THESIS SUMMARY

In the contemporary Internet of Things (IoT) era, people can interact with a multitude of smart devices, always connected to the Internet, in the majority of today's environments. With lamps, thermostats, and many other appliances that can be remotely controlled, homes and workplaces are becoming "smart." Furthermore, by using PCs and smartphones, users can access a variety of online services, ranging from social networks to news and messaging apps. The result is a complex network of *connected entities*, be they physical devices or virtual services, that can communicate with each other, with humans, and with the environment. This complex scenario opens up, at the same time, possibilities and issues.

By taking advantage of End-User Development (EUD) solutions, users can actively participate in the IoT by *personalizing* the functionality of their connected entities. Nowadays, in particular, many different visual programming platforms such as IFTTT and Zapier allow the personalization of the joint behaviors of connected entities through IF-THEN rules, i.e., in the form of "*if something happens on a device or a service, then execute an action on another device or service*." The growing spread of new smart devices and online services, however, makes this personaliziation a complex task, especially for users without programming experience. The trigger-action programming paradigm, indeed, is typically implemented at a low-level of abstraction, with representation models that strongly depend on the exploited technologies. This negatively influences the rule definition process: end users experience difficulties in finding and managing the functionality they are interested in, and they are likely to introduce dangerous run-time errors in the defined IF-THEN rules.

Stemming from these issues, this thesis presents a set of research works that aim at assisting end users in easily and efficiently personalizing the functionality of their connected entities. Through rigorous user studies and controlled experiments, we report on different approaches and practical solutions to *a*) simplify the definition of IF-THEN rules, *b*) promote the discovery of new rules and related functionality, and, *c*) enable the identification and the resolution of run-time problems in IF-THEN rules. For supporting these results, we first define a semantic representation, named *EUPont* (**E**nd-**U**ser **P**rogramming **ont**ology), to model abstract and technology-independent IF-THEN rules that can be adapted to different contextual situations. We demonstrate that *EUPont* is more expressive than the representation models offered by contemporary trigger-action programming platforms, and that it improves the processes needed by end users to define IF-THEN rules in terms of time, understandability, and ease of use.

After presenting *EUPont*, we then report on how we used it to facilitate users in discovering and managing rules and related functionality. We present, in particular, two different approaches, namely *EUDoptimizer* and *RecRules*. *EUDoptimizer* is an optimization tool that adopts semantic optimization methods to dynamically redesign layouts in trigger-action programming user interfaces on the basis of the choices made by the user during the definition of a rule. The tool is based on *SDP-FSM*, a predictive model for trigger-action programming that exploits *EUPont* and a state-of-the-art model of human performance in menu search named Search-Decision-Pointing. *RecRules*, instead, is an innovative

recommender system of IF-THEN rules that, by exploiting *EUPont*, is able to compute suggestions on the basis of the final behaviors users would like to define, e.g., increasing the temperature in a room.

Finally, we explore the urgent need of assessing the correctness of IF-THEN rules by presenting two end-user debugging tools, i.e., *EUDebug* and *My IoT Puzzle*. By exploiting *SCPN*, i.e., a novel formalism based on Petri Nets and the *EUPont* model, such tools are able to assist users in identifying possible loops, inconsistencies, and redundancies that their rules may generate at run-time. *EUDebug*, in particular, is built on top of an IFTTT-like interface, while the *My IoT Puzzle* exploits the Jigsaw metaphor, and it has been designed on a set of guidelines extracted from previous work on end-user debugging in different contexts.

Summarizing, the main outcomes of this thesis are the following:

EUPont, an ontological high-level representation for end-user development that allows the definition of abstract and technology-independent IF-THEN rules that can be adapted to different contextual situations, independently of manufacturers, brands, and other technical details.

EUDoptimizer, an optimization tool to dynamically redesign layouts in triggeraction programming interfaces in an interactive way, i.e., by considering the choices made by end users during the definition of a rule.

RecRules, a hybrid and semantic recommendation system of IF-THEN rules that allows users to discover new rules on the basis of the underlying functionality, rather than the involved brands or manufacturers.

EUDebug, an end-user debugging tool built on top of an IFTTT-like interface that enables end users to debug their IF-THEN rules at definition time.

My IoT Puzzle, an end-user debugging tool to compose and debug IF-THEN rules though the Jigsaw metaphor, designed according to a set of guidelines extracted from the literature.