TRANSACTIONAL BEHAVIOR

SCIANTA ANALYTICS

DEEP INSIGHT™

"The natural evolution of machine learning, Cognitive Computing attempts to imbue, in computer systems, the same insight and understanding we see in humans."

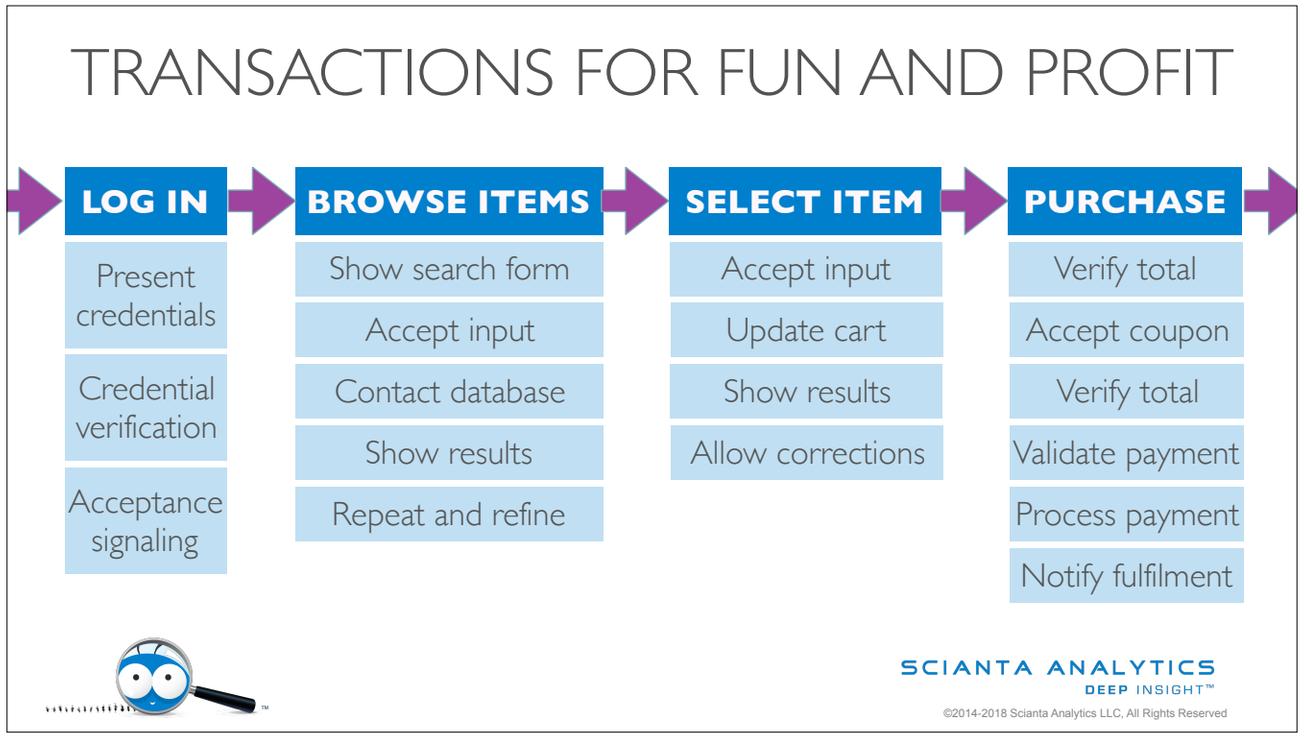
Earl Cox
Chief Scientist, Scianta Analytics
Splunk .Conf 2013

SCIANTA ANALYTICS
DEEP INSIGHT™

AGENDA

Introduction to Machine Intelligence	Data Handling 1	Data Handling 2	Anomaly Detection	Transactional Behavior	Impact Analysis
Academic Concepts	Collection	Retention	Anomaly Definition	Defining Transactions	Organizational Visibility
Data Systems	Storage	Format	Measuring Normality	Transaction Relationships	Types of Impact
Maturity Curve	Security	Labeling		Probability Measurement	Responsiveness

