Documenting System Specifications through OPM Web-Based Graphics/Text Equivalence: the Case of the Free Flight System

Dov Dori
Massachusetts Institute of Technology
dori@mit.edu

Abstract

The generation of bimodal formal system specification documents that bi-directionally complement graphics from natural language and vice versa through Object-Process Methodology (OPM) is presented. A sample of free text paragraphs from a document that describes the concept of Free Flight as part of the National Airspace System are converted to OPM specification. We then discuss the requirements of porting the OPM application to a Web environment, the challenges and the expected benefits.

Background

Companies in all domains are expending huge resources to develop new systems and products as they adapt to market demands and new technologies. However, lacking a paradigm, a methodology and a software environment that enables customers and technology suppliers to collaboratively engage in the ongoing development process, the current practice in systems development has been at best the exchange of free text documents.

A major problem in systems development is a lack of means for effective ongoing communication and agreed-upon documentation between the customer side and the supplier/developer side. Addressing this need requires a collaborative software environment not just for developing software systems, but also for systems that encompass hardware, humans, and business rules and practices along with IT-oriented or embedded/real time software. The Web is an ideal infrastructure for such an environment, as it enables stakeholders and teams in various locations and time zones to engage in collaboration that is enhanced by the unique features of the Web. Using an environmet with such features, executives and non-IT domain experts on the customer side can be actively involved alongside the developers on the supplier side in specifying the system of their desire.

Providing the "front end" for customer-developer communication, can add value to software development products that follow UML - Unified Modeling Language - the industry standard in developing software systems. UML is too complex to be used by non-IT professionals as it caters to software developers and requires nine different diagram types. Object-Process Methodology (OPM)[1,2] achieves model integration by incorporating the three major system aspects – function, structure and behavior, into a single model, in which both objects and processes are adequately represented without suppressing each other. When implemented in a Web environment, OPM provides the foundation and platform for supporting systems design and evolution throughout their entire lifecycle. It can be used to involve customers, manage requirements and documentation, simulate, verify, and enable controlled changes. By using simple graphics and natural language in a truly unified and intuitive model, OPM provides a missing communication means among customers, technology providers, and professionals from the various engineering disciplines who are involved in system development processes.

This paper demonstrates the generation of OPM-based system documentation in an integrated graphics-text environment as it is being converted from a free-text description to a formal specification. The system elected to be modeled is the Free Flight System, which is described below. We conclude by discussing the main characteristics of Web-based documents and systems and the additional benefits of applying OPM on the Web.

The Free Flight System

Free Flight is an innovative concept born out of the need for increased user flexibility, operating efficiencies, and safety to meet the growing demands for air transportation. RTCA, Inc.[3] is a private, not-for-profit corporation that develops consensus-based recommendations regarding communications, navigation, surveillance, and air traffic management (CNS/ATM) system issues. RTCA functions as a federal advisory committee. Its recommendations are used by the Federal Aviation Administration (FAA) as the basis for policy, program, and regulatory decisions

The RTCA Task Force on Free Flight developed the Free Flight concept during 1995. The Free Flight concept suggests that significant benefits can be achieved by concentrating on (1) removal of constraints and restrictions to flight operations, (2) better exchange of information and collaborative decision making among users and service providers, (3) more efficient management of air-space and airport resources, and (4) tools and models to aid service providers. National Airspace System (NAS) - Concept of Operations[4] is a document that refines the Free Flight operational concepts.

This paper uses small portions of the this NAS Concept of Operations document for the purpose of demonstrating how a free-text specification is amenable to a formal, yet intuitive system design and documentation through OPM. The document describes the evolution of the NAS in its near-term (through 2005), mid-term (2005 - 2010), and far term states (2010 - 2015). Selected paragraphs from the NAS Concept of Operations document are cited below. For each such paragraph, cited in italics inside a frame, an Object-Process Diagram is generated, which automatically yields a corresponding Object-Process Language script.

