Automating Unpredictable Processes:

Building Responsive Apps using Business Rules

By **Carl Hewitt**, *Chief Architect*, *Decisions*
and **Heath Oderman**, *CTO*, *Decisions*

DECISIONS

INTRODUCTION

The term 'Workflow' is not always clearly understood. It can refer to traditional business processes (flow centric, procedural or sequential workflow) where steps are taken to coordinate interactions between people and systems. Workflow and BPM platforms are not as good at configuring processes in which actions or activities can be done in any order - where these actions are not based on a sequence, but are based on and constrained by business rules. Workflow that does not progress in a serial or completely predictable fashion might require users to select and coordinate activities. These kinds of business needs often require the coordination of dependent workflows which make them difficult to architect, understand and change.

There are very few great tools that allow users to build configurable rule driven workflows. Popular applications like SalesForce allow some configuration of 'non directed' workflows - but lack the configuration and intelligence that workflow tools provide not to mention that the resulting functionality requires a great deal of human-powered rule enforcement. Configuration of these 'extensions' is often very technical and constrained.

Before we get into explaining rule driven workflows I want to be clear that they are not better than flow centric workflow, each is useful in solving a certain problem. Flow centric workflows are appropriate for many classes of problems and can be really flexible:

- Flow centric workflows can have complex routing
- Flow centric workflows can contain tasks that produce additional tasks/routing based on data gathered in the workflow. (ie, defining an approval list then going through each approver in the list)
- Flow centric workflows can use rules to determine what actions need to take place
- Flow centric workflows can use data to determine how many and what types of tasks are taking place
- Flow centric workflows can loop back to redo activities, can branch and merge for parallel activities, can even have functionality to 'skip' backwards and forwards

However... the idea of the order of activities or the order of actions is always present. The workflow engine is determining what happens next.

The primary advantage of flow centric processes is that they are easy to understand and construct, because the steps are defined so concretely. While flow centric workflows fit many problem domains easily, there are classes of problems that get constrained and are unnatural in a procedural model.

Take a sales cycle as an example. While a lead may follow a macro path (ie, contact -> prospect -> lead -> customer) the activities that are possible and the participants are different for each 'work-flow'. The macro path is more 'state' transitions for understanding and reporting than something that is determined by a workflow engine moving the pointer as to where you are in the process.

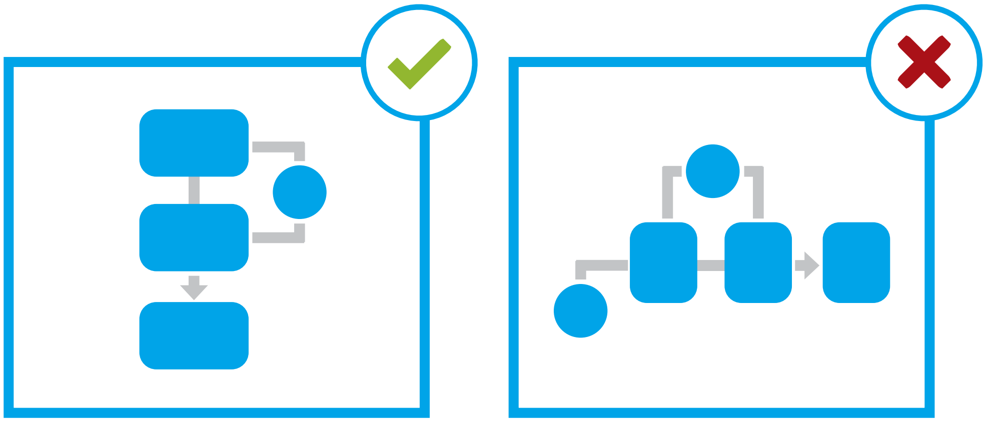
Given the more dynamic nature of these sorts of interactions - they have traditionally been relegated to non workflow software (CRM, ERP or Case Management software) or non structured systems (email, sticky notes, trello, evernote).

While the words 'process', 'flow', 'workflow' are generally understood in a BPMS to be a sequential or procedural workflow, there is not a generally accepted term to help differentiate a sequential and a non sequential workflow. Here are some alternatives that are descriptive:

- Flow Centric, Procedural, Sequential, Process First, Procedural, Directed, Flow Based, Sequence Ordered
- Rule Driven, Event Driven, Non-Sequential, Entity Centric, Data Oriented, Dynamic, Non-Procedural

We will call:

- Traditional workflows - 'Flow Centric'
- Dynamic workflows that are not sequential or procedural - 'Rule Driven"



Rule Driven Workflow Overview

A rule driven workflow is driven by the user - not the engine. Users determine which activities to work on. The engine provides constraints as to which activities are presented and/or required, but the interacting entity selects which event to work next. Contrast that with a flow centric workflow where the sequence of work is fixed, and the interacting system or user is not presented options for what to work on next.

Rule driven workflow is focused on the history of events and other entity state data that is defined in the workflow engine or that exists in another system and the activities that are appropriate based on the current state or other data of that entity.

Events can be user interactions, system interactions or based on monitoring external data sources. These influence the state of the data - giving the rules more information to determine what activities are available. Rule driven workflow still allows for the definition of activities - with all the bells and whistles including forms, tasks, deadlines, and notifications - it's just that the order of them is not forced by the engine.

If the entity is in x state, I can do the following actions: a, b, c. These actions might change the state of the entity in such a way that some of the activities are no longer appropriate and other activities might now be appropriate. The state of the entity may change in such a way that the entity may be viewed differently from a tracking standpoint as well. Reporting plays a big role in these rule driven workflows because unlike procedural workflow where the 'completeness' of the work is clear by where you are in the process, here is it based on the current data in the entities that are being 'worked on'

The activities of a non sequential workflow are defined in the same manner that you would define a procedural workflow (they are small workflows that can have forms, steps to manipulate data, interact with outside systems, etc).

