Discovering value leaks and service imperfections in business processes
Peters, E.M.

Citation for published version (APA):

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: http://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
Chapter 3
The Tooling and Process for Process Discovery

3.0 Introduction

This section provides an overview of the Comprehend product suite developed by OpenConnect Systems (OC), Dallas, Texas, USA, in conjunction with research performed at KU Leuven. OC designed Comprehend to perform automated business process discovery primarily for service sector operational processes where the service is performed on a large scale and where data-workers operate business-critical systems. Some ideal operational environments are manual claims processing, contact center workflow analysis and productivity improvement, telecom order entry and tracking, and so on healthcare insurers in the United States use the product extensively to reduce administrative costs associated with processing claims and adjustments, a particularly expensive form of rework. Comprehend users routinely improve operational first pass rates (a measure of process productivity) used by healthcare claims rates by several percentage points. These business users have spent years improving their processes and have reached diminishing returns for their efforts because they did not have the ability to measure worker activity in detail and/or process variations. Therefore, they could not calculate the true costs of their workflows. This section describes the technology and methodology at a high level. The technology description represents the architecture of Comprehend. The methodology section discusses each major step in the method including roles and process steps; it also includes actual system screenshots and describes inputs, transformations and outputs.

3.1 The Origins of Comprehend

Comprehend is comprised of several software components working together to provide data collection, business process discovery, process mapping, analysis and reporting of operational process information.

The journey, whose outcomes to date are presented in this Thesis, began in 2005. I had recently joined at OpenConnect Systems, Dallas, TX, which at that time was the producer of a software tool named WebConnect, a web to host (IBM 3270 Series) terminal emulator. The product allowed a person to use an internet web browser to connect to a “green screen” mainframe system thus allowing the legacy technology to utilize a modern web interface without the operator having to do anything more than what they did on the old system. By 2005, this market (known as terminal emulation) was no longer growing and had fully commoditized.

The WebConnect product consists of a number of components, the main grouping being a “listener” or data collector which can intercept the signals (packets) as they are transferred over
the mainframe 3270 communications protocol. It moves across the wire from the mainframe to
the desktop machine and finally a screen painter. The data in these packets shows the order in
which the user accessed the application at the screen level starting with the log-on procedure. It
then shows how they moved from data field to data filed, as well as from screen to screen, as
they performed various tasks. Since the technology was based on the overall 3270 standard for
terminal emulation, it captures the data regarding any 3270 (mainframe) emulator session, not
just those of OpenConnect. This led me to believe that if this data could be captured and stored,
it would represent a true sourced data about “work” as it “actually occurred” since it provided an
activity log of what the person “actually did” as they worked with an application to perform
some defined task. In essence, the data were captured from the work performed by the employee
at the user interface itself and, therefore provide source data about the actual instance of work.
My thought was that if one had this type of information, it could provide valuable insights into
how time and resources were being spent to accomplish some defined task as well as show
whether they were being used effectively. As a young industrial engineer, I had spent time
performing task analysis and productivity studies for a healthcare insurance company. I used
work activity analysis and other direct observation and self-reporting data collection methods to
track the productivity of medical claims examiners using ICT-based systems. In this approach,
one of the major concerns is the integrity of the data and its collection, ranging from the role of
the observer to the veracity of the data when self-reporting methods are utilized. This was a
shortcoming in the process since the actual work was in essence manually “unobservable” and
the recording methods were based on manual observation.

With my work at OpenConnect, my initial hypothesis was that since the actual observations were
indeed being recorded and captured by the technology itself, if it could be stored and re-created,
it would be an actual or fact-based representation of the actual activity rather than what
amounted to a best guess. Also, the recorder sequence of actions would represent an actual flow
of the process, not an idealized flow or something based on inference, speculation or opinion. In
essence, the event logs generated could be used as the basis for reconstructing actual processes
around applications.

As this effort was to be based on captured data, it would be a “bottom-up” approach. This would
be different from the normal “top-down” modeling effort. Specifically most commonly used to
top-down modeling efforts provide a taxonomy at the beginning of the process. I was clear the
effort needed to be based on an “emergent paradigm” since the data would need to be at the
lowest (atomic) levels and the groupings would need to be found or “discovered”. During this
time I became familiar with the work of Cook and Wolf (Cook & Wolf, 1995/1996). They had
coined the term “process discovery” while working to automatically uncover processes in the
software development effort. They suggested three techniques that could be used; Neural
Networks, algorithm-based techniques and Markov Chains. After much discussion, it was
decided to use Markov models. This technique discovered clusters and provided a representation
that can be best described as a probabilistic finite state machine. This was a revelation in two
ways; first, it showed the probabilistic nature of the work process and second, it showed that there were many issues in the model itself. Specifically, while we could initially understand the model, it contained a level of ambiguity in truly understanding the root cause of any deviation from the main path of the model (e.g., process anomalies). This was a key moment since we now understood that it was not the main process that had to be “discovered” but rather, it was the exceptions (or the noise) that were of key interest rather than a generalized process. What we understood from our field experience was that it was the process exceptions that were root cause of problem that drained productivity from the environment hence “leaked value” from the process. We focused on finding which techniques were most suited to understand value leaks in business processes and assist in understanding how to deal with these value leaks (e.g., close the knowledge-gap on the true cause of value leaks and service imperfections thereby providing guidance to management on ways to improve process performance)?

