DECISIONS

**RULES DRIVEN WORKFLOWS**

# How to Build Dynamic and Responsive Business Processes

Workflows are integral to how we do our jobs and the companies that we work for rely on them to operate effectively. Automating workflows is key to optimizing business processes, a vital part of remaining competitive. Building more dynamic and responsive workflows can set organizations apart from their competitors.

Automating workflows can not only be key to streamlining repetitive tasks it can also digitize enterprise knowledge. Workflow automation enables organizations to document 'how things are done' by standardizing processes in a defined series of steps. While this can be transformative for an organization, along with greater efficiency comes greater complexity. The vast amounts of data being generated across the globe and BPM platforms' ability to create workflows to capitalize on this data can lead to an infinite number of decisions growing the complexity of enterprise systems exponentially. Traditional workflow platforms are good at automating 80% of processes that follow a consistent and recurring pattern. Efficiently addressing the remaining 20% of situations that are unique or uncommon is where complexity can become overwhelming.

As workflows get increasingly complicated, organizations that have adopted a traditional workflow automation platform can continue to add new paths and flows to address each new case. With each new workflow built to address less regular situations, the number of flows and paths grows exponentially. This leads to a spiderweb of processes that can be overwhelming to monitor and maintain.

Or, instead of adding a new branch or path to each workflow, steps that do not follow the traditional path can just be handled manually. By reverting to manual processes, unorganized solutions emerge with data and knowledge being relegated to emails and sticky notes. This is not only inefficient but lessons learned from past instances can not be easily accessed to inform new cases.

A more dynamic and responsive rules-based approach can support more complex workflows. By more effectively executing on the hard to handle unique cases, leaders can differentiate themselves from also-rans.

## SOLUTION - HOW RULES-DRIVEN WORKFLOWS ARE DIFFERENT

Workflows can be either a traditional workflow or a rules-based flow. Traditional workflows have predefined paths or procedures. For simple sequential workflows, work moves along a linear path with each sequential step being executed when the previous step has been completed.

## Simple Sequential Workflow

Start — Action A — Action B — Action C — End

More advanced traditional workflows incorporate complex routing but the focus remains on the flow. These flows could include loops, skips, branch, and merge logic and usually incorporate rules that dictate the path that the work takes through the process. Although rules may pull data into the workflow to determine the right path, all the rules and logic are predetermined and defined within the workflow. Traditional workflows have limited flexibility to redefine the procedure based on context and real-time data.

## Complex Digital Workflow

Start → Action A → Action B → Action C → End
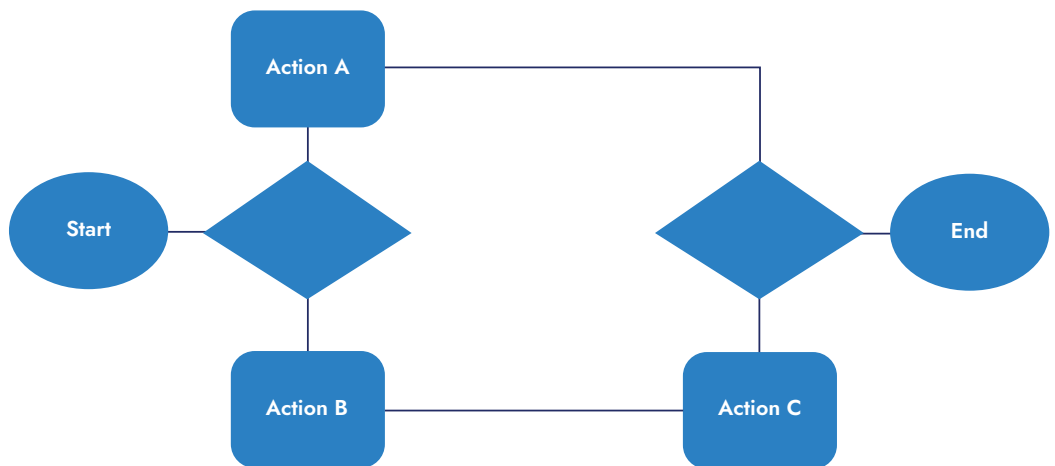
A rules-based approach can be more dynamic to handle more unique situations but sophisticated enough to avoid complex flows that overwhelm managers. By focusing on the rules, each unique possible path in the workflow is not required to be specified in the logic. By leveraging rules, only structure and constraints are needed to support an evolving process. Rules enable users to dictate the order that the steps in a workflow can be executed in, not the logic embedded in the workflow. This makes it possible for workflows to dynamically adjust to the context of the interaction so the steps in the workflow are directed by experienced professionals.

