COMS3002 - Software Engineering

Wade Downton (1171712)

August 2017

1 ii) Choice of Architecture

1. **Types of architectures and their pros and cons:**

   - Blackboard

   Description:

   A blackboard system is an artificial intelligence approach based on the blackboard architectural model. The architecture relies primarily on machine learning and AI to solve complex, NP-Hard problems.

   Pros:

   - Made to handle complex, ill-defined problems with the use of AI
   - Is able to easily adapt to new data

   Cons:

   - This architecture does not scale well to real world problems
   - Very complex to design and implement
   - Requires a large amount of pre existing data to train algorithms

   - Client-server (3-tier)

   Description:

   A distributed application structure that partitions tasks between the providers of a resource or service. A three-tier client/server is a type of multi-tier computing architecture in which an entire application is distributed across three different computing layers or tiers. It divides the presentation, application logic and data processing layers across client and server devices. The client will request from the application server and the application server will request from the database server. The client will never communicate without the use of the database server.

Pros:

- All files are stored in a central location
- Backups and network security is controlled centrally, as only the application server can access the database server
- Users can access shared data which is centrally controlled
- The workload on the client is lowered with the use of a server

Cons:

- The server are expensive to purchase and maintain
- If any part of the server side network fails then a lot of disruption can occur

- Monolithic application

Description:

A monolithic application is built as a single unit. The application is not separated into smaller partitions and services. The single unit involves the program and its own data storage. No centralization exists in this architecture without the use of a server.

Pros:

- Simple to design and create

Cons:

- Designed without modularity, i.e. Cannot add to or alter the application without difficulty
- Does not scale well for larger applications
- Can be difficult to debug, as problems can be difficult to locate on large scale applications

- Peer-to-peer (P2P)

Description:

An architecture that partitions tasks or workloads between peers or nodes, where no centralized node exists. Data is shared amongst all the nodes and it is easy to remove or insert nodes into the system without affecting the system.

Pros:

- More reliable as central dependency is eliminated
- The over-all cost of building and maintaining this architecture is comparatively very less than client-server

Cons:

- Weaker security over data than that of client-server
- Data recovery or backup is very difficult
- Difficult to update applications over this type of architecture

- Rule-based Architecture

Description:

The use of forward and backward chaining in order to determine rules that define the architecture of a system. Forward chaining observes the given data and makes decisions based on that. Backward chaining observes the goals and uses that to make decisions.

Pros:

- Easy to implement by defining separate rules in separate layers.
- Allows for modularity
- All "knowledge" in the architecture is uniform as it can be expressed in the same format defined by the rules

Cons:

- Limited amount of rules that can be defined.
- Complexity increases as the system size increases.

2. **Choice of architecture and why:** 3-tier Client-server Architecture

   This architecture provides for an easy, separate development of the back-end and front-end of the application. Data will be centralized on the server that will be used and this will provide for strong layer of user privacy and security.

   Modularity is easy as alterations to the application can be rolled out on the server side towards all clients at once. This architecture is best for this scale of application, and a lot of processing will be done server side so less workload will be put on the client.