It was at this point that I contacted Professor Dedene at KU-Leuven. We discussed his research in probabilistic finite state machines and its applicability to automated process discovery. This resulted in an OpenConnect sponsored Research Chair at KU Leuven to focus on the issues of automated process discovery and analytics. As an initial step, we sponsored an open, roundtable discussion that included various academic and industry representatives with products and interest in the subject. The session included presentations from the Gartner Group, BMC, Software AG, OpenConnect and KU Leuven. To the best of my knowledge, it remains the only multi-vendor, industry analyst and academic session devoted to this topic as of this day.

After the aforementioned industry and academic session, we worked with Professor Dedene and his team to explore different approaches to the problem of anomaly detection and elimination in Process Discovery. I remember clearly a dinner with Professor Dedene, where I mentioned that the problem of process anomalies reminded me of the data normalization issue where ambiguity was removed through the process of normalization. This triggered other thoughts leading to the development of the hybrid model described in Chapter 2 of this Thesis. This was clearly the major scientific breakthrough of the effort.

As stated earlier, the primary goal of automated business process discovery (process discovery) is to “understand” processes rather than to “control” processes. It is the increased understanding that leads to the closing, or at least narrowing, of the management knowledge-gap regarding the root cause of process inefficiencies.

Remember from chapter 2 how Process Discovery tries to go further than Process Mining, by explicitly addressing process variations and process exceptions. Process Discovery incorporates the exceptions and variations and investigates them as the root cause for value leaks in the process.
3.2 Architecture

*Comprehend*'s architecture is depicted below (Figure 3.1). *Comprehend* adds the critical time dimension to the usual volumes that previous analysis products use.

**Major Comprehend Components:**

- **Collectors**
  - Passive ‘Sniffers’ for mainframe and web traffic
  - Agent for desktop client / server applications
  - Custom data import and normalization

- **Analytics Component (AC)**
  - Logic for discovering, defining, delimiting and labeling activities, aka logical units of work
  - Builds events from clickstream data,
  - Augments input data with process metadata, and provides input to automated process mapping

- **Process Intelligence Component (PIC)**
  - Process mapping, rather than modeling, based on empirical data
  - Process analytics based on event data
  - Interactive Analytical workflows

- **Reporting**
  - PIC analysis is exposed to standard reporting engines
  - Production style reports, aimed at operational leads, managers.
  - Input in standard formats for ETL to existing data marts or data warehouses
  - Input to operations dashboards

![Figure 3.1 – Comprehend Architecture](image)
3. THE TOOLING PROCESS FOR PROCESS DISCOVERY

3.2.1 Collectors

3.2.1.1 3270 Collector

The 3270 Collector is a network "sniffer" (i.e. makes copies of packets as they traverse the network) that observes TCP/IP network traffic and records only TN3270 sessions (all other traffic is ignored). These session files are stored as collector specific files locally on the collector hardware. Because of the low-level access to the Ethernet device, the collector component stores and encrypts data on the fly. Additionally, since the collectors passively observe and record traffic as it flows across the communication channel, it makes no changes to the monitored servers or software, and does not cause any performance degradation. Additionally, password fields are redacted in the recorded screen shots and not stored in Comprehend.

On installation, the collector is provisioned through a local HTML wizard that requires the installer to provide the collector with an encryption key phrase that is shared between the Analytics Cluster and the Collector, and used to encrypt/decrypt the collector specific files. This key phrase is only used to generate a shared key for Collector-AC Server communications and is never required or used again. The installation also requires the administration password for the Analytics Cluster to register the collector with the AC Server. Any change to the credentials requires a re-installation of the Collector.

All further configuration of the collector is done through the AC server. The collector can be configured to monitor specific internet protocol (IP) & port ranges on the client side or the server side.

The 3270 collector captures all TN3270 traffic, and provides a temporary encrypted collector specific store of these sessions. However, all non-display modifiable fields (typically password fields) are never stored on the Analytics Cluster; they are discarded during initial analysis.