## Rules-Based Workflow

Start — Rule Engine Determines Actions — User Chooses A — Complete Action A — Rule Engine Determines Actions — User Chooses Action C — Complete Action C — End

Since the flows are not concrete, this approach may be harder to build and understand in the short term but in the long run, they are much better designed to handle changing rules, context, and complexity.

| | Simple Traditional Workflow | Complex traditional Workflow | Rules-Driven workflow |
|---|---|---|---|
| Rules | No rules | Incorporates rules | Dynamic rules |
| Flow | Linear Sequential flow | Complex flows | Contextual and user-defined |
| Integration | None | Integrated systems can add data and determine appropriate path | Integrated systems can interact with and direct the workflow process |

# How Dynamic Entities
# Work in a Rules-Driven Workflow

To understand how workflows operate we need to understand the components.

- The rules in the workflow are the mechanisms that provide the constraints that limit the actions that a user can take on an entity within the workflow.

- An 'entity' can be anything that stores data such as a document or a file

- The 'state' is the data that describes where the entity stands in the workflow process. For example, a document entity could be in a saved state or an unsaved state.

Rules determine what actions can be performed on the entity based on that state of the entity. The rules engine will evaluate the entity to determine the state of the entity and what options are appropriate. For example, the action of saving the document can only be available if it is in an unsaved state. If the document is in a saved state, delete would be an appropriate action.

The available or required options are built into the rules but the user has the ability to choose from multiple options to orchestrate the steps in the workflow. Essentially the user has the power to choose the order of what steps to take.

## Rules Engine
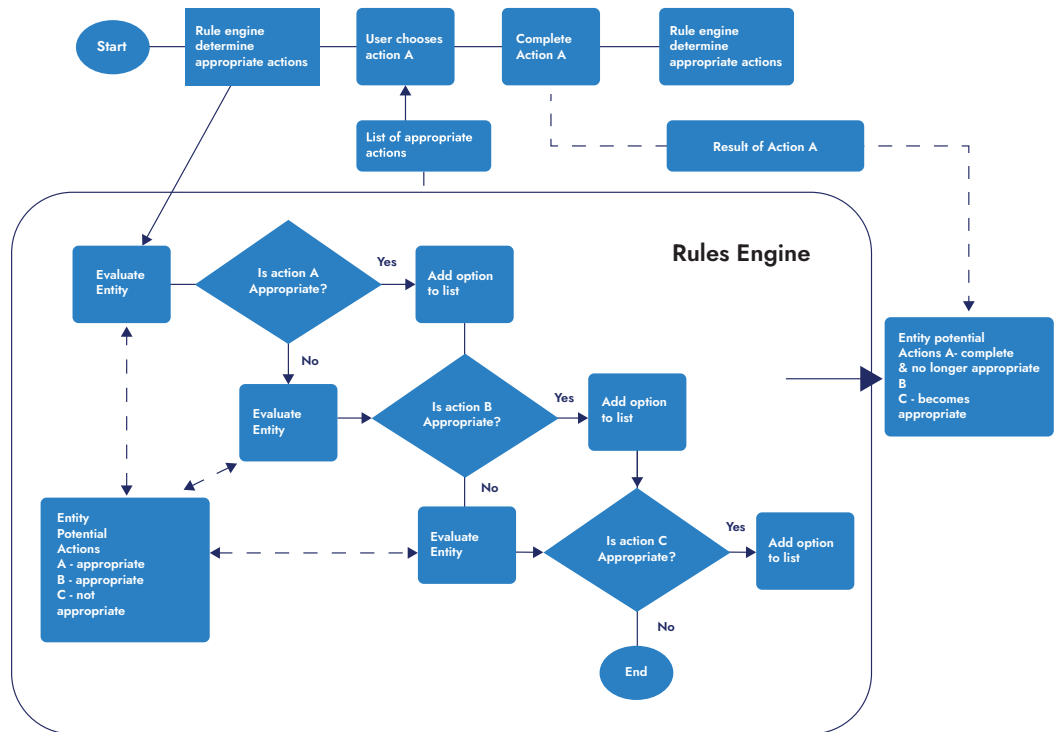


Because the rules evaluate the state of the entities in a rule-based system, it is possible for external systems to integrate with and take action on entities. This is unlike a flow-based system where external systems can only move the flow along or wake it up to move along the flow path. In a rules-based system, external systems can interact with entities, add new data, and change available options and activities.

