Internet-of-things (IoT) systems consist of spatially distributed devices with services. Compared to desktop applications, IoT systems are always running and need to deal with unresponsive devices and weak connectivity. In this thesis, we examine the following question: How can we simplify the development of IoT systems? We begin to answer this question by proposing a domain-specific language (DSL), called ComPOS, for composing IoT services. In the DSL, the user specifies the reaction to a message. The reaction can be programmed to request and receive responses in sequence and parallel. ComPOS can abort a running reaction when a new message arrives; this is to support unresponsive devices and weak connectivity. We demonstrate our language by creating a bird-spying system that takes photos of a garden and then stores the ones containing a bird. The ComPOS editor supports live programming for programming a running system. Programming in our DSL is divided into three phases: finding services (explore), composing services (assemble), and abstracting compositions as new services (expose). When developing a DSL, it takes effort specifying the syntax and semantics, building the editor, and integrating with the middleware. To reduce the effort needed to experiment with our DSL, we have created a tool called Jatte. The tool is a generic projectional editor that can be tuned for different languages using attribute grammars. We have integrated the editor built with the tool into an IoT development environment supporting discovery of devices and services.