3.2.1.2 HTTP Collector

The HTTP collector is a network sniffer that observers and records web-specific TCP/IP traffic and stores steams as user sessions. These sessions are encrypted and stored locally as collector files. Because of the low-level access to the network device, the collector runs as root. On installation, the collector is provisioned through a local HTML wizard that requires the installer to provide the collector with an encryption key phrase that is shared between the Analytics Cluster and the Collector, and used to encrypt/decrypt the collector specific files. This key phrase is only used to generate a shared key for Collector-AC Server communications and is never required or used again. The installation also requires the admin password for the Analytics Cluster to register the collector with the AC Server. Any change to the credentials requires a re-installation of the Collector. The configuration of the collector is done through the local interface; the local interface provides three logical functions traffic filtering, data exclusion and HTML stream configuration. Traffic filtering allows the collector to monitor only specific IP
addresses & port ranges on the client side or the server side. Data exclusion rules strip data out of the captured HTML stream ensuring that unwanted data is never stored locally, or transmitted to the AC. Traffic filtering is often used to strip sensitive information, (e.g. passwords, etc.,) out of the data stream. Finally HTML stream configuration is a set of rules that encompass session detection, page naming and content extraction.

3.2.2 Comprehend Analytics Component (AC)

The Analytics Component (AC) analyzes collected data, and stores it in a relational database. The analyst interacts with AC to determine when events start and stop, what data will be kept and discarded, groups collected data into activities which are logical units of work, and timestamps each event. The collector specific files are pulled from the collectors via a HTTPS connection, and are analyzed and then stored into the relational data store. The user interaction with the AC is via an Eclipse workflow connecting to the server process via HTTPS connection. On installation, three sets of credentials are required, database administrator, Comprehend database user and Comprehend AC administrator. The relational schema is created in the database using administrator credentials to the database. The Comprehend Server uses the database user credentials to access the database; this can be a normal user role without any data definition language specific (DDL) privileges. The Comprehend AC administrator is the system administrator for the AC and is responsible for configuration and installation. All of these credentials are available for modification through the administrative interface of the Comprehend AC. All data collected is stored in the relational data store. The “raw” data collected is stored encrypted as binary large objects (BLOBs) within the database. Users with access to the encryption key phrase can view this “raw” data.

This data includes:

Session level replay – the system will be able to decrypt the collector specific file and replay entire sessions.

Field level data – For any Web Page or TN3270 panel, the system can show all values observed for any particular field.

The following data is available to any user of the Comprehend AC:

- Statistical data relating to usage paths through the system
- Statistical data relating to page / panel use
- Statistical data relating to think time / system time for individual transactions

3.2.3 Comprehend AC User Interface

The Comprehend AC User Interface (UI) is a set of Eclipse plug-ins. These provide the interaction between Eclipse and the AC server. All communication is done via HTTPS and users
are authenticated locally via challenge by Eclipse. A secondary challenge prompts for the data encryption key phrase. On 15 minutes of inactivity, the challenge expires and upon the next invocation of data secured functionality, the user is re-challenged.

User credentials are managed through the administrative user interface of the Comprehend AC. Credentials can be expired, deactivated, created or changed via this interface.

3.2.4 **Comprehend Process Intelligence Component (PIC)**

The Process Intelligence Component provides fast *ad hoc* analytics and reporting on captured process events. All user interaction with the PIC is via a web browser, with a variety of user specified roles and credentials. The Process Intelligence Cluster is a set of java processes and an XML database that provides the analytics and reporting on the process events. All user interaction with the PIC is via an applet or HTML browser interface over a HTTPS connection.

On installation, an internal shared key is generated used that is used to encrypt/decrypt any secured data. The shared key is not used externally to the application. All user administration and provisioning is done via the web server, and authenticated users (with an appropriate role definition) are passed to the PIC UI. The PIC data security role controls what data elements within PIC are encrypted vs. unencrypted. This is done at an attribute level. User roles within PIC can be created such that these roles have one of three levels of access to the PIC data, decrypt, view, none. Decrypt access provides the ability to see the attribute values unencrypted. The server uses the shared key to decrypt the data and pass it over the HTTPS to the application. View access allows users to view the data, and if the attribute is encrypted, the user is able to see encrypted values of the attribute values, but not the values themselves. No access provides no visibility to the user that the attribute even exists within the PIC.

3.2.5 **Reporting**

Reports can be extracted from *Comprehend* through any standards-based (e.g., ODBC, JDBC, XML) report writing software. While LogiXML is the embedded reporting engine, *Comprehend* data can also be exported to Microsoft Excel and other analytical tools (e.g., Tableau. LogiXML) where data further examined or combined with external data for other types of analysis. Project teams routinely roll up data into business objects; say lists of all orders touched in a day, or all users who touch each order. This information can then be sent to existing reporting engines, and analysts can use extract transform and load (ETL) tools to integrate it into their environment, though it is also available on a standalone basis and integration is not required.