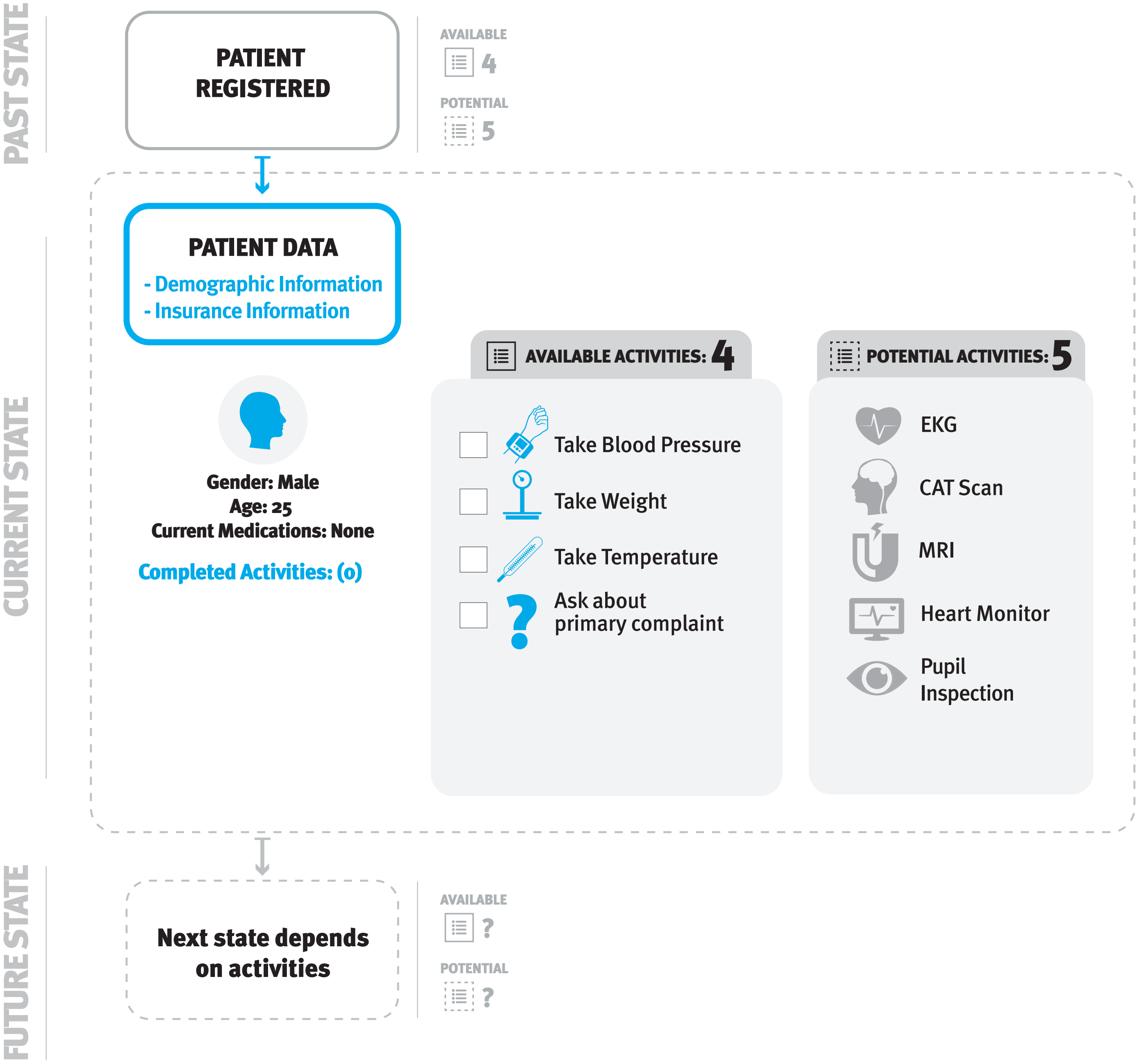
Example: Doctor Patient Interaction

Consider a patient filling out basic interview forms on website to schedule an appointment. It is routed to someone to schedule and the data [state] is recorded in a patient medical record [entity]. 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 workflow as each interaction [activity] may influence the next available interactions, and many of the interactions could result in the end of the appointment.

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

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

There are also many interactions [activities] that are possible, but the doctor isn't going to engage in those activities until there is additional data [state] that allow the doctor to use more logic [rules] to realize the applicability of these possible interactions.



PAST STATE

PATIENT REGISTERED

AVAILABLE 4

POTENTIAL 5

CURRENT STATE

PATIENT DATA

- Demographic Information
- Insurance Information

Gender: Male
Age: 25
Current Medications: None

Completed Activities: (0)