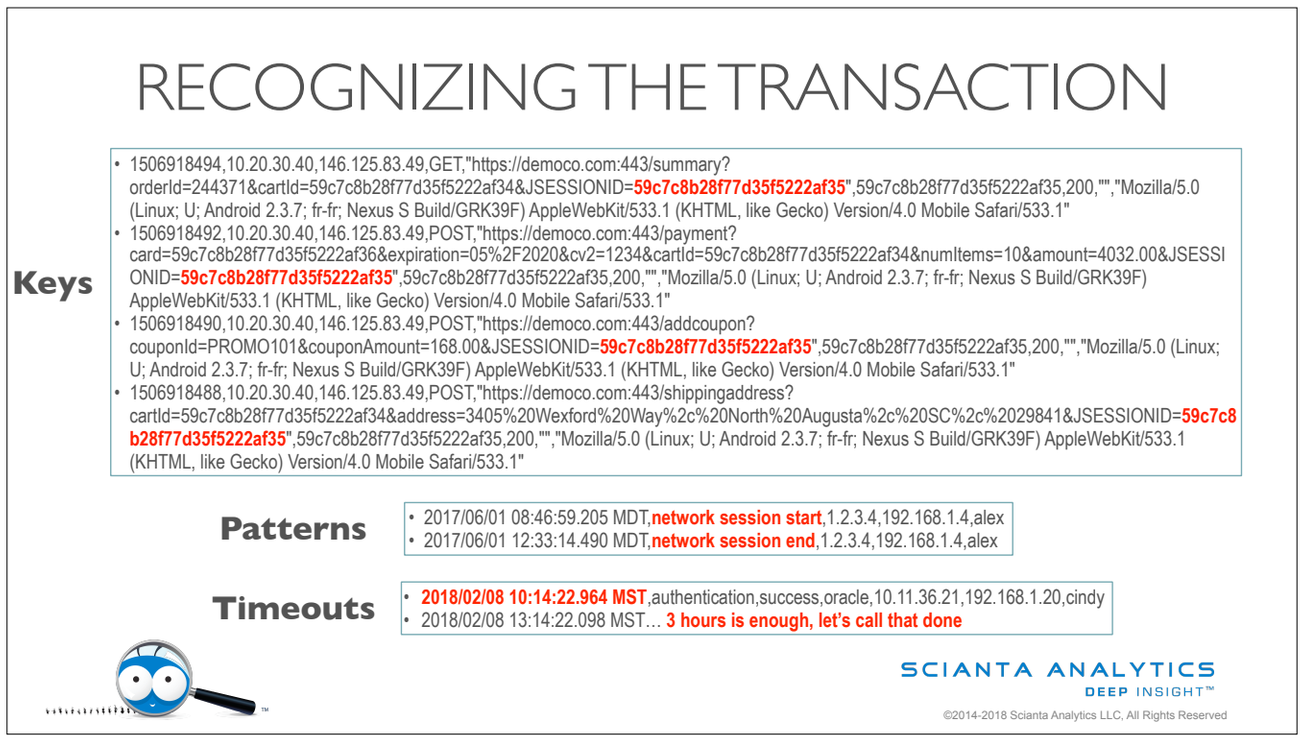
SCIANTA ANALYTICS
DEEP INSIGHT™



We've spent some time on understanding whether a single event is anomalous or not, which is pretty useful for understanding simple systems and simple events. As we start looking at more valuable events though, we often find that they're not simple; they're chained together in ways that make them into a new thing.

Transactions are a much more interesting view of the world because they allow us to look at an outcome instead of an isolated event. While any single problem in this chain is certainly a problem, the reason it's interesting is because it blocks the whole chain. If the system can bypass that problem and still support the transaction, then everyone is happy. So why not focus on the transaction instead of the event?

Specifically, we can look at each event in our data as an indication that an actor has reached a state, which implies movement from a previous state. Each of these movements is itself an event, which we can consider the normalcy of using the tools discussed in the last session.



