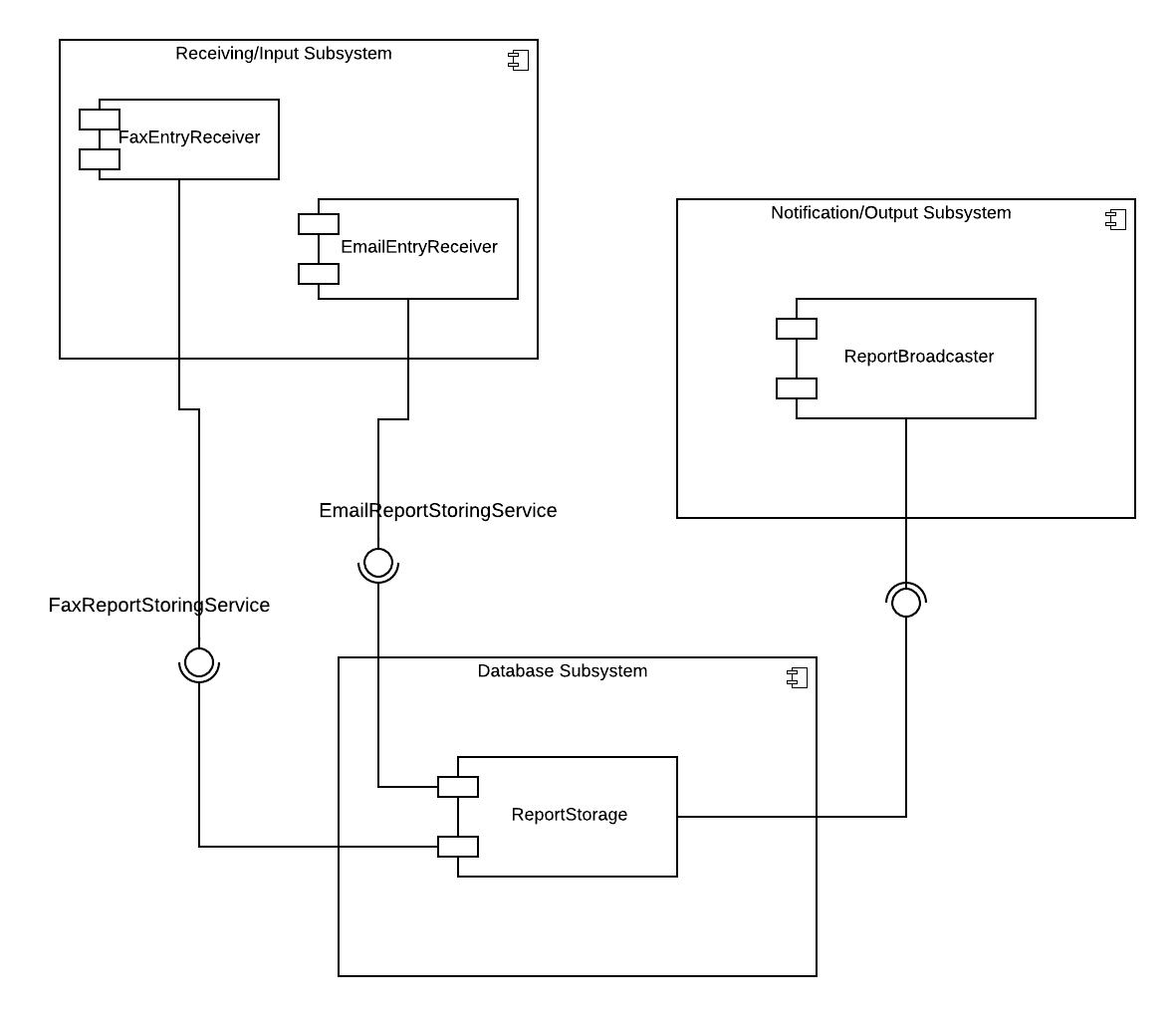
**Jeremy Cooley**

1. Consider a legacy, fax-based, problem-reporting system for an aircraft manufacturer. You are part of a reengineering project replacing the core of the system with a computer-based system that includes a database and a notification system. The client requires the fax to remain an entry point for problem reports. You propose an E-mail entry point. Describe a subsystem decomposition that would allow both interfaces. Note that such systems are used to process many problem reports per day (e.g., 2000 faxes per day).



1. You are designing the access control policies for a Web-based retail store. Customers access the store via the Web, browse product information, input their address and payment information, and purchase products. Suppliers can add new products, update product information, and receive orders. The store owner sets the retail prices, makes tailored offers to customers based on their purchasing profiles, and provides marketing services. You have to deal with three actors: StoreAdministrator, Supplier, and Customer. Design an access control policy for all three actors. Customers can be created via the Web, whereas Suppliers are created by the StoreAdministrator.

|  |  |
| --- | --- |
| **Access Control List** |  |
| **Actor** | **Operation** |
| StoreAdministrator | SetRetailPrice() |
| “ | MakeTailoredOffer() |
| “ | ProvideMarketingServices() |
| Supplier | AddNewProducts() |
| “ | UpdateProductInfo() |
| “ | ReceiveOrders() |
| Customer | WebStoreAccess() |
| “ | BrowseProductInfo() |
| “ | InputBillingInfo() |
| “ | PurchaseProducts() |

1. Select a control flow mechanism you find most appropriate for each of the following systems. Because multiple choices are possible in most cases, justify your choices.
   1. a Web server designed to sustain high loads
      1. **I would choose multithreading because each new user/client needs to be able to access their own of the client, or maintain their own browser history and shopping cart like on Amazon. This allows multiple users to see the same version of the main webpage, albeit tailored to them. It also allows each user to spend however long they like on a page, as the main server processor wait for a user’s input on their own thread.**
   2. a graphical user interface for a word processor
      1. **I would choose Event Driven Control because the word processor should wait in a main loop until the user triggers an event like pressing a button.**
   3. a real-time embedded system (e.g., a guidance system on a satellite launcher).
      1. **For this, I would use multithreading again because each subsystem of the guidance system can be set up to run at specific intervals, for instance, priority levels can be set so that the GPS Location Service runs every 20 milliseconds, while the 3D Orientation System runs every 5 milliseconds to process micro-adjustments and commands. Thus. Methods that write data can also be synchronized be locked to avoid concurrent processing and data corruption.**