AVAILABLE ACTIVITIES: 4

□ Take Blood Pressure

□ Take Weight

□ Take Temperature

□ Ask about primary complaint

POTENTIAL ACTIVITIES: 5

EKG

CAT Scan

MRI

Heart Monitor

Pupil Inspection

FUTURE STATE

Next state depends on activities

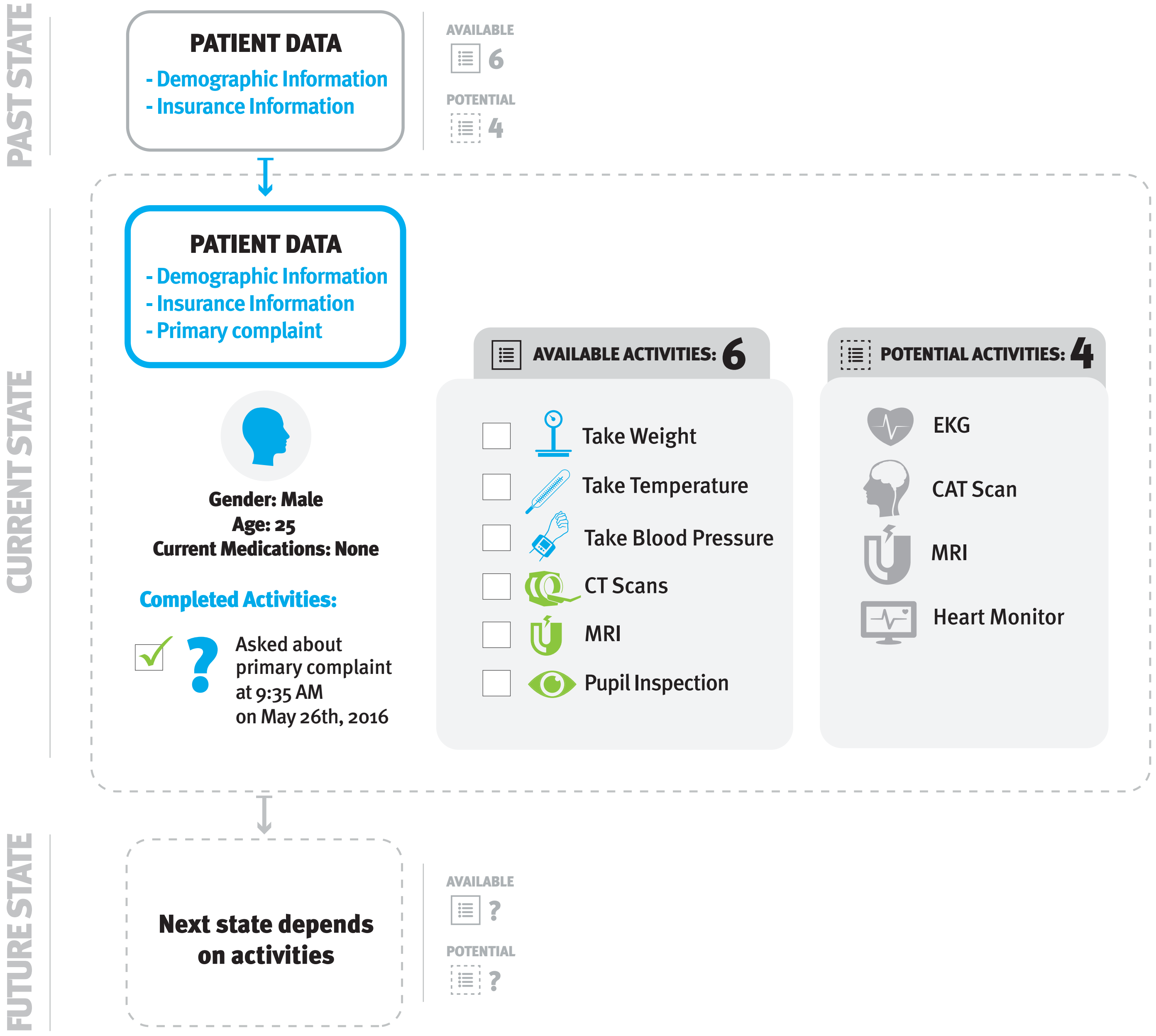
AVAILABLE ?

POTENTIAL ?

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

When the doctor starts to use the available interactions we can introduce some additional state (data) into the entity. The patient informs the doctor they are here because they cannot see out of one eye. This additional data would cause additional activities to now be available and appropriate while other activities may now be excluded.

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



PAST STATE

PATIENT DATA
- Demographic Information
- Insurance Information

AVAILABLE 6

POTENTIAL 4

CURRENT STATE

PATIENT DATA
- Demographic Information
- Insurance Information
- Primary complaint

Gender: Male
Age: 25
Current Medications: None

Completed Activities:

☑ Asked about primary complaint at 9:35 AM on May 26th, 2016

AVAILABLE ACTIVITIES: 6

☐ Take Weight
☐ Take Temperature
☐ Take Blood Pressure
☐ CT Scans
☐ MRI
☐ Pupil Inspection

POTENTIAL ACTIVITIES: 4

EKG
CAT Scan
MRI
Heart Monitor

FUTURE STATE

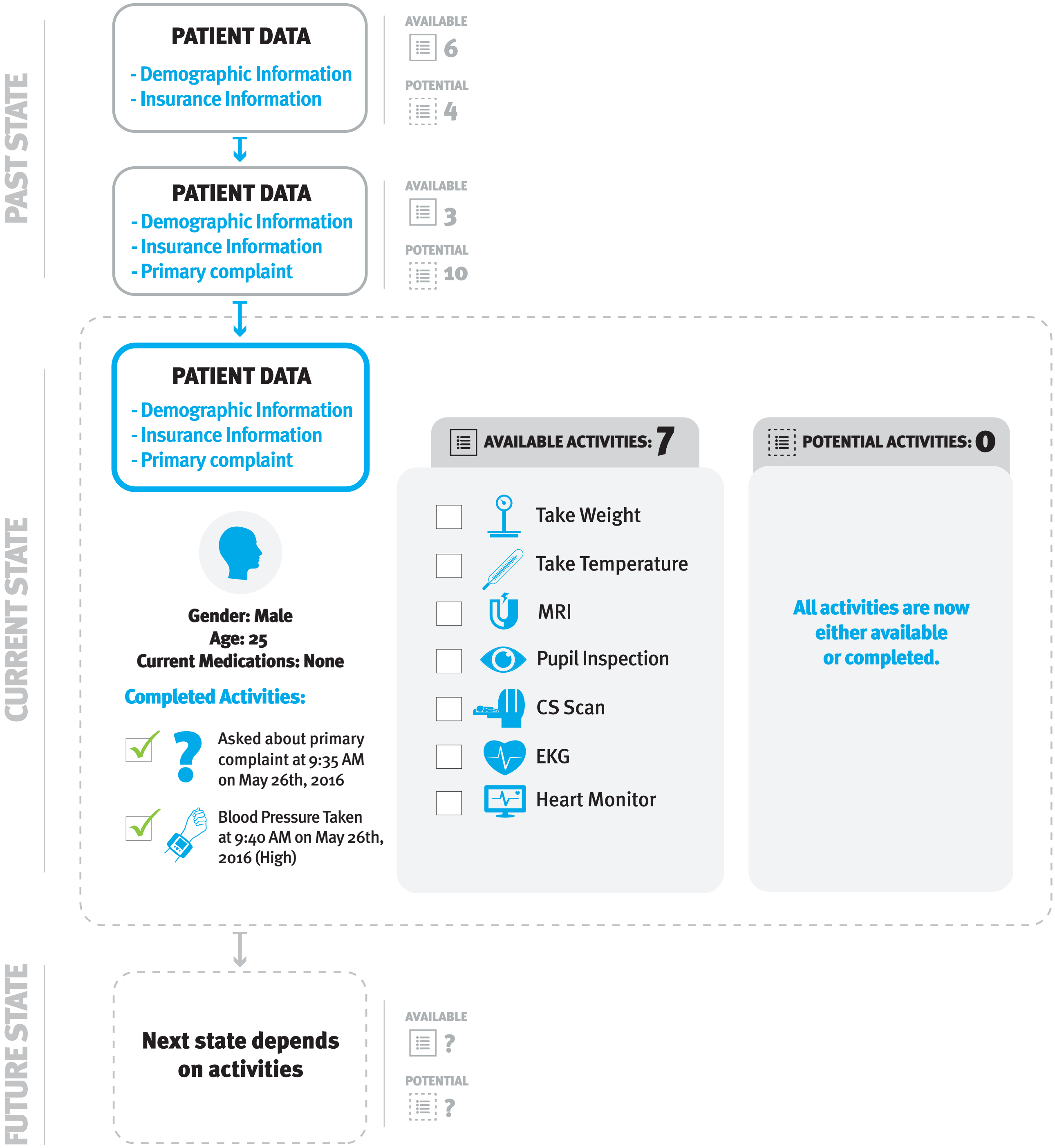
Next state depends on activities

AVAILABLE ?