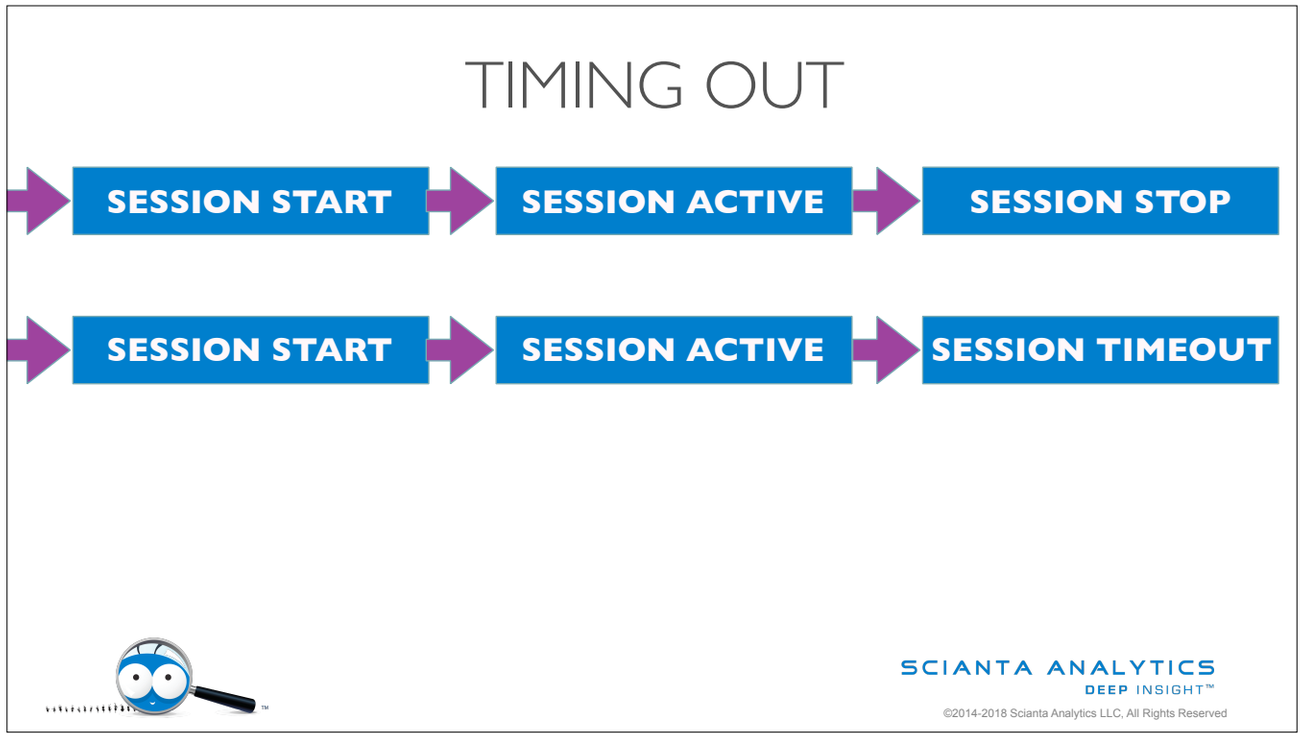
Using transactions is obviously valuable, but it can be challenging to determine where they are. Here are three ways to recognize transactions in typical event data.

The simplest mechanism is a transaction or session key. In a web app, the application developer is almost certainly using some sort of a key to keep track of the session. If you're lucky, they have logged that key so you can keep track of the session too. Unfortunately this doesn't help when the user's focus leaves that application, but it does give you a starting point.

Another simple mechanism is recognizable patterns that indicate session starts and stops. When there's a start for a user, we've got a session; until a stop comes, anything from that data source for that user is part of the same session.

