<https://www.academia.edu/44649891/Relating_Events_Messages_and_Methods_of_Multiple_Threaded_Objects>

In Ruth Breu and Radu Grosu’s “Relating Events, Messages and Methods of Multiple-Threaded Objects”, Greu and Grosu talk about an object model that might help make sense of the interrelation between events, messages, and methods. In Breu and Grosu’s own words, “this method integrates the description of the object’s life-cycles with the specification of the object’s methods” (1). The paper uses an example of a bank for most of its examples, the most important to be being “The Object Model” section of the paper. They create a model on page 5 that reminds me of the factory pattern, creating and showcasing a way for the banking system to be created. I think this paper as a resource is useful because it showcases OO patterns in a real case and explains thoroughly how they work. It also helps that the paper speaks on multithreading, which is very important currently with high demand processes emerging. I have not used this paper in the past but considering projects that I am in that require multithreading, I believe that it will become useful (especially during planning stages).

<https://www.academia.edu/44325488/A_Model_Driven_Architecture_Approach_Using_Explicit_Stakeholder_Quality_Requirement_Models_for_Building_Dependable_Information_Systems>

In Stefan Biffl, Richard Mordinyi, and Alexander Schatten’s, “A Model-Driven Architecture Approach Using Explicit Stakeholder Quality Requirement Models for Building Dependable Information Systems”, the authors speak on how to use design patters, or requirement models, to build well structured information systems. They describe why an architecture that is model driven that is supposed to “improve the quality of system requirements” (1). Overall, this paper lays out a way for people to build systems in a way that makes them dependable and work more efficiently by declaring what parts of a system have what tasks. This is eerily similar to OO programming as it tries to map out functionality in the best way possible using a pattern or requirements model. I think this paper is a useful reference as it lays out a full example of a “Harbor Traffic Control” scenario as a model and will be easy to reference when needing to find a MDA, or pattern, of a similar system. If you go to Figure 3 on page 5 of the article, you can see a diagram that looks and acts very much like an architecture diagram, which we have used in this class and I have used in multiple classes/projects before.