POTENTIAL ?

The Doctor checks the blood pressure and finds it to be very high, this additional data both removes ‘take blood pressure’ as an appropriate action but adds the action of ‘hook up EKG machine’ to look at heart function.

There may be one or more activities that become relevant in the process that allow other people to participate as well. The patient may cancel the visit, or physician may order a blood sample. There might also be several activities that put the entity into a final state or cause all activities to be eliminated.



PAST STATE

PATIENT DATA
- Demographic Information
- Insurance Information

AVAILABLE 6
POTENTIAL 4

PATIENT DATA
- Demographic Information
- Insurance Information
- Primary complaint

AVAILABLE 3
POTENTIAL 10

CURRENT STATE

PATIENT DATA
- Demographic Information
- Insurance Information
- Primary complaint

Gender: Male
Age: 25
Current Medications: None

Completed Activities:

☑ Asked about primary complaint at 9:35 AM on May 26th, 2016

☑ Blood Pressure Taken at 9:40 AM on May 26th, 2016 (High)

AVAILABLE ACTIVITIES: 7

□ Take Weight
□ Take Temperature
□ MRI
□ Pupil Inspection
□ CS Scan
□ EKG
□ Heart Monitor

POTENTIAL ACTIVITIES: 0

All activities are now either available or completed.

FUTURE STATE

Next state depends on activities

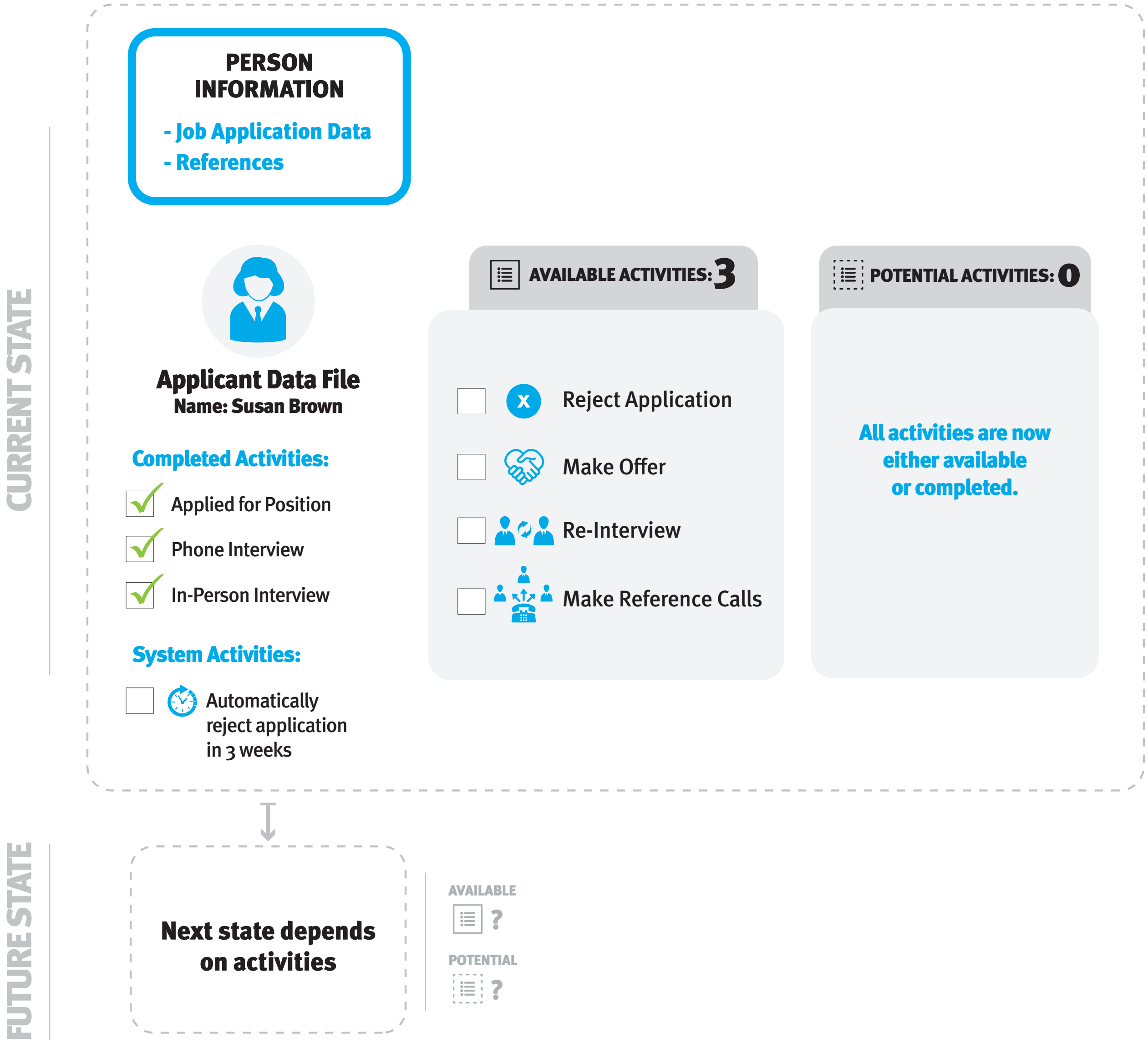
AVAILABLE ?
POTENTIAL ?

Examples: HR Employee File

When an employee is hired, their basic information is stored in a database providing a view into the employee's position history, performance reviews, documentation, and other records.

Based on the data [state] of the data in the employee's file [entity] different activities are appropriate. If the employee has not yet been hired, 'make offer' action [activity] would be presented.

A procedural workflow like 'start complaint investigation' could also be an activity. Because this is a sequential and controlled process, it would be better structured as a normal workflow with a series of steps that are taken, assignments with deadlines, clear notifications, full auditing, etc.



