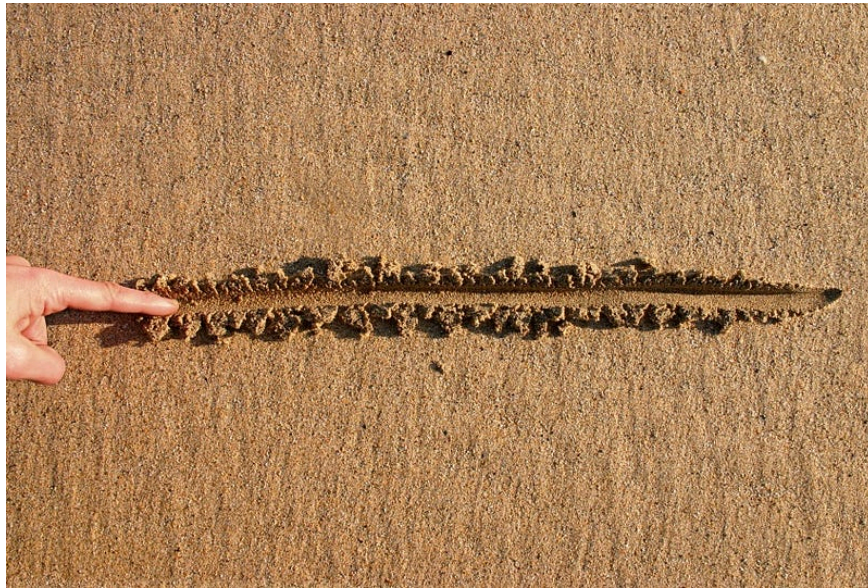


Lines in the Sand: Defining the Boundary Between Rules and Code



When introducing a rule engine to the IT side of the aisle, one of the questions that often comes up is how to codify the distinction between logic that should be programmed directly into an application or CRM platform, and logic that should be written in a rule application on the decision platform.

IT teams are dedicated to maintaining control and managing all aspects of technology in the organization. Therefore, the concept of disrupting that structure to introduce a decision platform to handle some of the functionalities of the infrastructure can be a difficult one to accept. However, the fact that decision management tools are in use by most major organizations indicates that there are plenty of good reasons to allow that control to be shifted.

Considerations

Unfortunately, when it comes to deciding where that line in the sand should be drawn between what should be transitioned into a rule app and what should be programmed into an application, the

guidance we offer nearly always starts with the Consultant's classic refrain of "Well, it depends..."

There are a number of considerations to assess when deciding where that boundary should exist- for example:

- The complexity of the logic contained in the rules
- The frequency with which the logic in the rules is expected to change
- The technical acumen of your business folks
- The degree to which the components of the decision are inter-dependent and/or calculated
- The amount of externally sourced data required by the logic in the rules
- The volume of actionable "next steps" triggered by Rules

Guidelines

There are several general guidelines associated with those considerations that you can use to help work through the process of finding the right spot for your organization to draw that boundary.

1) The more complex the logic required for a decision, the better suited it is for a rule application.

If you just need to check if a field has been populated or that the date for a given event is not in the future—by all means, consider having that directly in the application code. There is a certain degree of overhead involved when executing rules; whether it's pulling in the rule app from a catalog, making a web request to an execution service, or triggering an integration, there is a certain degree of effort required to just get the rule executed. Where rule applications truly shine is when there is a complex set of logic required to make a decision. We've spent nearly two decades building up structures and tools that allow incredibly complex sets of logic to be written in a way that's easy to understand and follow—logic that would be dizzyingly complex if written purely in code. Would it work for the

simple stuff too? You bet—but it's up to you to make the call of where it makes sense for the logic to live.

2) The more frequently you expect the logic to change, the better suited it is for a rule application.

The process of introducing a change in code in your average organization is not generally a quick endeavor. It can involve a change request, backlog refinement, sprint planning, execution of the change, code review, testing, QA, and finally a scheduled push to production. One of the benefits of using a decision platform is that many those steps can be bypassed, allowing for changes to be made much faster. If your logic tends to change on a daily, weekly, monthly, or quarterly basis, using a rule app rather than hard-coded logic will reduce a lot of overhead. If your business folks are particularly technical, that makes this even more important, as they may be able to perform changes to the rules directly with little or no support from IT.

3) The more inter-connected and calculated the entities involved in the decision are, the more benefit you'll get from using a decision platform.

One of the great benefits of using a platform specifically designed for one purpose is that you get a large amount of functionality included out of the box. InRule builds a full dependency network of all fields and calculations, allowing any change in data to automatically filter down to update all fields and calculations that depend on that data in any way—without the rule author having to build out that functionality. There are many other features like decision tables and vocabulary that give you for free what you would have to hand-roll in a custom application implementing decision logic—all of which saves time, effort, and testing, as well as making long-term maintenance easier.

4) The more external inputs required by a decision, the more it may make sense to extract the data-load behavior into an external application.

Rule applications are evaluated in a sequential manner—that is, they execute linearly from start to finish. If the execution engine is

required to make repeated requests outside the pre-compiled application to retrieve additional data from outside sources (REST services, databases, etc.), that will negatively affect execution time. By loading more of that supporting data prior to initiating an evaluation of the rules, you allow the engine to execute more efficiently and improve the performance of the overall process.

5) The more actions initiated as “next steps” by the rules, the more it may make sense to extract the action steps to an external application.

This one is very closely related to the previous item. Due to the sequential nature of rule executions, if there are a number of actions needed as a result of the decision being made (calls out to a webservice, inserts into a database, etc.), it may make sense to have the rule app build up a list of steps to take, but delegate the actual execution of those actions to whatever process initiated the request and handles the response.

Side Note: CRM Integration

When using InRule as part of a Dynamics or Salesforce integration, the last two items in the list above are already being done by the part of the integration deployed into the CRM platform.

Conclusion

Every organization has unique needs and constraints that impact their ability and desire to implement rules to fulfill various roles and determining which of those should be transitioned to a decision platform is an important component in the success of adopting InRule. While the considerations listed here are by no means all-encompassing, they can help you get the discussion started around where to kick off the road to decision automation.

. . .

Originally published at <https://www.inrule.com> on June 18, 2019.

