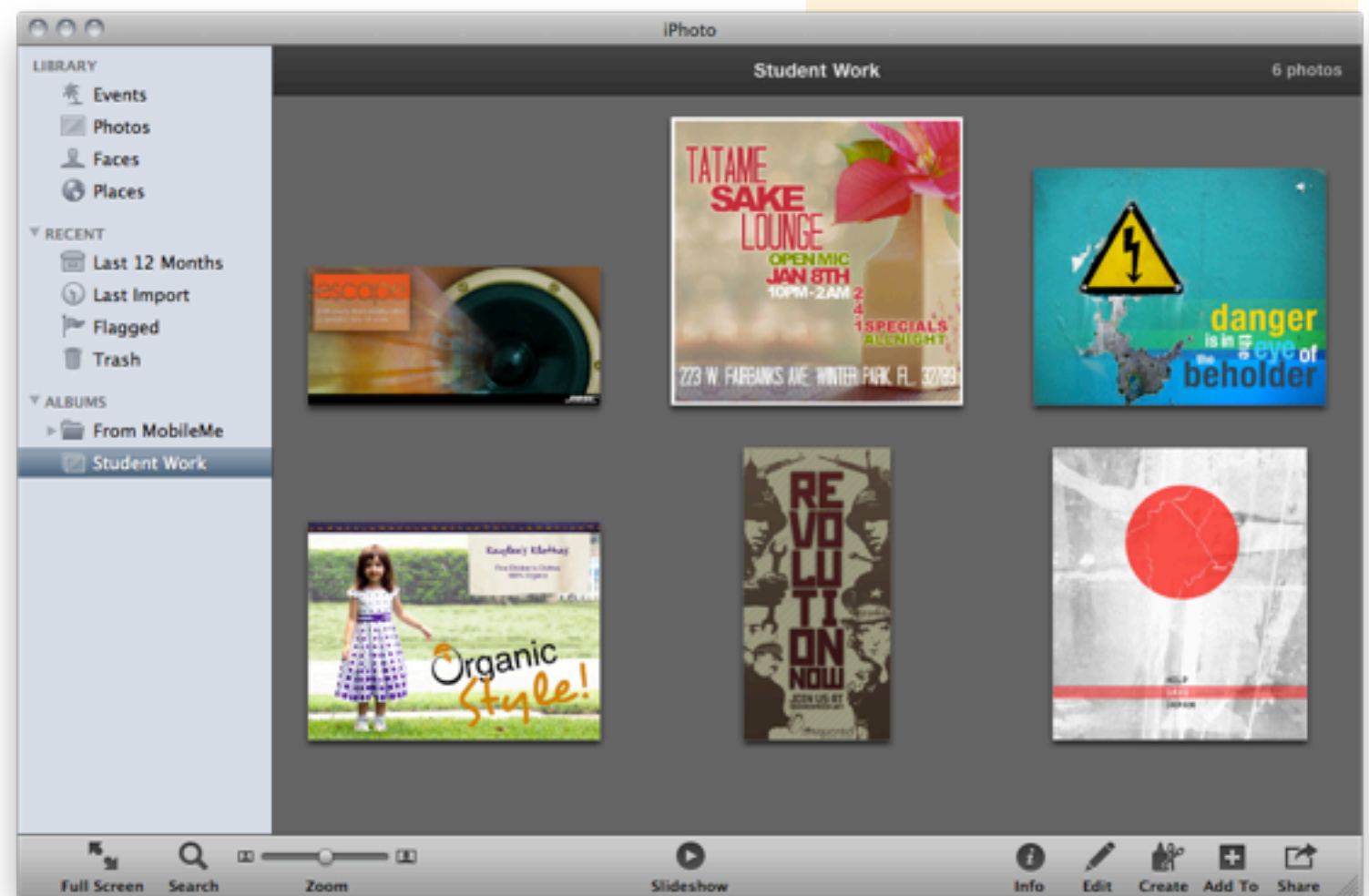
Users do not interact with the code base,
but they do interact with an interface.

Clarity of purpose in the workflow should
always be the main goal.

When a WBA is bad, it tells you in a myriad
of ways.

- ▶ When it's good, you just know it is.

Application Element: the “hook” that creates
a sustainable and positive user
experience.



Designing for the Activity

Basic Framework for Application Design

Define the solution completely...

Know the best way to implement it

Interaction Element: the technology that drives the solution.

Always the biggest variable to deal with.

User Interaction and Usability should
always be the main guiding force.

Avoid forcing technology on a solution.

```
function doBeforePaste(control){
    maxLength = control.attributes["maxLength"].value;
    if (maxLength)
    {
        event.returnValue = false;
    }
}

function doPaste(control){
    maxLength = control.attributes["maxLength"].value;
    value = control.value;
    if (maxLength) {
        event.returnValue = false;
        maxLength = parseInt(maxLength);
        var oTR = control.document.selection.createRange();
        var iInsertLength = maxLength - value.length + oTR.text.length;
        var sData = window.clipboardData.getData("Text").substr(0,iInsertLength);
        oTR.text = sData;
    }
}

function LimitInput(control)
{
    if (control.value.length > control.attributes["maxLength"].value)
    {
        control.value = control.value.substr(0,control.attributes["maxLength"].value);
    }
};
```