Architecture Specification

for

Key Word In Context

Version 2.0 Approved

Prepared by Lynn Barnett and Victoria Potvin

University of Central Oklahoma

February 4, 2015

Contents

[1. Architectural Style: Mixed Shared Data and OO 3](#_Toc412666505)

[1.1 Shared Data 3](#_Toc412666506)

[1.2 OOT – Object Oriented Techniques 3](#_Toc412666507)

[1.3 Advantages 3](#_Toc412666508)

[1.4 Disadvantages 3](#_Toc412666509)

[1.5 Conclusions 3](#_Toc412666510)

[2. Diagrams 4](#_Toc412666511)

[2.1 Component Diagram 4](#_Toc412666512)

[2.2 Deployment Diagram 5](#_Toc412666513)

# Architectural Style: Mixed Shared Data and OO

The project will be implemented using a mix of Shared Data and OO Architecture.

## Shared Data

* The components of this architecture are processes and data.
* The connectors of this architecture are direct memory access, subprogram calls, and system I/O.

## OOT – Object Oriented Techniques

Object oriented techniques aim at encapsulating data and defining class relationships well to create easy to maintain software.

## Advantages

* Shared data helps performance in the case of large datasets because the data does not have to be copied by components.
* Shared data allows for easier implementation of interactive functions.
* Information hiding and well defined relationships due to OO mean an easier to maintain piece of software.
* Classes are easier to change without unintentionally affected other classes.

## Disadvantages

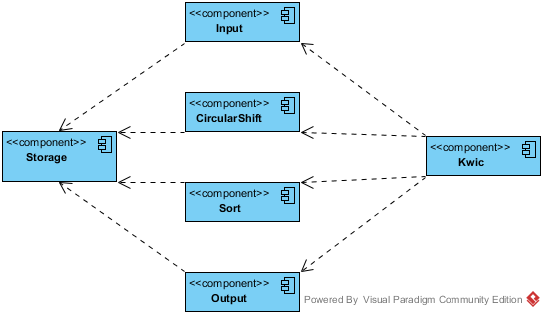
* With Shared Data architecture processing by components must be done sequentially, therefore not in parallel.
* Object oriented techniques can sometimes result in bloated and over-complicated applications.

## Conclusions

Because the system is simple, there is not much need for parallel processing or worry about a bloated system. Shared data is a good choice since we desire to create the software as a web application and in the future it will be more interactive. OOT is a good choice because our software will be easy to maintain in the future.

## Diagrams

## Component Diagram



## Deployment Diagram