3.2.6 **Security Provisions**

The following sections summarize the security provisions and default security roles built into *Comprehend*:
3. THE TOOLING AND PROCESS FOR PROCESS DISCOVERY

3.2.6.1 Data storage protocols

All data that is stored is encrypted using 256-AES encryption, where legal. No captured data is ever stored or transmitted in the clear. At the AC/Collector level, there is the ability to configure Comprehend to strip data out of the event stream ensuring sensitive data is never stored.

3.2.6.2 Data transmission protocols

All network communications between major components of the Comprehend solution is performed over SSL. Where possible, the data stream within the SSL wrapper is also encrypted. Additionally, Comprehend is generally deployed on the customer campus so that sensitive data is less subject to interception.

3.2.6.3 Audit

Comprehend provides an audit log of system usage and configuration changes.

3.2.6.4 Usage Log

System usage can be logged and time-stamped, providing an audit log of tasks executed in the system.

3.2.6.5 System Change Logs

Whenever there is a change to the configuration of Comprehend, a system change record is generated. Specifically, changes that relate to:

4 Changing a role definition (add/delete/modify)
5 Changing the configuration of a collector (source monitored system)
6 Changing a data exclusion rule (omit data from a collector data stream)
7 Changing the encryption passphrase
8 Changing the level of security on an attribute in PIC
9 Adding new attributes to PIC

3.2.7 Standard Roles

While in any implementation there is configuration of roles. Comprehend includes the following default roles:

3.2.7.1 System Administrator

The system administrator has only rights to install and do base level configuration of Comprehend. This would include data retention policies, server start/stop/status, system upgrades, etc. The System Administrator does not have any rights to view any of the data collected by Comprehend.
3.2.7.2 Data Security Administrator

This role controls access & visibility rights to the data collected by Comprehend. This includes the ability to configure the systems the collector is monitoring, any data exclusion rules, as well as the control of what attributes are encrypted/viewable by role.

3.2.7.3 Security / User Administrator

The Security or User administrator provides user level administration to Comprehend. This role is responsible for managing the user to role mapping (done in the existing user management solution integrated to Comprehend). As well as would modify roles based on new use requirements by the user community.

3.2.7.4 AC – Developer

The Analytics Cluster developer is responsible for analyzing the system usage data, and creating processes. This role typically has access to view the unencrypted session level data. However, this is not required.

3.2.7.5 PIC – Analyst

The PIC analyst role(s) are the power users of the process analytics in PIC. These analysts investigate the process variations; examine data attribute values that are driving the variations.

3.2.8 Comprehend Artifact Capture

The key benefit of automated business process discovery is, as the name suggests, the ability to automatically capture the elements of work in a digital environment from the work process itself and re-create the activities and events as they actually occurred during the execution of the process. Artifacts (e.g., data, time, user id, etc...) are captured automatically by the previously described collectors from ICT-based work sessions on multiple platforms (TN3270, client-server/desktop, web) or from various log or audit files. Additionally, Comprehend has facilities to augment collected data with relevant metadata, to query databases in order to collect process changes prior to deployment, and in cases where manual capture is the only available method, an xml data input mechanism is also available.

As shown previously in Chapter 2, the basis for the input data for analyzing the artifacts is the Actor – Event – Object model, which is open-ended to any type of process characteristics (such as time, risk probabilities etc) that desire to be analyzed. The UML representation for this model is shown in Figure 3.2. This representation easily accommodates the transformation of the data from logging components into process representations that can be reviewed for accuracy by process owners or domain experts.
3. THE TOOLING AND PROCESS FOR PROCESS DISCOVERY

3.2.8.1 Session Capture

All information about work is discovered through capturing the data in real time from the process itself. Specifically, data is captured from the actual work session that is being performed by a resource (person) as they interact with the application through the user interface (screen). In this manner, every interaction (keystroke, data change, etc.) the resource makes with the screen during the session is extracted and captured along with a time-stamp of when the page was viewed, the data change made as well as the IP address of the client. In essence, the session is described by who the resource was, the time the session took place, a view of what was viewed as well as what was changed. Information can be captured from mainframe/3270 platforms, client-server/desktops or web-based sessions.

With session capture every page or screen image is captured and stored. Specifically, data is extracted from each data stream, time stamped as to when the page was viewed and includes the Internet Protocol (IP) address of the client workstation on which it appeared. For IBM mainframe-based systems, the full TN3270 user payload is extracted for both pages (if working under a terminal emulator) and a screen image is recreated. In a desktop based client-server/environment, any desired data can be captured and stored and the data is extracted from the screen and stored as data in the database. A similar process is used for web-based systems. Each type of session is described in more detail below.