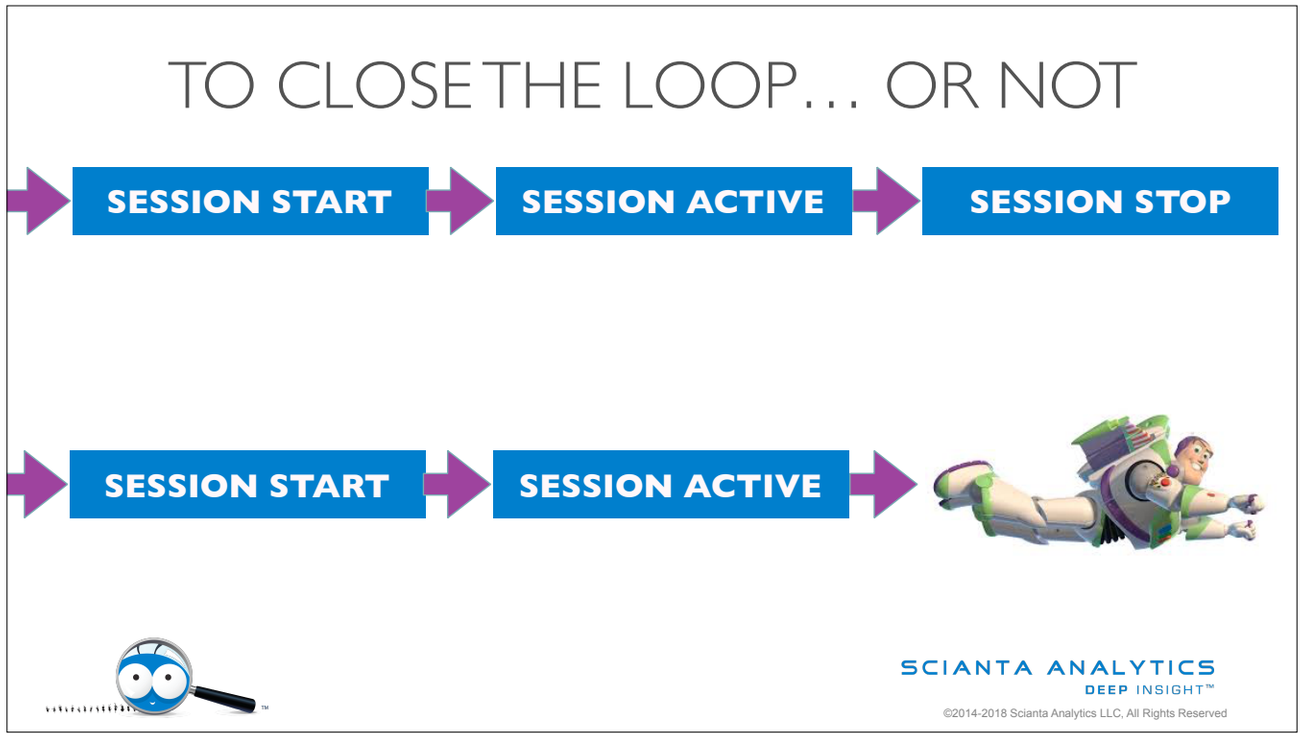
Unfortunately, no one can guarantee that there will be a session stop signal, so it's often useful to set a timer on a given transaction and assume that it has ended after the timer runs out. Alternatively, your system might force an end to the session when a resource is used in a new session: this is common with DHCP or VPN servers, for instance.

Because timeouts are a possibility, there is a third type of transaction pattern, which is entirely time-driven.



Let’s dig a little deeper into those timeouts. When our cognitive computing system tracks transactions, it is maintaining a table of session states. That means that it knows which sessions are open now or closed now, which it needs to know so it can evaluate whether a session is normal or not.

A transaction session that uses a limited, finite resource needs to end so that the resource is released. While your cognitive system may not be responsible for controlling that resource, you certainly don’t want it to “think” that a resource is still held when that’s not true in reality. So these transactions have a stop signal, and they have timeouts after which the session is marked closed.