CURRENT STATE

PERSON INFORMATION

- Job Application Data
- References

Applicant Data File
Name: Susan Brown

Completed Activities:

☑ Applied for Position
☑ Phone Interview
☑ In-Person Interview

System Activities:

□ Automatically reject application in 3 weeks

AVAILABLE ACTIVITIES: 3

□ Reject Application
□ Make Offer
□ Re-Interview
□ Make Reference Calls

POTENTIAL ACTIVITIES: 0

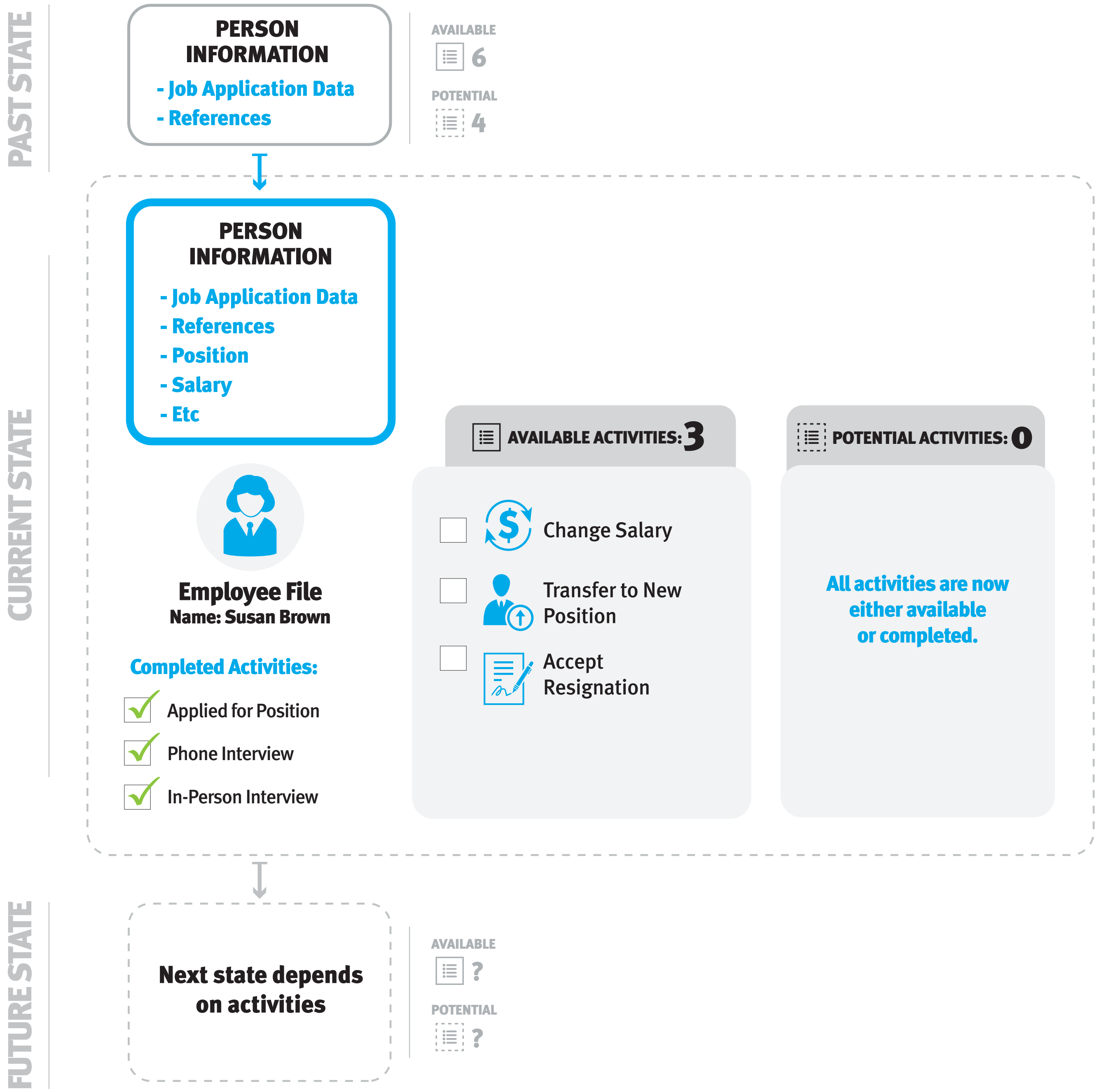
All activities are now either available or completed.

FUTURE STATE

Next state depends on activities

AVAILABLE ?
POTENTIAL ?

If they are a current employee, simple activities like ‘change salary’, ‘transfer to new position’ and ‘accept resignation’ would be present.



PAST STATE

PERSON INFORMATION

- Job Application Data
- References

AVAILABLE 6

POTENTIAL 4

CURRENT STATE

PERSON INFORMATION

- Job Application Data
- References
- Position
- Salary
- Etc

Employee File
Name: Susan Brown

Completed Activities:

☑ Applied for Position

☑ Phone Interview

☑ In-Person Interview

AVAILABLE ACTIVITIES: 3

□ Change Salary

□ Transfer to New Position

□ Accept Resignation

POTENTIAL ACTIVITIES: 0

All activities are now either available or completed.