Rules essentially add a second layer of logic to entities to make them much more dynamic.

# How Rules-Based Workflows Work in a Doctor-Patient Interaction

Consider a patient filling out basic interview forms on a website to schedule an appointment. It is routed to someone to schedule the appointment which once completed changes the state of the interaction from 'form submitted' to 'appointment scheduled'. This data is then stored in the patient's medical record, which would be the entity that stores the state data. So far in this example, we see a standard procedural workflow.  However, after arriving for the appointment the process of the Doctor/Patient interaction is not a good candidate for a procedural flow-based workflow. Once the patient is in the office, each interaction, or activity, may influence the next available interactions.

At the point the doctor walks into the room, there is already data in the patient record to be evaluated. Age, gender, and medications have all been captured in the procedural part. There are a few interactions or activities that are already emerged as appropriate:

- Take blood pressure
- Weigh patient
- Get the reason for the visit

These activities are all available, but the doctor could choose to do them in any order. In a traditional workflow, the doctor would be required to do them in the set order that the workflow dictated.

There are also many additional activities that are possible, but the doctor isn't going to engage in those potential activities until there is additional data that enable the doctor to use more logic or rules to determine the applicability of these possible actions.
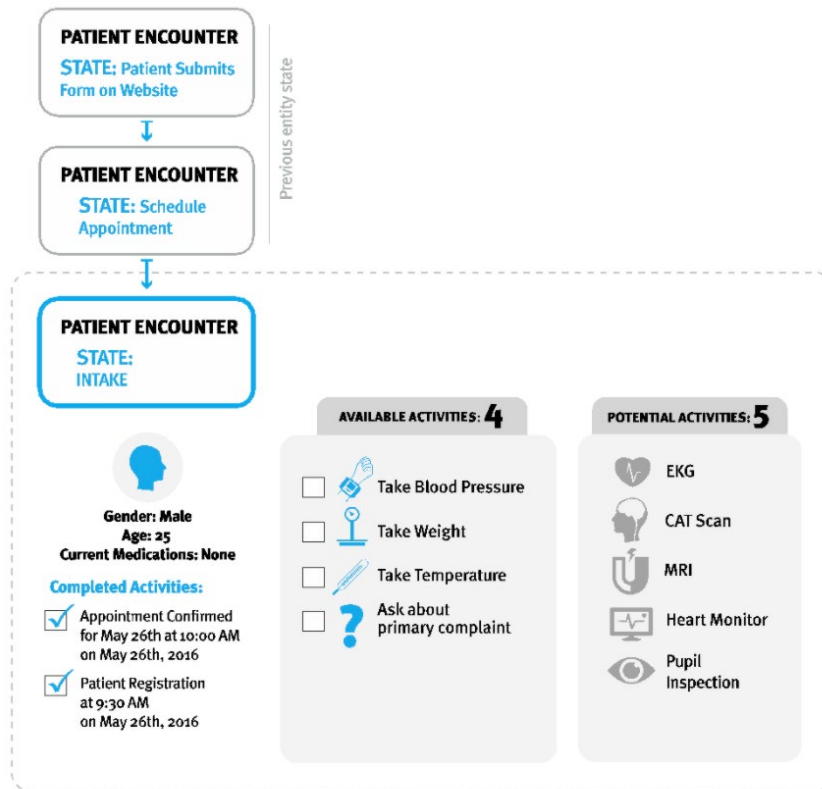
When the doctor starts to perform the available activities we can introduce some additional data into the entity. The patient informs the doctor they made the appointment because they cannot see out of one eye. This additional data would cause additional activities to become available and appropriate while other activities may now be excluded. Additional activities might include:

