DATA FORMATING: (perform, queries); (presented, user); (presented, user); (presented, user); (presented, user); (presented, user); (presented, employee); (presented, employee); (presented, employee); (presented, employee); (representing, unit); (representing, specialty); (representing, complaint); (representing, type); (creates, instance); (generates, assigns); (alters, data); (being, system)

PERSISTENCY: (stored); (stored); (stored, state); (store, information); (storing, complaint); (storing, complaint); (storing, data); (have, state); (have, HELP); (has, access); (retrieves, list); (retrieves, list); (retrieves, list); (retrieves, list); (retrieves, list); (retrieves, list); (retrieved, ); (retrieve, entry); (retrieve, information); (retrieve, type); (retrieving, data); (retrieves, details); (retrieving, details); (retrieving, complaints); (retrieves, data); (determine, type); (updated ); (updated); (updated ); (updated); (update); (informing, user); (described, login); (changed ); (queried ); (queried ); (queried ); (left, state); (left, state); (left, state); (updated, system); (updated, system); (updated, system); (saves, complaint); (saved, system); (saves, data); (saved ); (stores, information); (stores, information);

CONSISTENCY: (ensures, information); (ensures, information); (ensures, information); (ensures, information); (ensures, information); (ensures, data); (ensures, data); (ensures, data); (ensures, data); (ensures, data); (ensures, data); (ensured ); (ensured ); (assured ); (assured ); (confirms, operation); (confirms, update);

DISTRIBUTION: (sent, server); (sent, server); (sent, server); (sent, server); (sent, server); (sent, server); (sent, server); (sent, server); (sent, server); (sent, server); (sent, server); (retrieved, ); (transmitted, server); (sending, data);

SECURITY: (validates, password); (validate, employee); (validated, system);

ERROR HANDLING: (ocurrs ); (ocurrs ); (ocurrs ); (ocurrs ); (ocurrs ); (dealt, complaint); (verifies, data); (verifies, data);

**Towards the architectural definition of the Health Watcher system with AO-ADL**

La arquitectura propuesta es la siguiente:



Figura Arquitectura propuesta por Pinto y otros

|  |  |  |  |
| --- | --- | --- | --- |
| Aspecto | Atributo de Calidad | Correcto? | Justificacion |
| Data Formatting | Usability | Si | (1) |
| Persistency | Security | Si | (2) |
| Consistency | Security | No(Avialability?) | (3) |
| Distribution | Availability | Si | (4) |
| Security | Security | Si | (5) |
| Error Handling | Security | No(Aviability?) | (6) |

(1)

El modulo Data Formatting (Figura 1) es invocado cada vez que la información es mostrada al usuario.

(2)

El modulo Data Store (Figura 1) pareciera el relaciondo con este aspect, el cual contiene metodo de Seguridad. En palabras de la autora:“In the Data Store component all the data about complaints, employees, health units, specialities and so on is saved. This component has a provided interface to set and get information from the HW System, and two provided interfaces to provide services to the Enrollment ’base’ component and to the Authentication ’aspectual’ component. Notice that this means that the Authentication component (see Section 3.1), which plays the role of an ’aspectual’ component in its interaction with the UserGUI component, is playing the role of a ’base’ component in this interaction with the Data Store component.”

Al proveer infomación al módulo Authentication, se asume que esta (nombre de usuario, contraseñas, etc) debe poseer mecanismos de seguridad, por lo que no esta mal la relación con Security, aunque es cierto que quizás otro atributos puedan complementar a éste.

(3)

Similar a DataFormatting (Figura 1), pero al revés. En este caso la información es chequeada cuando entra al sistema. “The Consistency ’aspectual’ component is applied, during the interaction among the UserGUI and the HW System components, every time the HW system has to check the consistency of the received data”. Disponibilidad pareceria ser el atributo adecuado.

(4)

El módulo “Distribution” (Figura 1) es donde se ve la necesidad de elegir alguna táctica para satisfacer disponibilidad. Es en el módulo “Replication” (Figura 1) donde este problema se ataca. La autora afirma: “In order to provide Avialability, the Replication component stores a copy of the data the HWS system sends….”

En el trabajo de Haimei Zhang y otros (Figura 2) también se propone una arquitectura al problema del HWS. Entre los concerns detectados se destaca:

“Dependability and Availability: A serve or a processor can fail, usually in various ways, and must be built reliable using internal redundancy so that the register service remains available”

Entonces, él detecta el atributo de calidad en ese context. Justamente el aspecto tiene pares de verbo-objeto como (sent, server), (transmitted, server), (sending, data), por lo que se esta refiriendo justamente a eso.

(5)

La arquitectura incluye un módulo “Authentication”. La función del mismo es “interceptar la comunicación entre el empleado y el sistema HWS para chequear las credenciales del usuario cuando la operación lo requiera”. Teniendo en cuenta los pares del aspecto temprano y el atributo de calidad relacionado con este aspecto, se puede afirmar que lo detectado es correcto.

(6)

En la Figura 1 aparece un modulo “Error Handling”. Basicamente su funcionamiento es “Concretely, there are different exceptional situations in the HW system that require different solutions, such as communication problems, data storing and retrieving problems, invalid data problems and data consistency problems. All these problem categories can be represented as provided interfaces of the ErrorHandling”. No parece muy clara la relación con Seguridad, aunque por las palabas del aspecto temprano mucho mas no se puede hacer. Pareciera que seria mas clara la relación con Disponibilidad, ya que este modulo quizás debería implementar tácticas de “detección de fallas”(Fault Detection).



Figura Arquitectura propuesta por Zhang y otros