FUTURE STATE

Next state depends on activities

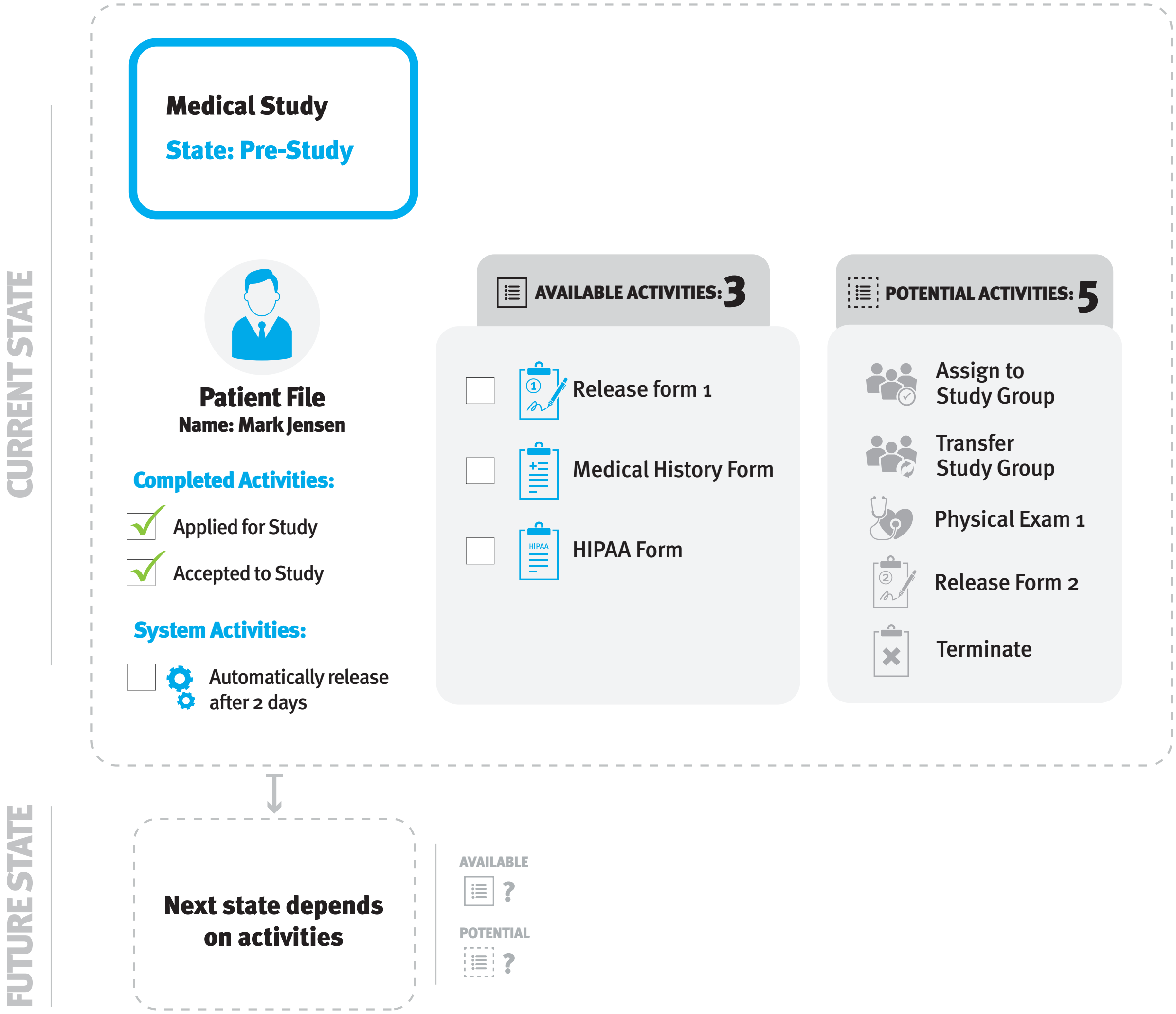
AVAILABLE ?

POTENTIAL ?

Example: Medical Study Documentation Gathering

For every patient involved in a medical study, before, during and after the study documentation must be gathered, checked and filed. When certain documentations, contracts and other structured information is gathered, the specific patient study is progressed through stages. The patient study cannot progress to the next stage until the required information is gathered. SLA based triggers monitor and report on studies that are past their required windows for activity.

The actions that are available are based on the current data (specifically the missing data), any anomalies the rule engine discovers and the stage of the study. The stage is evaluated on every new piece of information being added so a new stage can be assigned and SLAs calculated. 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 using forms and requesting to be sent into monitored email boxes.



CURRENT STATE

Medical Study

State: Pre-Study

Patient File

Name: Mark Jensen

Completed Activities:

☑ Applied for Study

☑ Accepted to Study

System Activities:

□ Automatically release after 2 days

AVAILABLE ACTIVITIES: 3

□ Release form 1

□ Medical History Form

□ HIPAA Form

POTENTIAL ACTIVITIES: 5

Assign to Study Group

Transfer Study Group

Physical Exam 1

Release Form 2

Terminate

FUTURE STATE

Next state depends on activities

AVAILABLE ?

POTENTIAL ?

