## Articulating Objects

**Abstract**
Jacob W. Jespersen

Interactive computer systems are the enabling technology in an increasing part of our lives; part of this story is about how people involve interactive systems in their businesses. Here, interactive systems are commonly a prominent part of people's practices, and thereby influence how people think about their work.

This is also the case when someone's job is to develop interactive systems for others to use. To this end, system developers use *development environments*, which are specialized interactive systems that work as *tools* in the construction of new systems. Today, such tools have an overly general perspective on the parts of an interactive system that most directly concerns interactivity: the *user interface*. The common approach is to view the user interface as just another subsystem that must be built and somehow linked to the parts of the system that need or feed data. This general perspective fails to recognize the complementary relationship between a system's user interface and its functionality, between form and function, so to speak.

In what is an alternative perspective, I investigate an approach to graphical user interface development that is closely tied to a business domain model. With inspiration from semiotic theory, and knowledge from the field of user interface design, I propose a human-computer interaction model based on the concept of *articulated objects*. The basic premise is that *business objects* represent business phenomena (encoded according to a domain model), and that the tasks people carry out with business systems, e.g. receiving payments, invoicing, etc. are possible via interactions with business objects. Thus, I conceive people interacting with the computer by manipulating articulated business objects via editing their properties and invoking their methods.

I have investigated what *articulation* entails in such a scenario, and on this basis designed a programming language – YUIO - that operationalizes this concept. My work contributes to interactive systems development research with an improved understanding of how it is possible for developers to construct user interfaces on the basis of business objects. The intention is *not* to bypass user interface design, e.g. activities such as task analysis, visual design, and user involvement. On the contrary, I envision YUIO programming to integrate well with these activities. Specifically, my intention is to support developers in expressing, maintaining, and evolving how business objects – conceptually - are user interfaces to themselves. In this respect, I intend the YUIO approach to complement, and in some cases replace, the traditional style of user interface programming in development of interactive systems for business.