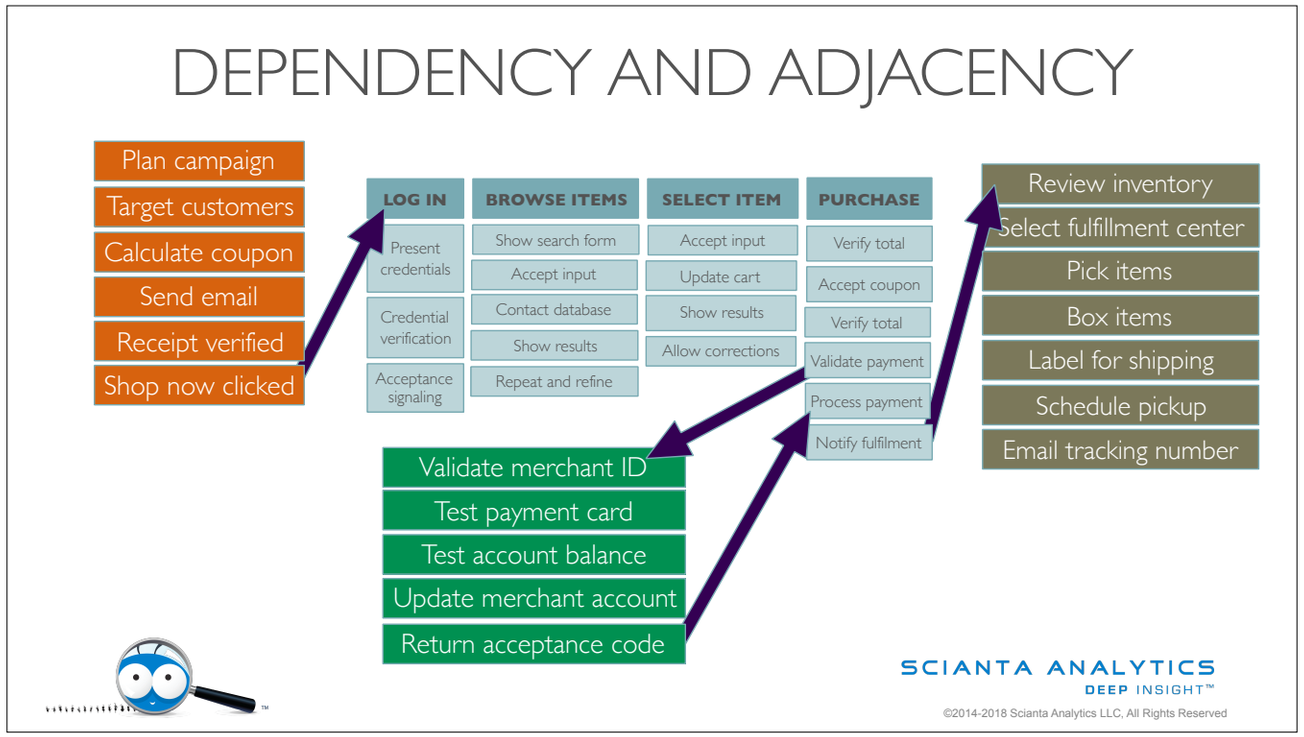


But a transaction that works with an infinite resource doesn't have to do that, so you could have sessions that simply always stay open. These are extremely rare in the real world, so be cautious if you're considering building one. You won't be able to review the session as a whole because you won't know if it's done. It's often better to force a timeout.