3.2.8.2 Mainframe Session Capture

Mainframe session capture is based on intercepting and copying the signals sent between the mainframe itself and the client workstation (Figure 3.3). In this manner, all TN3270 newtork
traffic is captured. User workscreen information is captured based on analyst-specified information transmitted between the user and the application. After initial capture, the appropriate panel rules are determined based on the observed consistent placement of text on the panels of interest. Next, data/field tags are created to extract information from the TN3270 stream based on field tag name, panel position regular expressions, or other criteria. These fields can then be manipulated through JavaScript or regex (Linux/UNIX regular expression parsing).

**Figure 3.3 – Mainframe Session Capture**

### 3.2.8.3 Client-Server/Desktop Session Capture

Client-server/desktop session capture differs from mainframe session capture in that the information is not intercepted from the network communication transmission but rather from the actual changes on the desktop device itself (Figure 3.4), by a resident agent with a small memory footprint. This is necessary as the important data may never traverse the network. In this manner, user interactions, process name, window titles, before and after data, focus change; Windows handles (for differentiating identical windows) are all available for capture and analysis. In this environment, field level data values are extracted from Windows controls through the use of code embedded rules or “triggers”. Triggers are implemented on a central server environment and “pushed” to the desktop by standard enterprise methods. All data is logged locally on the client/desktop and sent or “pushed” to the desktop collector server in near real time, in order to reduce the memory footprint that would occur by storing data locally.
3.2.8.4 Web Session Capture

Session capture in terms of web/internet data is similar to the mainframe oriented methods described earlier, or can be captured by the desktop collector agent should the project require more detailed analysis. In one former case, it is pure, passive “click-stream” information that is of high interest. In web data capture, page naming rules are defined based on matching URL stems, page content and page tags (Figure 3.5). In the latter case, Comprehend can collect and report on detailed workflow analysis among web and desktop applications.

The rules for content extraction are described in XPath. XPath is a programming language that XML uses path expressions to select nodes or node-sets in an XML document, hence the name.

3.2.9 Activity Definition

The Comprehend Analytics Component is where activity definition takes place. Activities consist of a set of low-level actions comprising a logical unit of work that is being observed.
Activities generally consist of several user interactions and may navigate several screens, panels or desktops. In the definition of activities, screen-to-screen transitions are combined based on observed actual user usage patterns. Groups of activities form the basis of “events” (e.g., input claim) that are generated by Comprehend as a component of the discovered business process. All information is based on data that are extracted from activities that are defined by member pages of the activity itself (Figure 3.6).

3.2.10 Event Generation

Analytics Component – Event Generation

As described in Figure 3.6, activities define the core elements associated with business events. Once defined, data associated with all pages, screens, clicks and keystrokes are collapsed into an event. Each event is described by an event descriptor that includes both attributes in XML documents (e.g., screen name, user id, and IP address) and actions as attributes of the process. This also includes the source/destination of the information as well as a timestamp, and an activity name associated with the event.

Figure 3.6 – Definition of Core Elements
3.2.11 Process Intelligence Component

The Comprehend Process Intelligence Component (PIC) is a process intelligence engine enabling more abstract business processes analysis of activities defined by the Comprehend Analytics Component (AC). Where AC analyzes logical units of work (activities), PIC provides the ability to perform analysis on an entire process (a collection of activities) based on any attribute or combination of attributes contained in the processes. Additionally, PIC allows for multi-system analysis by combining real-user interaction on disparate systems and augmenting this information with data from existing system interfaces, such as log files, databases, or essentially any user defined data feed, for a full end-to-end business process analysis.

3.2.11.1 PIC Process Perspective

The PIC can perform multiple analyses on the same set of data. Each analysis is called a perspective. Each perspective is based upon the presence of a single designated attribute that is selected by the user. Every event in the system that contains the designated attribute is grouped by unique attribute value, ordered by time, and passed to the analysis script for processing. Of course, additional attributes are also typically present, such as critical data fields to be analyzed, userid, timestamp, and activity name.

The grouping and ordering of events creates a series of traces of which each trace is associated with a specific value of the specified attribute in the perspective. Perspective analysis generates the nodes and transitions on the activity map, trace attributes, and rollup events. Trace attributes are data elements annotated onto each event that can be filtered and charted just like a normal event attribute. Rollup events are synthetic events created by the script, contain only trace attributes, and are used to summarize output from the analysis of a trace.
3. THE TOOLING PROCESS FOR PROCESS DISCOVERY

PIC Process Perspectives are data views based on a specific attribute. They are created through an analysis of the event data where the events are used to construct processes consisting of activities. Activities are a particular type of event, (i.e., login).