- Doing an inspection of pupil reaction
- Ordering a CT or MRI scan
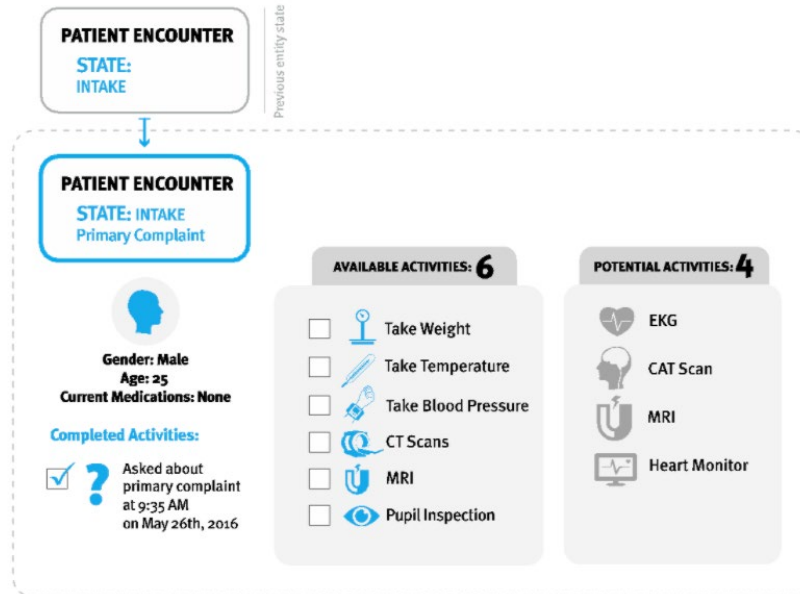


After executing the activity of 'checking the blood pressure', the doctor finds it to be elevated above a healthy level. With this additional data, the rules engine can simultaneously remove 'take blood pressure' as an appropriate action but add the action of 'hook up EKG machine' to look at heart function.

The outcome of activities may put entities in a state that spawns additional activities. There may be one or more activities that become relevant in the process that allows other people to participate as well. Lastly, several activities might put the entity into a final state or cause all activities to be eliminated. For example, the patient may cancel the visit, or the physician may order a blood sample completing this flow.
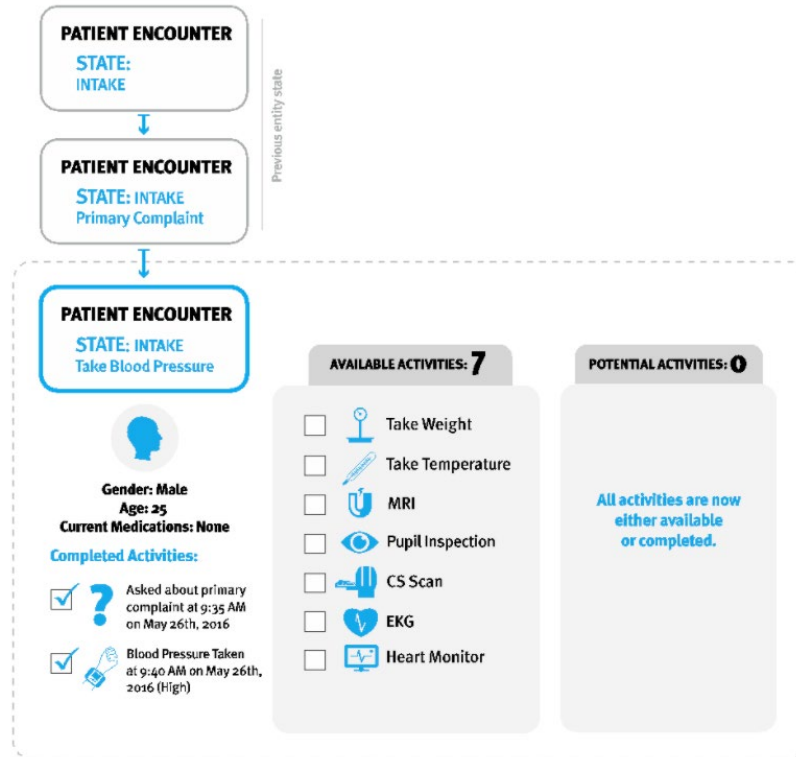


# Hybrid Workflows

While flow-based workflow and rule-based workflow are quite different, their strengths complement each other. Using them together can produce a very powerful solution.