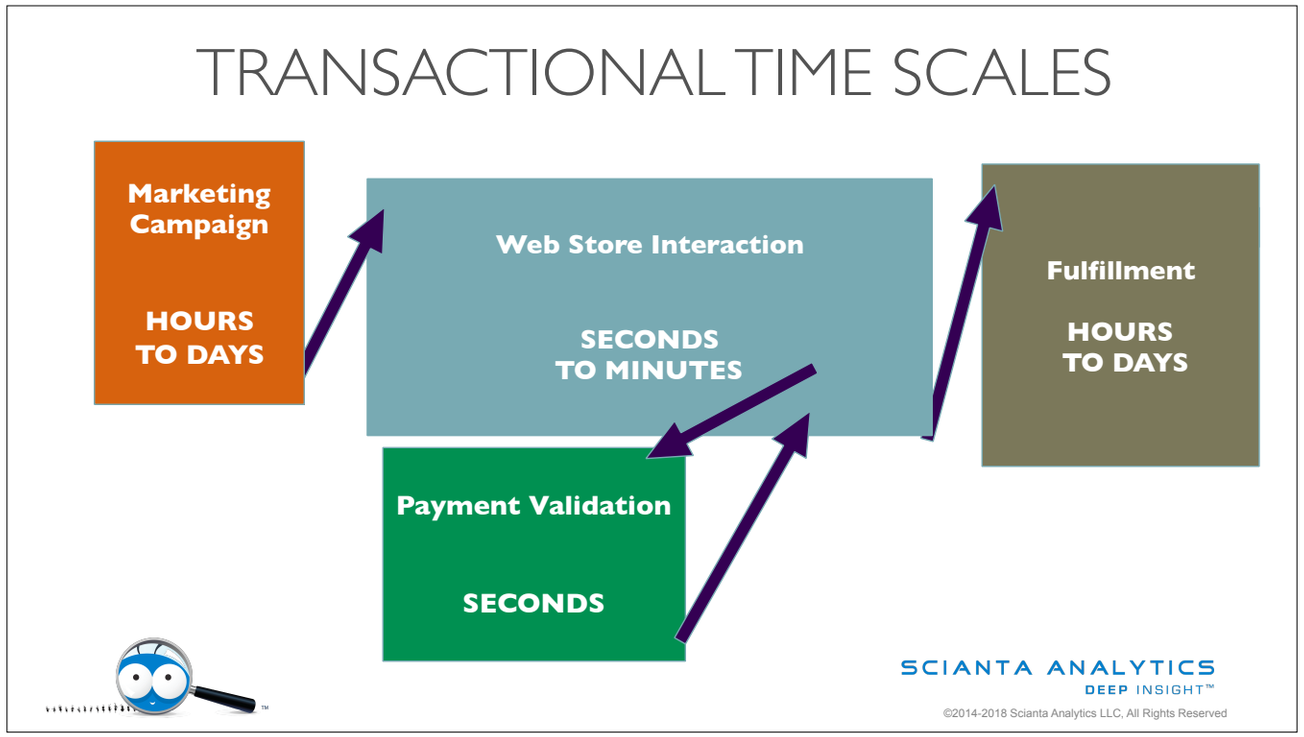
AGENDA

Introduction to Machine Intelligence	Data Handling 1	Data Handling 2	Anomaly Detection	Transactional Behavior	Impact Analysis
Academic Concepts	Collection	Retention	Anomaly Definition	Defining Transactions	Organizational Visibility
Data Systems	Storage	Format	Measuring Normality	Transaction Relationships	Types of Impact
Maturity Curve	Security	Labeling		Probability Measurement	Responsiveness

SCIANTA ANALYTICS
DEEP INSIGHT™



The same concept that defines transactions can be used to review transaction sets, because a given transaction may exist in a web of adjacent and dependent transactions. For instance, our web store transaction may be the result of an email marketing campaign. If it's a successful transaction, then it will trigger a payment card validation process, and if that is successful then it will trigger a pick and ship process. These transactions are related to each other. A transaction depends on another when it cannot proceed without the first one. For instance a failure in validating payment will stop the webstore shopping experience. A transaction is adjacent to another when they can be related but do not have to be. The email campaign may have caused the shopping experience, but our user may also decide to shop without the email.



And here you can see why transaction timeouts might be an interesting challenge. Some sessions can be open for a very long time indeed.