> *There is increased information sharing and collaboration among users and service providers. Collaboration includes information exchange, plus shared and active user participation in decision making. For situations such as demand-capacity imbalances or severe weather, collaboration supports determining when, where, and how transitional route structures are established in the airspace to meet a short-term problem. Collaboration also supports strategic problem resolution. All parties share common situation awareness, using the best information possible.*

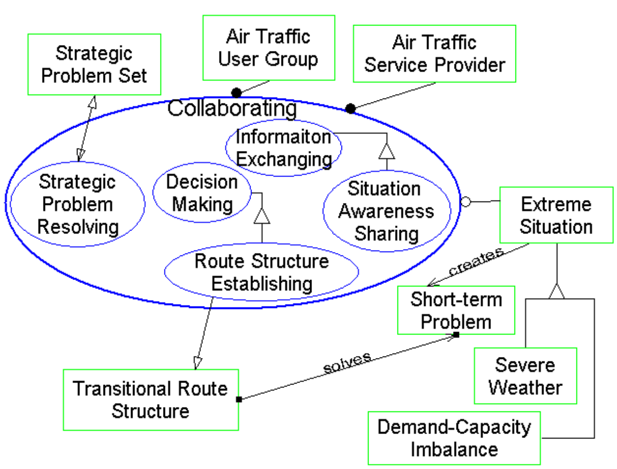

Figure 1. Object-Process Diagram (OPD) of the near-term goals of National Airspace System Evolution

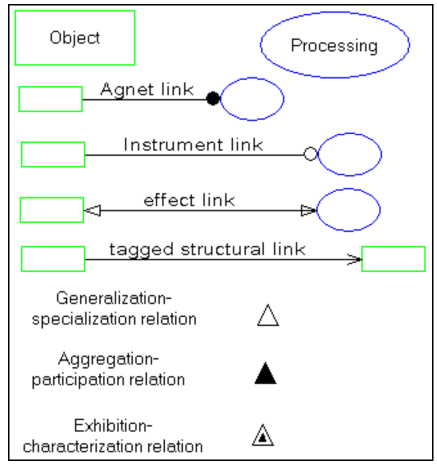

Figure 2. Legend of basic OPD symbols

Figure 1 is an Object-Process Diagram (OPD) of the near term NAS Evolution, while Figure 2 is a legend showing the basic OPD symbols.

> **Air Traffic User Group** and **Air Traffic Service Provider** handle **Collaborating.**
> **Sever Weather** and **Demand-Capacity Imbalance** are **Extreme Situations.**
> **Extreme Situation** creates **Short-term Problem.**
> **Collaborating** requires **Extreme Situation.**
> **Collaborating** zooms into **Information Exchanging, Decision Making,** and **Strategic Problem Resolving.**
> **Strategic Problem Resolving** affects **Strategic Problem Set.**
> **Situation Awareness Sharing** is **Information Exchanging.**
> **Transitional Route Structure Establisging** is **Decision Making.**
> **Transitional Route Structure Establisging** yields **Transitional Route Structure.**
> **Transitional Route Structure** solves **Short-term Problem.**

One can trace back the sentences in the document to the OPL setneces. For example, the document's free text sentence "*There is increased information sharing and collaboration among users and service providers*" has been converted into the OPL sentence **Air Traffic User Group** and **Air Traffic Service Provider** handle **Collaborating.** The non-bold phrases are reserved OPL words, while the bold ones are non-reseved. An example of a reserved word is handle, which stands for agnet, defined as a human or group of humans that are involved in the process without being affected. This word is the textual equivalent of the two agent links in Figure 1 that look like black lollipops connecting the object **Air Traffic Service Provider** and the object **Air Traffic User Group** to the process **Collaborating**. The bold words, which make up the non-reserved phrases, are taken from the names of objects and processes in the free text with the needed modifications required by OPL, such as changing *collaboration* to **Collaborating**, as process names in OPL must be in the gerund form (i.e., end with the ing suffix). It should be pointed out that not all the ideas in the free text citation are modeled. In our example, the word *increased* is not accounted for. This word implicilty refers to an existing situation, with respect to which the collaboration is increased. For now we are satisified with being able to model the basic structure and behvior that is implied in the free text and leave out subtleties such as the one discussed. The next OPD is

> *"For Traffic Flow Management (TFM) Initiatives, users are allocated available capacity based on the affected resource (e.g. airport, airspace, etc.) in the form of an arrival interval and the designated number of flights they may have arrive in that interval."*

based on the following excerpt.

