

Rules for my personal assistant: IFTTT

It was one of the first times I had travelled alone by train. Nobody would pick me up, so the map of the destination was imprinted as a picture in my memory. Sitting in the train I daydreamt about the idea of a portable phone so that I could chat with my friends. The word 'mobile' only meant 'being able to move' and 'walkman' was the only portable device I knew.



The portable phone that I imagined soon became reality and I daydreamt about this device becoming my personal assistant: answering appointment requests with polite messages and suggesting alternative time-slots when I would not be available.

Of course this assistant should be acting according to my rules:

- A meeting with more than one hour traffic time must not start before 10 AM.
- Motivation: my deep abomination of traffic jams.
- Applicability: every month except July and August.

The time has come to turn my fantasy to reality.

This is the fifth post in a series on the way rules are used in apps and software 'for the millions'. Earlier posts dealt with [Outlook](#), [Salesforce](#), [Shopify](#), and [Excel](#).

Workflow automation apps on our mobile devices use rules to automate tasks performed on or with your mobile. In this post we examine IFTTT, launched in 2011 as one of the first applications in this category. In 2015 it was re-branded under the name IF and relates to a suite of apps named DO. Similar tooling is offered by Zapier (integration software) and Microsoft Flow (integrate Microsoft Apps)

Condition - Action rules

"IF This Then That" are the words that IFTTT is an acronym for and says it all: rules in IFTTT have this structure and perform an action when some condition is met. These conditional structures are named 'applets'. This sounds rather technical. That is probably the reason why they are also named 'recipes'.

The conditions consists of triggers that reflect a change in an app on your mobile like receiving a mail, getting an invite, being named in a Facebook post and the updating of the weather forecast. The web services that have implemented the IFTTT triggers are named *channels* and are offered by brands like: Gmail, Facebook, BMW and Philips HUE.

The cooking analogy is taken further since the data that is contained in a trigger is named 'ingredients'.

The actions automatically let your mobile do something like sending an SMS, tweeting a picture, opening a car (hopefully yours), or turning on the lights. These services are typically offered by the same partners that offer triggers.

The commercial model is that partners pay for offering a channel, get analytic data in return and will be promoted by IFTTT.

To see if I could realise my rule of having my personal assistant respond to appointment requests, I first search for a service that has a trigger on a calendar invitation. Since the search does not deliver any results I browse through the long list of recipes related to 'calendars and scheduling'. There is no existing 'recipe' for my rule so I create it myself.

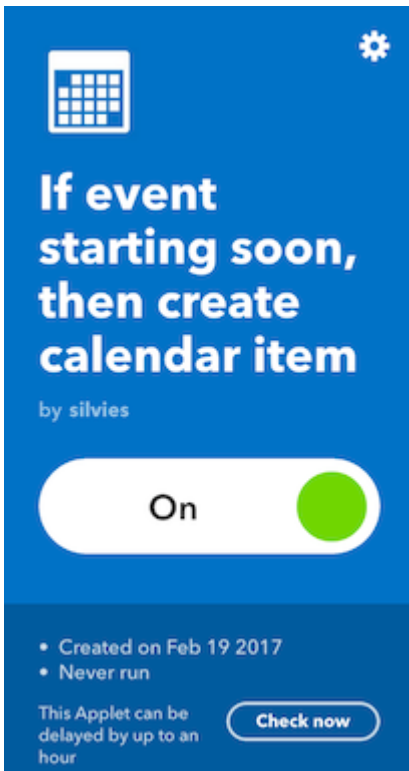
The presentation of the rule at first looks attractive to me:

if  **this** **then that**

It's clear to me that I need to click the + sign to add a trigger. Once I do that I need to:

1. select a channel
2. provide my account details for the channel (in this case office365)
3. select a trigger
4. select an action channel (with account details)
5. select an action
6. add ingredients to action

It all feels very limited and technical in the end. Office365 offers one trigger (event starts in 15 minutes) and one service (create new calendar item). The ingredients are all static text. There can only be one condition and one action and the resulting rule is silly and reads:



If event starting soon, then create calendar item.

I would never use this personal assistant.

What could be improved from a business rules perspective?

Use normal (business) language

A **nerd** programs an applet and a **chef** makes a recipe but **people with dreams like me** react on events, have too many tasks, want to automate them and are willing to provide instructions to a computer or configure rules in an app. There is no need for new vocabulary. IFTTT terminology is too technical for a broader audience than those who live or would like to live in Silicon Valley.

Make a concept model

The list of channels to select a trigger or action service is impressive and colorful:



... but very inconvenient to use. Yes, you can search, but what are the keywords? And how do I write a general rule about calendar item reminders for Microsoft, Google and every new channel for calendars? I am missing a semantic layer to organise the services. I am missing a concept model.

Offer more expression power

The rules that I would like to create for my personal assistant need more than one trigger and more than one ingredient. The action depends on the date, time, location of the event and travel time to the event. So I would need to combine information from different channels in one rule condition. Or maybe I would like to have my personal assistant respond differently for different combinations of these 'ingredients' like in the following decision table.

Appointment start time	Traffic Time to meeting	Response and action by PA
before 10:00 AM	between 1 hour and 2 hour	Decline. Suggest to start one hour later.
after 10:00 AM	between 1 hour and 2 hour	Accept.
after 9:00 AM	more then two hours	Tentative. Please call me.
before 9:00 AM	n/a	Decline.

This table also provides a much better overview when you have multiple rules. In other words, we need more expression power and a better way to present rules.

So the business rules perspective provides some ideas to improve the upcoming feature and personalise automated actions on your mobile.



To close, be advised that the recommendations for IFTTT are recommendations for every tool and every organization dealing with rules. **Remember:** rules should be expressed in natural language, based on a conceptual model and be directly executable.

Next in this series is about rules making good interviews and analysing the results in SurveyMonkey.