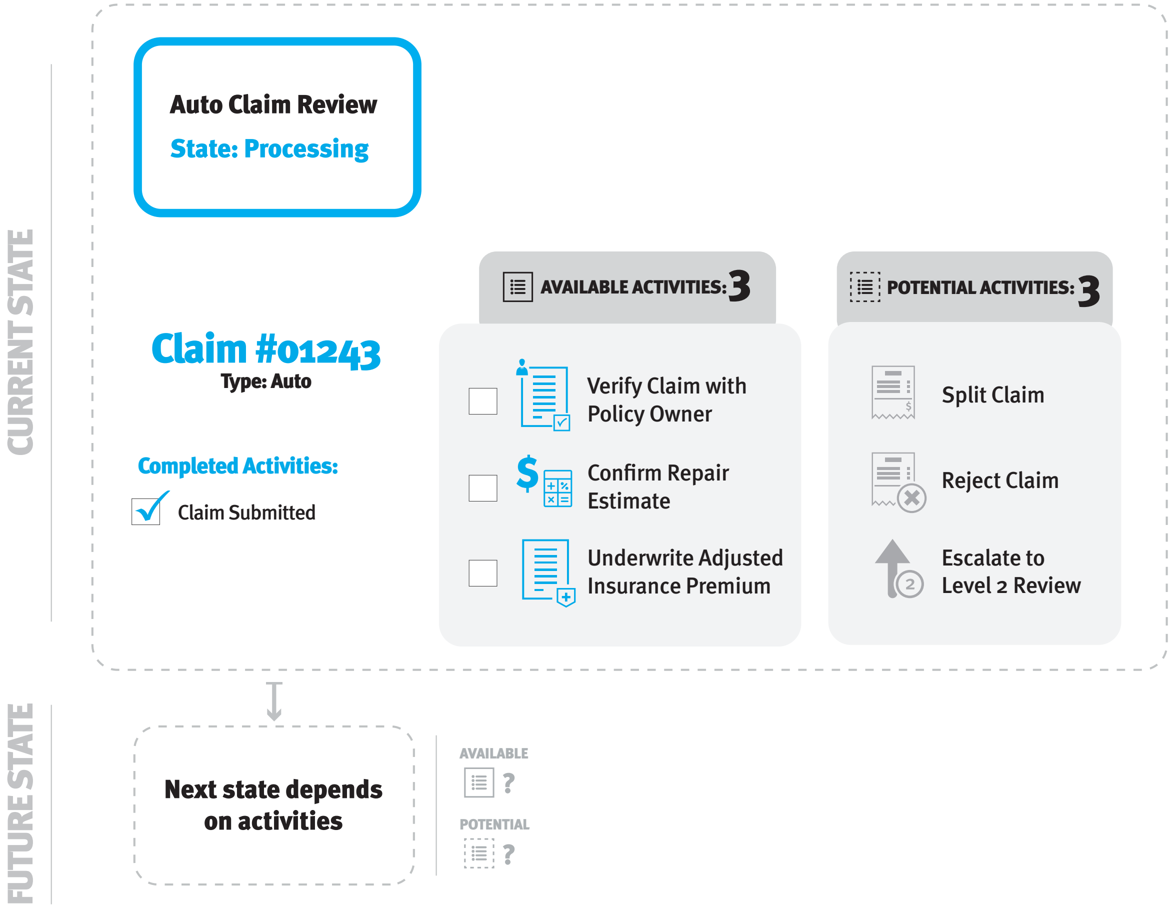
Example: Insurance Policy Claim Review

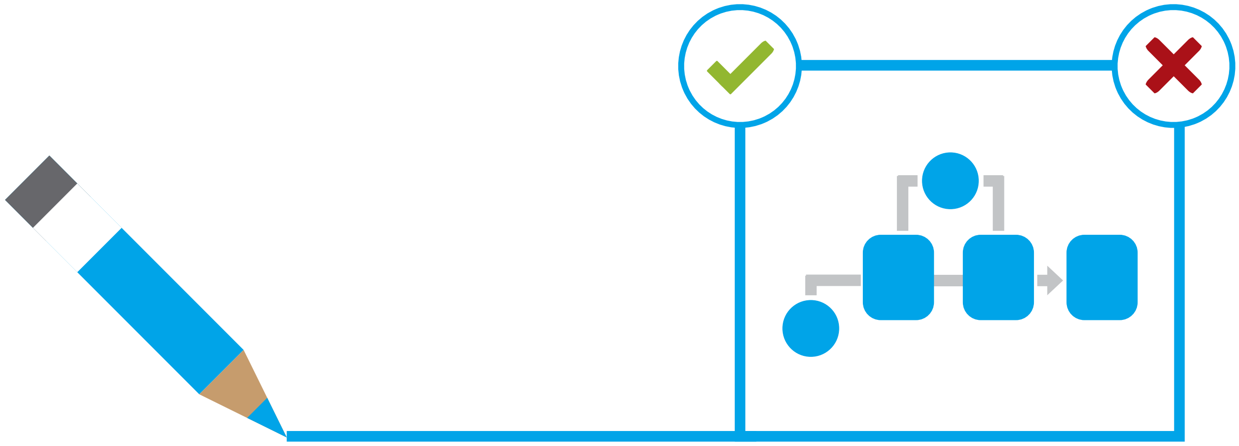
Data exists in an external database representing claims that need to be reviewed. The data is fed into this system through a variety of sources: CSV, Excel Files, Manual Data Entry, etc. A view (report) of this data is shown as a task list to be worked on.

Reviewing these claims requires a combination of people working in external systems throughout the company. In a case like this you would need rules to figure out what interactions are needed, and who should do them. Some activities can be done at the same time while others are exclusive and can only be done when another activity is not taking place.

The activities described above are imagined to happen in an existing software system, so for these interactions you would need your workflow process to monitor for the result and then pull the data back into the entity. If the user's interaction with the external system does not satisfy the rules, that activity will simply remain available or eventually get escalated.

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





Designing a Rule Driven Workflow

Current software to solve this class of problem generally takes one of two strategies:

- Completely unstructured data that rely on things like checklists and experts to interact with the state of the object
- Very data intensive screens where the expert can change the state of the entity without any real guidance

Trying to solve this problem using a procedural workflow would involve building complex internal routing leading to a spiderweb of possible paths in a workflow diagram. The benefit of a true workflow tool is lost because you do not have simple interactions with the system being able to add intelligence, rules and guidance.

Decisions fully supports a rule driven workflow model using the same graphical designers that are used in normal workflow.

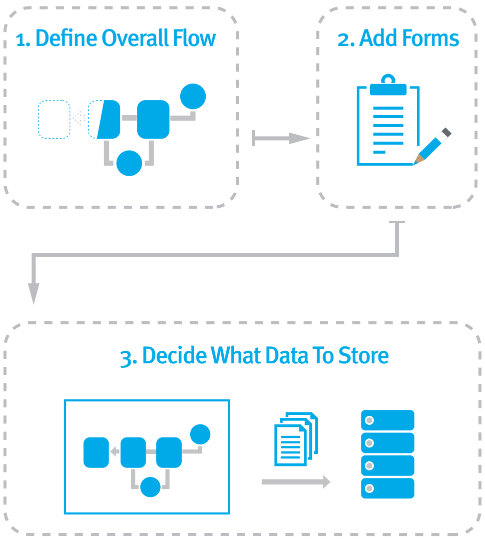
The difference is in how the problem is approached. Instead of conceptualizing the problem based on an overall flow, the initial activity is around defining the entities structure. This structure can be stored in the decisions database or may already exist in external data stores.

Once the entity is defined, behavior can be associated with it. Actions are built using flow and form engines, and rules evaluate the state of the entity to determine which actions are appropriate.

FLOW CENTRIC

RULE DRIVEN

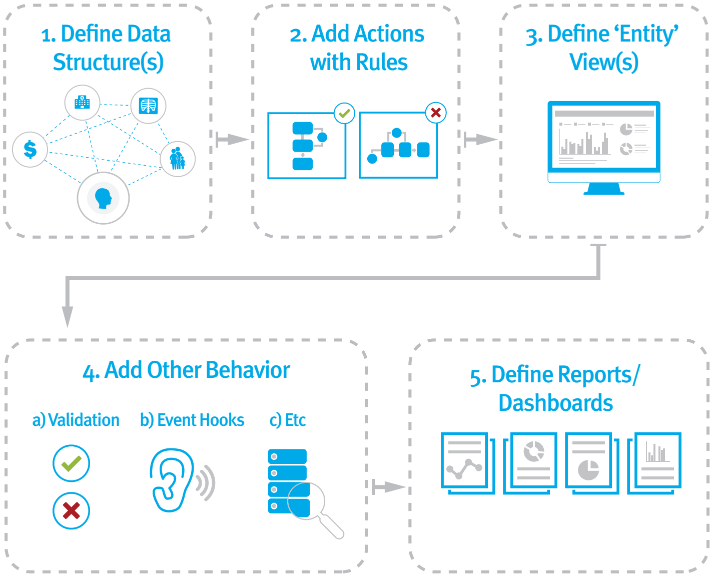
STEPS TO CREATE.



