Functional Specification  
*The Conversation: 3.0-E-002*

# Table of Contents

1. Overview
2. Scope
3. Risk Assesment
4. Feature List
5. Data Specification
6. Code Components
7. Testing Plan
8. T&R Estimates
9. Open Issues

# Overview

These features allow users to interact in a threaded newsgroup (forum) on a per-film and/or per-screening basis. These newsgroup threads can be created ad-hoc by authenticated users, or can be migrated from Screenings via the chat “promotion” feature described below.

Threads, once started, contain timestamped responses that chain from the initial post, but do not chain from each other. So while a specific “chain response” may address and follow another, the relationship is inferred by time and context, and not enforced. (See NYT Blogs for an example of such a thread structure).

Each thread and following comments will likewise have the author’s “username”, a “reply” button (thread only), and “share on facebook/twitter” links, with appropriate beacons, and an “abuse” button to report offensive material.

Threads may be film-specific, or aggregated sitewide and available on the site homepage.

.

# Scope

This is a narrow enhancement, for all browsers and users but a segregated AJAX service.

# Risk Assesment

This feature set is considered low risk, as all features and data will consist of new services, data tables, and html elements. With appropriate attention this should have minimal effect on current capacity, and few if any dependencies..

# Feature List

The following features should be implemented:

1. Add “Promote” button to chat messages. This button moves chat to a new thread in the “thread view”.
2. Threads are displayed in chronological order, but may be sorted by:
   1. Date of Post
   2. Time in Movie
   3. Author of initial Thread
   4. Activity in response to Thread
3. Thread functionality should be HTML Localized, and should publish to the following locations:
   1. Film Page
   2. Home Page
   3. Theater Page
   4. 3rd Party Pages (via iFrame)
   5. Facebook Page (via iFrame)
4. Threads may also be created in the “thread view”
5. Threads and comments must come from authenticated users
6. A hashtag syntax should be used to pull relevant content from Twitter, and display in the Thread list.
   1. Incoming Tweets are considered new threads
   2. All responses to Tweet Threads are re-tweeted, from Constellation
   3. Twitter button should allow users to tweet the thread, or comments, on demand
7. Any thread or comment can be shared on Facebook
   1. All responses will be published to the Constellation Application on Facebook, specific for the relevant film
   2. Facebook button should allow users to feed the thread, or comments, on demand

# Data Specification

This feature set requires the following modifications to the data specification.

1. Table: conversation
   1. conversation\_id (int pk auto)
   2. fk\_author\_id (int)
   3. conversation\_item (text)
   4. conversation\_response\_count (int)
   5. conversation\_date\_created (datetime)
   6. fk\_film\_id (int)
   7. fk\_screening\_id (int)
   8. conversation\_status (tinyint)
2. Table: conversation\_response
   1. conversation\_response\_id (int pk auto)
   2. fk\_author\_id (int)
   3. conversation\_response (text)
   4. conversation\_response\_sequence (int)
   5. conversation\_response\_date\_created (datetime)
   6. fk\_conversation\_id
   7. fk\_film\_id (int)
   8. fk\_screening\_id (int)
   9. conversation\_response\_status (tinyint)

# Code Components

**Symfony**

*Widgets*

1. Conversation (New)

*Components*

None

*Pages*

None

Helpers

1. Conversation Helper (Mod)

**Python**

*Services*

1. Conversation (New)

**PERL**

*Services*

None

# Testing Plan

TBD:

1. Users should …:
   1. …
   2. …

# Time and Resource Estimates

The above featureset and will require the following time and resources:

1. User Signup posts to User’s wall**Time: 2 Hours  
   Resources: 1 Developer**
2. User Purchase posts to User’s wall provided the $1 opt in isn’t checked  **Time: 1 Hours  
   Resources: 1 Developer**
3. User Chat posts to User’s wall  
   **Time: 2 Hours  
   Resources: 1 Developer**
4. Theater screening contains post-screening messaging

**Time: 2 Hours  
Resources: 1 Developer**

1. Theater screening implements 1 minute playback (TBD)

**Time: TBD  
Resources: 1 Developer**

Development Total Time: 36 Hours (7 Days)

GMR Testing Total Time: 10 Hours (2 Days)

Client Testing Total Time: 8 Hours (1 Day)

Updates and Fixes Total Time: 5 Hours (1 Day)

Deployment Total Time: 2 Hours (1 Day)

**Total Time: 61 Hours (12 Days)**

# Open Issues

The following questions need to be answered prior to final approval of functional specification and development:

1. **One Minute Playback**   
   Does this require an on-demand FFMPEG Media Parse? How do we stream that asset?
2. **Feed vs. Recommendation vs. Comment**Is there any reason to specify these as different items?
3. **Non-Facebook Users**

Is there any way to allow non-Facebook users to BECOME Facebook users in our UI?