Many real-life processes are a combination of rule-driven and flow-centric workflows. This could be any typical workflow process including a classic approval process, document creation and execution workflow in a contract management setting, or a ticketing type workflow for resolving problems. In these classic workflows, you have states and data that are very linear and a process that is well defined.  As an example below is a typical six-step process:

1  The request is submitted by a user

2  The request is validated by rules

3  The request is submitted to a manager for approval

4  If approved the request is routed for fulfillment

5  The fulfiller finishes the process the main flow

6  The requesting user is notified

Modern rule-based workflow engines are designed to leverage both rule-based and flow-based approaches in a single workflow. When you create a flow centric workflow like the one above, a modern platform will allow you to define an entity to store and manage the data that drives the process. This type of platform will also support a rule entity that has more than simple data elements and can embed a rule-based workflow into a traditional flow-based workflow. This dynamic entity can have relationships and additional actions/activities that are not directly related to the main flow.

In this example, if the flow is submitted to the manager for approval and is in that third step you could enable several other contextually relevant activities:

**A**   **Add additional detail** - allowing the requester to make a more compelling case to the manager.

**B**   **Request Estimate** - allowing the manager to send RFPs out to get bids so he/she can estimate the cost before approving. (This could be its own multi-state, procedural workflow)

'A' is valid in step 3 for the requester, and 'B' is valid in step 3 for the manager. These two  actions help give the workflow more utility and more meaning, but they don't move the workflow from step 3 to step 4 or even back to step 2.  These are the kind of available actions that are normally found only in rule-driven workflows, but modern workflow engines can imbed these dynamic flows in flow-centric workflow so both strategies can work together for the greatest flexibility and efficiency.

# Building Workflows in a Modern Rules-Based Workflow Engine

The approach of building a rules-based workflow and a traditional workflow is quite different. In a traditional workflow, the first step is conceptualizing the problem based on an overall flow. With a rules-based workflow, the first step is defining the entity structure.

Once an entity is defined, behavior can be associated with it. Rules-based workflow engines will provide mechanisms for creating actions and associating them with the entity. In production, the rules engine evaluates the state of the entity to determine what actions are appropriate. For example, an "Employee" entity could be created, and its actions menu could contain an action like "Get Employment History" that would act on that entity.

By defining entities and behaviors you do not need to create a spiderweb of complex routing connecting each step of the process to the next sequential step. The true benefit of a workflow tool is not lost because you keep the granular interactions with the system but can add intelligence, rules, and guidance.

**INTERNAL AND EXTERNAL ENTITIES - THE KEY TO ROBUST INTEGRATION**

Rules-based workflow engines support two types of entities, internally defined entities, and external entities. These two types of entities are vital to robust integrations.

The data elements in defined entities can not only include simple types like text and numbers but also other complex structures - allowing domain data to be modeled. Once the entity is configured, a special project is created where the rules platforms allow you to define actions and other behavior for your entity. Internal entities are stored in the database of the workflow platform.

In the same way that internal entities are defined, entities can also be read in from external systems.  In addition to designing the data structures, adding actions, and handling events, you also have to design flows for fetching these external entities. These flows can use any integration technology including direct database and web service calls.

The ability of external entities to enable systems outside of the workflow engine to interact with the internal entities within the rules engines can be quite powerful. In traditional flow-based platforms, this is not supported.

# Insurance Policy Claim Review - Working With an External System

An Insurance policy claim review process is a good example of how external systems can interact with a rules-based workflow.

In this example, data exists in an external database representing claims that need to be reviewed.

Reviewing these claims requires a combination of people working in external systems throughout the company.  Potential activities might include:

- Verify claim with the policy owner
- Confirm repair estimate
- Underwrite adjusted insurance premium

In a case like this, you would need rules to figure out what activities are needed, who should do them, and when. If some activities can be done simultaneously while others are exclusive and can only be done when another activity is not in process, rules are required to manage this.

For rules to orchestrate these interactions you would need your workflow process to monitor the activities performed by the external systems to capture the result of the process. This is where an external entity comes in. It can monitor activity and a workflow would be built to pull the data back into the rules engine. Using this approach, if the user's interaction with the external system does not change the state of the external entity sufficiently to satisfy the rules, that activity will simply remain available or eventually get escalated.