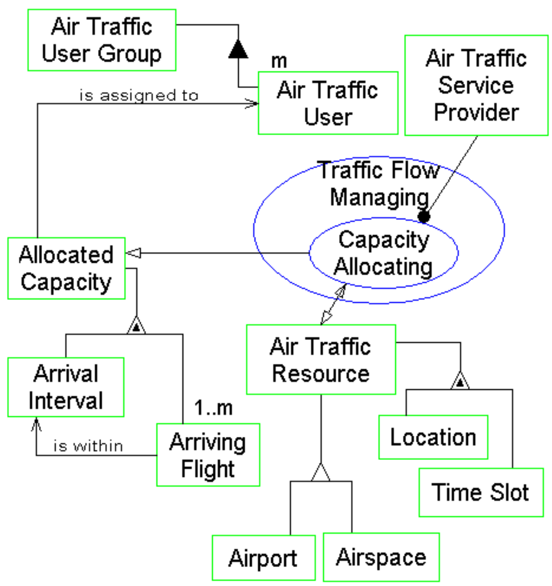

Figure 3. Traffic Flow Management initiatives

The OPD in Figure 3 and the corresponding OPL paragraph below are the part of the system that is modeled as faithfully as possible to reflect the free text citation.

> **Air Traffic User Group** consists of many **Air traffic Users.**
> **Traffic Flow Managing** zooms into **Capacity Allocating.**
> **Air Traffic Service Provider** handles **Capacity Allocating.**
> **Capacity Allocating** affects **Air Traffic Resource.**
> **Airport and Airspace** are **Air Traffic Resources.**
> **Air Traffic Resource** exhibits **Location** and **Time Slot.**
> **Capacity Allocating** yields **Allocated Capacity.**
> **Allocated Capacity** exhibits **Arrival Interval** and from one to many **Arriving Flights.**
> **Arriving Flight** is within **Arrival Interval.**

Again, every sentence in the OPL script can be mapped to the original free text, as well as to an exact graphic construct in Figure 3. Interestingly, the OPL script is almost twice as long as the free text. This is reasonable in view of the fact that even parenthetical remarks or

> *"Specialized users begin operating tilt-rotor craft having unconventional transition states that place them in a category between helicopters and fixed wing. In the initial years of operation, tilt-rotor craft operations tend to be few; localized, and follow traditional traffic flow. But as additional commercial uses are found for these vehicles, new operational regulations capitalize on their inherent capabilities."*

examples in the free text, such as (e.g. airport, airspace, etc.) are explicitly modeled.

Next, the document discusses the incorporation of new types of aircrafts such as tilt-rotor craft. The OPD in Figure 4 and the corresponding OPL script reflect the taxonomy of the various aircraft types by classifying them through an attribute of their wings called Relative Movement, which can be fixed (for airplanes), tilting-rotating (for tilt-rotor carfts) and rotating (for helicopters). The pharases "**fixed**", "**tilting-rotating**" and "**rotating**" are three different values (or states) that the attribute **Relative Movement** of **Wing** can be at. This OPM formalism demonstrates the power of the methodology to get to the bottom of where differents among various objects arise from and provide a sound framework for their classification which is both formal and intutive.

Each OPL sentence is of some type. For example, the sentence "**Helicopter Wing** is a **Wing,** the **Relative Movement** of which is **rotating.**" is an instance of an OPM *qualification sentence*. The system that generates the OPL sentences from the graphics constructs this sentence by first recognizing a pattern where a root object (**Wing** in our example) exhibits an attribute (**Relative Movement**) that has states (**fixed, rotating**…) such that each state is linked with a qualification relation to a specialization of the root object (e.g., the state fixed is linked with a qualification relation to the object **Airplane Wing**, which id a specialization of the root object **Wing**).

> **Aircraft** consists of between **2** to **3 Wings, Body,** and between **1** to **6 Engines.**
> **Wing** exhibits **Relative Movement.**
> **Relative Movement** can be **fixed, tilting-rotating,** and **rotating.**
> **Airplane Wing** is a **Wing,** the **Relative Movement** of which is **fixed.**
> **Tilt-Rotor Craft Wing** is a **Wing,** the **Relative Movement** of which is **tilting-rotating.**
> **Helicopter Wing** is a **Wing,** the **Relative Movement** of which is **rotating.**

Features of Web-based OPM Documents

Web applications are client-server systems in which the client side of the application is limited to browser capabilities. Requirements and benefits of applying the OPM system over the Web are as follows.