An example of an attribute is a *Comprehend* field tag such as UserID. This attribute (Figure 3.7) represents the values of the User IDs used with the defined processes. In this way, the perspective provides the analysis of how users move from activity to activity.

As PIC acquires new events from AC, the perspective is updated. The Perspectives panel allows the analyst to create Perspectives based on attributes to create a specific view of the data. They can be edited to change properties associated with it. The use of open ended user defined scripts allows new trace attributes (often called business objects), the combination of existing attributes into trace attributes, or the suppression of events relating to existing attributes.

PIC Analysis Scripting allows the user to customize the data created by the analysis process. The default analysis performed by the PIC server generates the transition graph from activity to activity, and calculates two default trace attributes. The default attributes are \( \text{deltaT} \) and \( \text{deltaP} \). These attributes contain the time in milliseconds between the current event and the next or previous event respectively (Figure 3.8).
3. THE TOOLING AND PROCESS FOR PROCESS DISCOVERY

Figure 3.8 – Combined Process Map

The following picture shows the example from Chapter 2.8 in the Comprehend PIC. In this example, the input data was provided through the Event-Actor-Object data format. It shows how PIC adds more information to the models presented in chapter 2.

Figure 3.9 – Example Comprehend PIC

3.2.11.2 PIC Process Integration and Filtering

Filters are another dimension (beyond X and Y axes) that allows an analyst to test a hypothesis, based on the empirical data in the system can also be set to refine a process map based on any other values in the tag set (e.g., user-id, claim number, etc…). Thus filters provide additional
views to be generated that can examine the timing of events, sources, or other items of interest. This function allows for the interrogation of the data in order to examine and understand root cause process variations from any perspective on the process.

3.2.12 Comprehend Reporting

In addition to drawing process maps, Comprehend includes a number of graphing and charting capabilities (Figure 3.10). These charts show descriptive statistics on any attribute of interest (e.g., claim edit codes by examiner by date, etc...). A standards-based interface also exists for exporting data to popular reporting and analysis tools such as Microsoft Excel, export to an included relational database, synthesis of events, or any number of user defined extensions to Comprehend's out of the box capabilities.

![Figure 3.10 – Example Graphing and Charting Capabilities](image)

3.2.13 Summary

Comprehend was built to scale to accommodate data capture and analysis in the largest enterprises in the world. It is common to store information on thousands of users with just a handful of industry standard servers. Additionally, Comprehend is flexible. Should the requirements change, it is simple to change the data collected, analyze the process in a different way and determine whether the hypotheses necessary to improve a process can be accepted or rejected, based on empirical data, not assumptions and guesswork.

3.3 The Process of Using Comprehend

The concept of automated business process discovery is based the analysis and instrumentation of data that is automatically captured from the work activity itself rather than by human observation of the process. Automated Business Process Discovery tools capture the required
data and transform it into a dataset that can be used for analysis. This differs from other process analysis methods where the work activity is observed by a person, often with the assistance of tools (stop-watch, video camera, etc…) and then recorded for analysis. As discussed in Chapter 1, the standard method is often error-prone, costly and leads to sub-optimal conclusions. In contradistinction to the afore-mentioned approach, the methodology used with Comprehend centers on the automatic capture of data from the process itself as the starting point for the analysis of the current environment. Its use supports a discover, design, build/implement approach which is based on an emergent, bottom-up approach to understanding what is known as the “as-is” environment. Whereas most top-down modeling methods move quickly past this step to get to future solution design, the use of Comprehend facilitates a deep understanding of the “as-is” state and looks to uncover the root cause of the process variations. Each variation is seen as a potential process value leak that, when corrected, has the potential of recovering significant economic value for the organization.

Comprehend’s approach, while software-centered, is collaborative. The Comprehend analyst works closely with both a solution architect and business subject matter expert to validate the discovered process and its variations as well as propose solutions. Also, there are no parallel activities in this methodology where one group focuses on a specific set of tasks while another works on different set. This lack of parallelism maintains unity of focus and common shared understanding. It also prevents errors due to misunderstanding of the same facts.

The following sections describe both the roles required for a Comprehend-based automated business process discovery project as well as the actual process steps to complete an effort. An example is presented that shows the use of Comprehend including process steps, input/output of each step and tool screen-shots for illustrative purposes. The case of an adjustment made to a healthcare claim is the subject of analysis in the example. The full detail of the healthcare claims processing environment can be found in Chapter 5, Case Study 1.

### 3.3.1 Roles

The following roles are defined for this process:

a. Comprehend Analyst
b. Solution Architect
c. Configure Developer
d. Business Subject Matter Expert (SME)
e. Business Approval Committee
Each role is defined in more detail below in Figure 3.11.