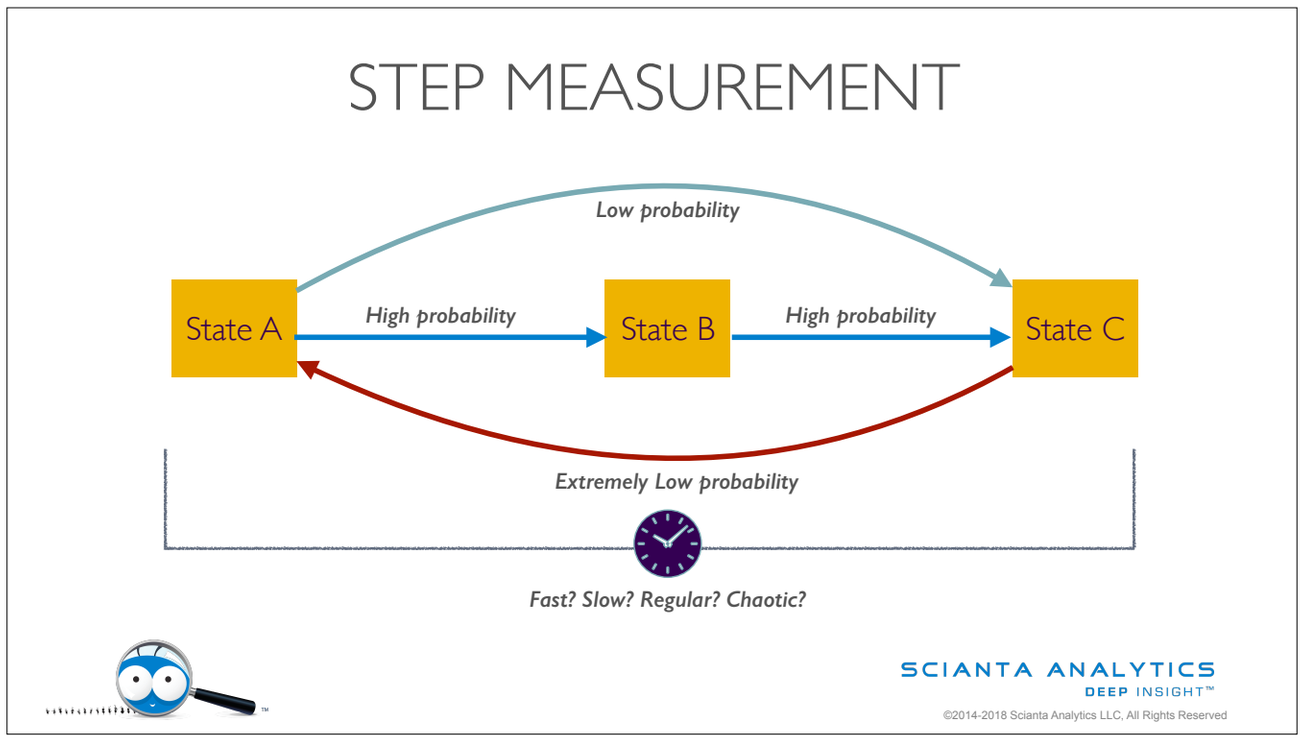
When transactions are adjacent, this is not an issue, because a fast transaction can close while a slow one is still in action. For instance, our payment validation and web store interaction are complete even though our marketing campaign is still active and the ordered items still haven't left their shelves.

But when transactions are dependent, the child must complete before the parent can be closed. It would be an error for the data engineer or cognitive system to close the web store transaction while the payment validation was still in progress, for instance. Note this doesn't mean that the user can't! If they cancel the transaction, that's that.

AGENDA

Introduction to Machine Intelligence	Data Handling 1	Data Handling 2	Anomaly Detection	Transactional Behavior	Impact Analysis
Academic Concepts	Collection	Retention	Anomaly Definition	Defining Transactions	Organizational Visibility
Data Systems	Storage	Format	Measuring Normality	Transaction Relationships	Types of Impact
Maturity Curve	Security	Labeling		Probability Measurement	Responsiveness

SCIANTA ANALYTICS
DEEP INSIGHT™



Okay, let's dig into what we want to do with a transaction… the first question that we have within a session is how likely each step was. For instance, let's say that most actors tend to go from State A to State B and then on to State C. Each of those movements has high probability, so we don't want to alert users that they happen. Going from State A directly to State C is low probability, so maybe that's worth an alert. Going from State C to State A is unheard of, so that's absolutely worth an alert. If this conversation is reminding you of the 95th percentiles and control charts we talked about earlier, that's good. In this model, we are assuming that each of these transitions has what's called a normal distribution of probability. That may not be an accurate assumption for all transactions though.

Another interesting attribute to measure in a transaction is how long it's taking. For instance, we might expect a software agent to proceed through our web store very quickly and regularly, while a human shopper would be much more unpredictable.