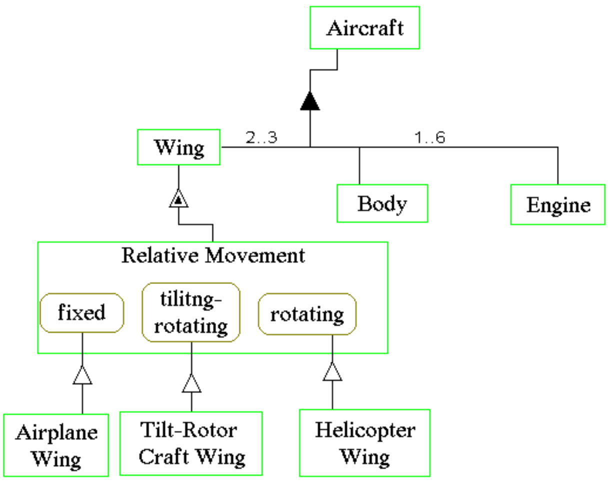

Figure 4. Operation of Unmanned Aerial Vehicles (UAVs)

Complex dynamic and distributed architecture – The general architecture of Web applications is client-server, which consists of three components: a Web server (or several servers), a network connection (mainly http) and client browsers. The Web-based OPM system enables multiple designers to engage in system design and bimodal (graphics/text) documentation as clients, with the server hosting the system database of objects, states, processes, and relationships among them.

Heterogeneously skilled users – The Web can allow universal access to public-domain system specifications for view by users of all skill levels. This requires complex GUI that is flexible enough to accommodate the needs and expertise of the novices as well as the most experienced system architects.

Security and privacy support – OPM system designs usually feature corporate proprietary information that needs to be restricted to a select group of users. ISO 7498-2 (1989) defines five main categories of security services: authentication, access control, confidentiality (privacy), data integrity, and non-repudiation. Applying the appropriate security and privacy features will enable secure development of disparate design groups over geographic and time zones. Various levels of view/modify for the various groups will ensure that while all the stakeholders can access some level of detail, alterations of each subsystem are allowed only to those developers charged with the responsibility for that subsystem.

Heterogeneous information sources – Web applications must handle and integrate heavy, complex, hierarchical data, as well as unstructured or semi-structured data. The data may be stored in different systems and distributed over multiple sites, for example in a distributed DBMS, which increases security, provides for shared access, and boosts performance. The Web-based OPM application will need to link each such multimedia object to the actual object or process in the system, such that users can access these multimedia objects to get complementary information on the things (objects or processes) to which they are attached.

Change Management– Updates of data input into Web applications are done in real-time or near real-time. The OPM Web application will need to employ such techniques as publish/subscribe to ensure that at any given point in time only one person can have the authority to change a subsystem, and take care of reconciling and/or propagating any possible effect on other subsystems. At the same time, the system should provide for real-time collaborative systems development.

Summary and Future Work

This paper has demonstrated to value of bimodal text/graphic documentation that are geenrated in real time by an OPM system. This systems development environment opens up a host of new possibilities for collaboration among the various stakeholders in the development of systems, as it brings together domain experts and systems architects that can communicate with each other using simple graphics and natural language. The bimodal nature makes it less likely for a design error to go undetected when developers on the customer and supplier side concurrently look at the design as it unfolds in both graphics and text. This saves rework and expedites time to market. Currently we have a single-user systems that does the text-graphic bidirectional complementing. Porting it to a multi-user system over the Web, as dicussed above, is both challenging and rewarding, and we are working to accomplish this.

References

[1] Dov Dori, Object-Process Analysis: Maintaining the Balance between System Structure and Behavior. *Journal of Logic and Computation*, 5, 2, pp. 227-249, 1995.

[2] Dov Dori, Object-Process Methodology - A Holistic Systems Paradigm, Springer Verlag, Heidelberg, New York, 2001 (ISBN 3-540-65471-2; Hardcover, in press).

[3] http://www.rtca.org/

[4] RTCA Select Committee for Free Flight Implementation National Airspace System - Concept of Operations. RTCA, Incorporated, 1140 Connecticut Avenue, NW, Suite 1020 Washington, DC 20036-4001 USA, December 2000.