![OpenConnect Automation/Improvement Methodology](image)

**3.3.1.1 Comprehend Analyst**

The *Comprehend* Analyst has two primary roles: to analyze and report on the operation of the claims processing function using OpenConnect's *Comprehend* Analytics Cluster (AC) and Process Intelligence Cluster (PIC) analyses; and to perform a detailed analysis of existing, manual workflows used in creating and supporting the automation process.

The *Comprehend* Analyst creates a *Comprehend* Analysis Report (CAR) summarizing PIC's analysis. The Business Approval Committee uses the CAR to select edit codes to be automated or improved.

The *Comprehend* Analyst also creates Initial Requirements Documents (IRD). Each IRD contains edit code-specific workflows based on AC analysis. These workflows provide edit code business process details used to discover potential improvement areas or automation opportunities.

**3.3.1.2 Solution Architect**

The Solution Architect works with the Business Subject Matter Experts (SME) to convert the IRD to a full Business Requirements Document (BRD) by adding existing business logic obtained in SME interviews. The Solution Architect passes the BRD to the Business Approval Committee.
Committee for final approval. The Business Approval Committee passes the final approval version of the BRD to the Configure Developer to code the transaction.

3.3.1.3 Configure Developer

The Configure Developer uses a BRD to implement claims automation transactions using the Configure Development Environment. The Configure Developer also performs unit testing before deploying a transaction to the runtime environment for user acceptance testing.

3.3.1.4 Business Subject Matter Experts

Business SMEs are critical to the improvement process and are involved at multiple stages. They consult with the Comprehend Analyst and Solution Architect to insure correct design of the claims business workflows are properly set up specific to their line of business, if needed. After the BRD has been created, the Business SMEs validate the business rules. The validated BRD goes back to the Business Approval Committee for acceptance.

3.3.1.5 Business Approval Committee

The Business Approval Committee is responsible for selecting the edit codes to automate or improve based on the CAR and their extensive business knowledge. Once the BRD is created and the Business SMEs have validated the business rules, the BRD goes for final approval with the Business Approval Committee.

3.3.2 Discover

The Discover stage is executed by the Comprehend Analyst to discover potential process automation or improvement opportunities. The Comprehend Analyst uses PIC to analyze events generated by AC. The analysis is performed using the following criteria:

  a) Impact to the business by Edit Code
  b) Distribution by Line of Business
  c) User efficiency by Edit Code

The Comprehend Analyst generates the CAR which includes the top ten edit codes, the top three edit codes that follow each top ten edit codes, and the top three edit codes that follow those. This CAR also includes a graph for each top ten edit code showing the distribution by line of business. The process to create the Comprehend Analysis Report is provided in more detail below.

  • Edit Code Analysis

This section describes the Edit Code analysis steps using PIC (see Chapter 5 for a complete description of the healthcare claims process, edit code analysis activity).
3. THE TOOLING PROCESS FOR PROCESS DISCOVERY

- Impact to the Business by Edit Code

Business Impact analysis can help determine which edit codes to automate or improve based on the actual cost to the business.

![Figure 3.12 – Chart Group](image)

- Process Intelligence Cluster Process Map

The business impact is determined by the total time spent by users to work specific edit codes. The actual impact to the business is calculated using the formula:

$\text{total \ time \ in \ hours} / \text{hours of work each day} \times \text{hourly \ rate}$

The following steps create the top edit code by user time chart:

- Place your mouse over one of the activities
- Right click and select 'New Chart'
- If this is the first chart, you will be prompted for a chart group (Figure 3.13)
  - Enter 'Edit Codes' and click 'OK'
- You will now be prompted for a chart name and properties (Figure 3.14)
  - Chart Name: Edit One by User Time
  - Chart Type: Line/Bar Chart
  - Chart Group: Edit Codes
  - Check the 'Floating' check box
  - Click on 'Next'
3. THE TOOLING AND PROCESS FOR PROCESS DISCOVERY

Figure 3.13: New Chart Group

Figure 3.14: New Chart Dialog
3. THE TOOLING PROCESS FOR PROCESS DISCOVERY

Figure 3.15: Add X-Axis Attribute

g) Configure the X-Axis attribute as shown in Figure 15
   a. Attribute: Edit_Code_1
   b. Max Distinct Values: 50000
   c. Click 'Next'

Figure 3.16: Add Y-Axis Attributes

h) Configure the Y-Axis values as shown in
3. THE TOOLING AND PROCESS FOR PROCESS DISCOVERY

i) Figure 3.16
   a. peUserTime - Sum - Bar - Left
   b. peSystemTime - Sum - Line -Right
j) Click on 'Finish'
k) You now have a chart showing User Time and System Time by Edit Code One
l) Sort the chart and resize the view to see the data better
m) You may notice the top value, or one of the top values, is 'No data'

n) To remove this value from the results we need to add a filter as follows in Figure 3.17
   a. In the Attribute Filters section click on the 'Add an Attribute' button
   b. Select Edit Code 1
   c. Click on 'Next'
   d. Select 'Greater than' (Figure 3.18)
   e. Enter a '!' exclamation point in the value input box
   f. Click on 'Finish'

Select the filter you just added.