When all activities on an external entity have been satisfied, the item is 'complete' and drops from the list of entities that need to be worked on in the workflow.

In this case, external entities systems are interacting with the entity in the rules engine and can influence the state of the entity and dictate whether it still needs attention.
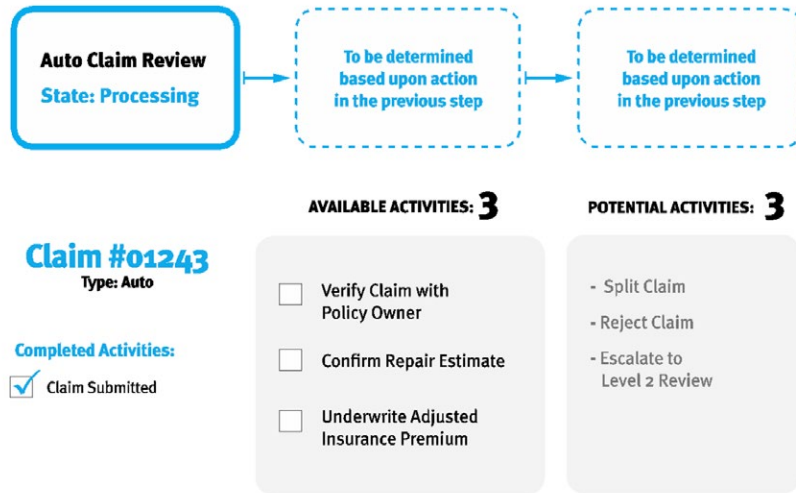


## Advanced Behaviors

While the elements in a rule-centric workflow platform revolve around the data and available actions, in a modern rules-based workflow platform additional behaviors can be configured to enhance capabilities, including validation and events.

Validation rules can be configured using the rule engine and enforced when entities are edited or saved. This eliminates inappropriate data from being saved to an entity.

If you want to execute processing around stages in an entity's lifecycle, you can hook the before/after save/delete events. This would enable you to trigger new workflows, adjust data storage, send notifications, or anything else that the flow engine is capable of doing once these events occur. These event hooks function as workflows that take in the entity data as input.

## Medical Study Documentation Gathering - Hybrid and Events

By leveraging advanced behaviors with hybrid workflows, much more advanced functionality can be achieved without excess complexity. Below is an example of how the capabilities of modern workflow platforms can automate involved processes without excess complexity.

For every patient involved in a medical study, before, during, and after the study, documentation must be gathered, checked, and filed. When certain documentation, contracts, and other structured information is gathered, the specific patient study is progressed through stages. The gathering of each of the documentation pieces is done with a small sequential workflow that uses traditional tasks and assignments to get the information both via forms and by requesting data be sent to monitored email boxes.

The patient study cannot progress to the next stage until the required information is gathered. If patients do not progress through the study in the appropriate time frames, SLAs could be breached. To manage this, SLA based triggers monitor and report on studies that are past their required windows for activity. Every time a new piece of information is added to the entity, the stage can be evaluated and an SLA can be calculated.

## Importance of Reporting and Progress Management

In a rule-driven workflow, you do not have a true and concrete overall process so you can't gauge how far you are from the finish line. Therefore progress management and visibility become an important topic.

Even though there is not a flow to manage it, the entity can transition between multiple macrostates. To take the HR sample used earlier, a person might be a 'Prospect', 'Employee', or 'Former Employee'. These states not only influence the activities that are available but could be viewed differently on a dashboard. A reporting engine is required to compute different views to get an accurate picture of the process. Entities in different states may also need to be handled differently from a security perspective, monitoring the state of entities and their access credentials is another important component of building dynamic workflows.

## Final Thoughts

Rules-based workflows are quite different from traditional flow-based workflows. Understanding the differences and how they work together is key to an optimized and dynamic process. Rules-based workflow enables businesses and employees to be more reactive to customers and stakeholders by more effectively addressing edge cases. Delivering on edge cases is where successful businesses differentiate themselves. To learn more about Decisions rule-based business process management platform, visit decisions.com.

**DECISIONS**