SEQUENCE MEASUREMENT

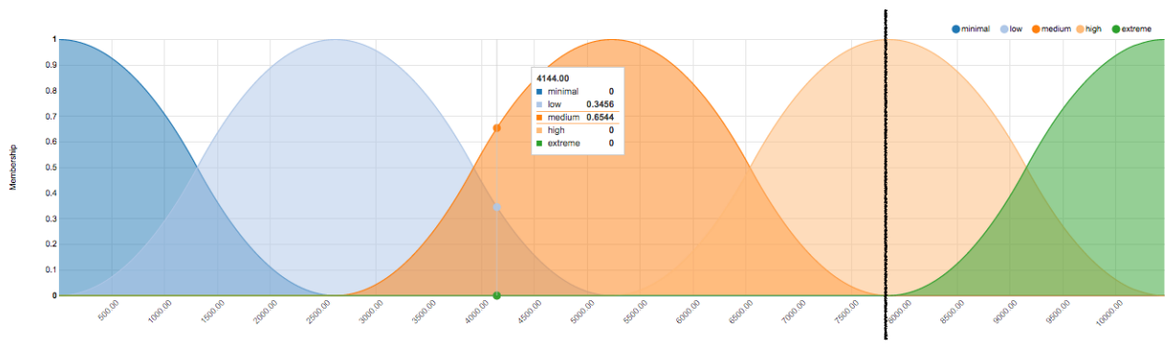
Alice	Bob	Charlie	Deb	Eve	Frank	Greg	Harry	Irene	Jane
Step A	Step A	Step A	Step A	Step C	Step A	Step A	Step A	Step A	Step A
Step B	Step B	Step B	Step B	Step A	Step B	Step B	Step B	Step B	Step B
Step C	Step C	Step C	Step C		Step C	Step C	Step C	Step C	Step C

SCIANTA ANALYTICS
DEEP INSIGHT™

Another effective way to consider transactions is to model each actor's sessions against everyone else in their cohort. Just like we can do with individual events, only even more powerful because it looks at the entire set of events within a single behavior.

FUZZY AND CRISP REPRESENTATION

```
If  Value > $CRISP_THRESHOLD then &Alert("Value is over threshold!");
```




```
If  Value is over medium then &MeasureRisk();
```

Fuzzy representations and rules can open some interesting doors as well. This is a very deep subject that we're not going to spend a lot of time on right now. Suffice it to say that you don't have to set a qualitative number for every measurement and encode that value into every alert. Instead, Fuzzy rules allow the user to say "tell me when the value is very high" or "tell me when moving from State A to State B is slow" or "tell me when there are many users going from State A to State C".

The traditional approach is to state a quantitative number that is the point at which you're concerned. A more advanced approach is to calculate that number in standard deviations from mean, which requires your user to have taken Statistics 101. An even more advanced approach is to use qualitative, fuzzy representation in your rules.

FUZZY MESSAGING

0-2	2-4	4-6	6-8	8-10	10-12	12-14	14-16	16-18	18-20	20-22	22-0
0	0	0	0	1	0	3	6	19			

The number of failed transactions is growing rapidly!

SCIANTA ANALYTICS
DEEP INSIGHT™

Fuzzy representations also lets you communicate results back to the user in a more useful way. For instance, let's say your cognitive computing system is evaluating rates of change as well as facts of change. Wouldn't you rather be able to say "The number of failed transactions is growing rapidly" instead of "The number of failed transactions is now 19"?

Alerting systems that depend on their users knowing enough context to say if 19 is good or bad are the ones that fail when people get sick or go on vacation.

"The natural evolution of machine learning, Cognitive Computing attempts to imbue, in computer systems, the same insight and understanding we see in humans."

Earl Cox
Chief Scientist, Scianta Analytics
Splunk .Conf 2013

SCIANTA ANALYTICS
DEEP INSIGHT™

Computers are force multipliers, not analysts; but they can help an analyst be more productive and successful. I hope this has been a useful overview of transactions and behavior; next, we will discuss impact analysis. Thank you!

Thank you!