

Tuple - Voltaire Project

BitWise Developers

March 18, 2014

1 General

1.1 Purpose of Document

This document provides a description of the proposed solution submitted by Bitwise Developers in response to the invitation for bids on tender for the Voltaire system for Tuple.

1.2 Introduction

Organizations that offer training courses usually provide artifacts such as notes and other training material to students. These documents can take a long time to research, write and format. In addition, the content for the course, how it relates to other parts of the course, and how the course relates to other courses, need to all be figured out. All of this results in a lot of time and work, which could have been spent somewhere more productive. Tuple proposed a solution, a software based system for semantic content management and artifact generation. It should be able to link different types of content with various relationships, generate a curriculum out of this and generate artifacts such as notes for it.

2 Executive Summary

2.1 Stakeholders

2.1.1 Project Owner

Client: Tuple (Pty) Ltd

Contact Person: Craig Edwards

E-mail: craig@tuple.co.za

2.1.2 BiTwise Developers

Bitwise is a team like no other. We do not just take on any project, but find projects that are innovative and worth our time. When we find such a project, we put in the effort to produce results beyond the expected. We have a perfect mix of low level coding specialists and high level abstract thinkers, programmers that make it work well and designers that make it look good, risk takers that try new things and careful planners that ensure we are still on track. We are a diverse team that can tackle almost any problem and come out on top. We are Bitwise.

Lead Developer: Francois Oberholzer

E-mail: u12039803@tuks.co.za

Developer: Priscilla Hammond

E-mail: u11025477@tuks.co.za

Developer: Jacques Lewis

E-mail: u28183488@tuks.co.za

3 Technical Proposal

3.1 Project Description

3.1.1 Problem Statement

A system needs to be developed to manage content semantics and metadata using standards-based semantic modelling techniques and technologies. The relationships between content has to be defined, stored and queried. The system should also be capable of generating artifacts containing the content such as notes or examination papers.

3.1.2 Objectives of the project

- Define and store content objects which contain the metadata of content stored elsewhere.
- Define the relationships between content and represent this in the metadata.
- Generate a curriculum/structure with relating pieces of content, being so far as possible requirements of each other and extract from this a set of requirements for that structure not included in the structure.
- Generate artifacts for the structure.

3.2 Proposed Solution

Bitwise Developers believe that we can solve the problem by using our knowledge and resources, and by applying our skills as software developers. We propose a possible solution as follows:

3.2.1 Solution Overview

Walter Maner said in 1978 at the ACM's annual conference that although curriculum generators promise to be effective instructional delivery systems, making educationally sound use of them awaits the development of computable measures of problem complexity which have psychological reality, which reflect observed item difficulties. This poses a serious dilemma for lesson designers which will tend to be resolved in favour of computable but empirically invalid measures.

It is difficult to understand the complex process of learning and understanding. Some pieces of knowledge are requirements for others, one must understand what a number is before one can add them together, and must be able to add together before differentiating and integrating them. Yet to understand the rate of change does not require this, although that is what differentiating and integrating is all about. People also perceive difficulty differently. More than this, ones preconceived notions of a problem determines how well one is able to grasp it, and ones ability to grasp a problem determines ones notions of that problem. To create a perfect curriculum is hard even for people, who have intuition. To let a computer generate a curriculum makes this part harder, it can't think about the psychological ideas people have and how that influences them in the same way we can. But perhaps this is why a computer is better suited to tackle this problem. It can look at the problem in a logical way, taking many more details into account we cannot think of, without having any subjective human ideas get in the way.

What is needed is a proper system that can represent the relationships between units of content and create a perfect curriculum. We embark on this challenge of creating this system. We will have Content, this can be any type and is stored somewhere. We generate metadata for it. We will define

a set number of relationships and link the content together, this is included in the metadata. The actual content will not be stored, only the metadata pointing to the content and the relationships between the content. From here we can group similar content, and define requirements for them. We can add content and link them in such a way as to minimize requirements, until we only have a small list of requirements for the whole group, everything else being covered by the content in the group.

Once a group or structure or curriculum has been set up, artifacts need to be generated for it. This includes notes and tests. It can identify the main points of the structure and summarize it or test it.

We will start by setting up the structure, modelling the content and relationships. Once this is in place, we will experiment with different algorithms to generate structures. Once we are satisfied with the structures we will develop algorithms to summarize the structures into notes and tests.

4 Deliverables

4.1 Documentation

- Requirements Specification
- Functional design from a technology neutral perspective
- Mapping of functional design unto specific technologies and architectures
- User Manual
- Installation Manual

4.2 Source Code

- All project source code
- All test cases
- Build scripts
- Deployment scripts

5 Communication Channels

Meetings: The client and Bitwise will commit to regular meetings.

E-mail: All BiTwisE members will be reachable through their respective e-mail addresses throughout the duration of the project.

Phone: All BiTwisE members will be contactable by their respective cellphone numbers.

Other: BiTwisE Developers are open to other means of collaboration, for example Google hangouts, or Skype etc.