1. Define Overall Flow

2. Add Forms

3. Decide What Data To Store



1. Define Data Structure(s)

2. Add Actions with Rules

3. Define 'Entity' View(s)

4. Add Other Behavior

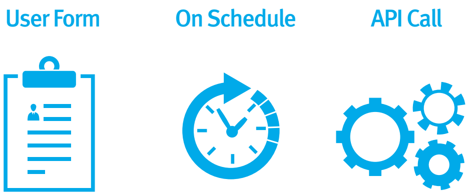
a) Validation

b) Event Hooks

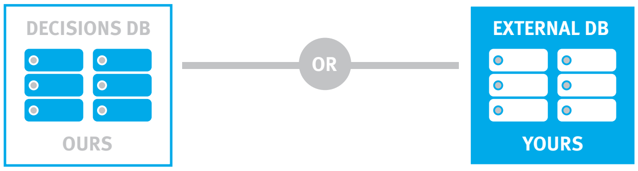
c) Etc

5. Define Reports/ Dashboards

HOW THEY START.



Entity data must exist - either in Decisions database or in an external database.



Flow centric apps can create this data so scheduled jobs, user initiated and Api initiated are all possible ways to start rule driven apps.

INTERACTIONS.

There are a lot of options, but most commonly the interactions will be through assignments and tasks and a process tracking page.

While you can have tasks associated with a non-sequential workflow, the primary interaction is through the entity itself - whether a page or reports against the entities.

Other Behaviors

While the ‘workflow’ elements in an rule centric workflow pattern revolve around the data and the actions/activities that are available. Additional behaviors can be configured.

Entity View Page

You can design, using the page or form designers, specific user interfaces that let you view and/or edit the entity data. These user interfaces will often also host the ‘actions’ a user can take on this entity.

Events

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

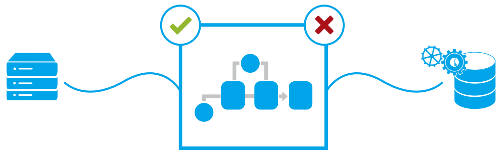
Validation

Validation rules can be configured using the rule engine and enforced on edit and save of an entity.

Reporting and ‘Progress’ Management

In a rule driven workflow you do not have a true and concrete overall process that lets you relate where you are to where the finish line is - therefore ‘progress’ management and visibility becomes an important topic. Even though there is not a flow to manage it, the entity can transition between macro states that are interesting from a business perspective. To take the HR sample from above, a person might be ‘Prospect’, ‘Employee’ or ‘Former Employee’. These states not only influence the activities that are available but also would likely be viewed differently from a dashboard perspective and handled different from a security perspective.

Using the reporting engine, different views can be computed. If you wanted to evaluate the entity and add additional computed data elements, this can easily be done either in your action flows or based on events like ‘before save’.



Rule Driven Workflows and External System Interactions

In a flow based workflow, external system interactions - where external systems call into the workflow - are focused on moving the flow forward, or at least waking it up so the rule engine can make different choices. You are bound to the to the current location of the flow.

With an rule driven workflow, you are not having to interact with the flow but with the state (data) that makes up the entity. Interacting with the state of an entity is an event that causes the workflow to change and adjust its available actions and activities.

Types of Entities

Entities can either be stored in the Decisions Database or can be configured as 'External Entities'.

Defined Entities

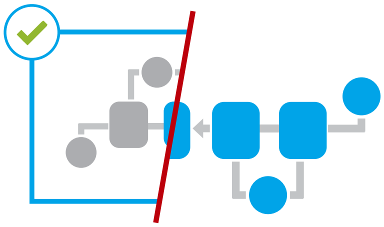
Entity structures can be defined graphically in the Designer Studio in Decisions. The data elements can not only include simple types like text and numbers but also other complex structures - allowing you to model your domain data.   Once you have configured your entity, a special designer project is created where you are allowed to define actions and other behavior for your entity.

When you define an entity in Decisions Studio table structures are created using normal table structures in the database.
Entities that can be designed either are:

- **Simple Entities:** These entities can have relationships to other entities as well as data members. These entities are searchable, can be used in reports and can handle actions
- **Container Entities:** Container entities are extensions of the Folder concept in the Decisions Platform. These entities can be configured to show in the tree, have a direct url to view them and can contain other entities (without having to define relationships). Because they are containers, they interact naturally with other services and concepts in the Decisions Platform like Documents, Assignments, Comments, Tags and many more

External Entities

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



Hybrid: Adding 'Rule Based Activities' to Flow Centric Workflow

Many real life processes are a combination of rule driven and flow centric workflow. In Decisions it's very easy to build a procedural workflow that governs the macro life cycle of the data. This could be any workflow process that you might normally think of: a classic approval process, a document creation and execution workflow (contract management), a ticketing type workflow for problems or change management. In these classic workflows you have states and data that are very linear and a well defined process. As an example:

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 (system or person) finishes the process
6. The requesting user is notified

When you create a flow centric workflow like the one above in Decisions, you can define an entity to store and manage the data for the process. This becomes additional detail that lives with the process and shows up in all reports. Since this is a real entity in Decisions it has more than simple data elements. It can have relationships and additional actions/activities. Using this, there can be activities and actions that are not directly related to the main flow.

If the flow is submitted to the manager to approval and is in that third step you could enable several other contextually relevant activities:

- **Add additional detail** - allowing the requester to make a more compelling case to the manager
- **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 Decisions can use these in concert with flow centric workflow for the most flexibility and efficiency.

Conclusion

Both flow centric and rule driven workflows are useful. Flow centric workflows are easier to conceptualize when first using a workflow platform like Decisions, but more complex business processes can lead to very complicated and large workflows. Entity based workflows can help solve this problem by allowing these actions/activities to be responsive to the state of the entity.

Even with an rule driven approach, flow centric workflow are still very useful providing natural ways to setup the entities initial state or launched as sequential handling of processes from the entity.

DECISIONS