[Image: Figure 3.17: Add Attribute Filter]
3. THE TOOLING PROCESS FOR PROCESS DISCOVERY

Figure 3.18: Add Filter Value

Figure 3.19: Edit Code One by User Time
By clicking on each of the activities you can see which edit codes have the highest user time and system time associated with them (Figure 3.19). However, this is only part of the analysis. It is also necessary to create a graph showing the total user time spent outside of the activities while working claims. We do this by creating a chart showing the deltaT time for the transitions (see Chapter 4, Section 4.1.17.1 for a description of creating PIC process perspectives).

The following steps create the deltaT analysis chart:

1. Right click on a transition and select 'New Chart'
2. Provide the following values:
   a. Chart Name: Edit One Delta Time
   b. Chart Type: Line/Bar Chart
   c. Chart Group: Edit Codes
   d. Floating: Checked

3. Click on 'Next'
3. THE TOOLING PROCESS FOR PROCESS DISCOVERY

4. Select Edit_Code_1 for the X-Axis attribute (Figure 3.21)
5. Enter 50000 for the Max Distinct Values
6. Click 'Next'
7. Select the following (Figure 3.22):
   a. Transition: checked
   b. Y-Axis attribute: deltaT
   c. Function: sum
8. Click on the 'Add' button
9. Click on 'Finish' (Figure 3.23)
Clicking on the various transitions provides the total time users spent working on claims outside of the standard claims activities. For example, we do not have a membership activity defined so
if users navigate to any of the membership screens while working on an edit, the time spent in membership will be included in the deltaT for the loopback transition.

Distribution by Line of Business

Next, we want to discover the line of business distribution for each edit code. That requires another chart. The following steps create the line of business distribution chart:

1. Right click on an activity and select 'New Chart' (Figure 3.24)
2. Provide the following values:
   a. Chart Name: Line of Business
   b. Chart Type: Pie Chart
   c. Chart Group: Edit Codes
   d. Floating: Checked
3. Click on 'Next'

Figure 3.24: Line of Business Chart Creation
3. THE TOOLING AND PROCESS FOR PROCESS DISCOVERY

4. Select the Line of Business attribute for the X-Axis (Figure 3.25)
5. Click on 'Next'

6. Select Edit_Code_1 for the Y-Axis Attribute (Figure 3.26)
7. Click on the 'Add' button
8. Click on 'Finish' (Figures 3.27)
The next step in the process is to create the same charts for the other edit codes (Edit Code 2-3). These charts are used to create the CAR described next.

### 3.3.2.1 Comprehend Analysis Report (CAR)

The CAR displays the top ten edit codes in position one based on the impact to the business. It also display the top three edit codes in position two for each edit code in position one, and the top three edit codes in position three for each code in position two.

The purpose of the CAR is to show the top edit codes for potential automation and to highlight edit code patterns (Figure 3.28). Many edit codes fall into patterns, so that working one may fix others. The Analysis Report allows the business to see the relationship between edit codes and perhaps to see certain edit codes appear in the list more often than others.
3. THE TOOLING AND PROCESS FOR PROCESS DISCOVERY

3.3.2.2 Business Approval and Direction

The CAR then goes to the Business Approval Committee. The committee will analyze the report and choose a group of edit codes to automate or improve. The list of edits will be provided to the Comprehend Analyst for further analysis. The chosen edit codes will be used in the Design stage of the process.

3.3.3 Design

In the Design stage, Comprehend discovers details of edit code-specific or edit code group-specific claims processes, then feeds them into the automation or improvement plan for the top edit codes. This stage uses PIC to determine which session IDs to import into AC for session level analysis.

Detailed Design Stage Description

The improvement process Design stage consists of the following steps:

- Identify Attributes for Analysis
- Create Filters
- Add Edit Code Filter
- Add Data Filter
- Identify Power Users
- Create Session ID Charts
- Export Data from PIC
- Import Data into AC
- Analyze Sessions
- Build IRD
- SME Sessions
- Build Business Requirements Document
- Finalize BRD
- Business Approval
Identify Attributes for Analysis

It is necessary to determine which few attributes are critical to claims process performance before doing deeper analysis. The most common attribute is the edit code or list of edit codes found in the Discover stage. Additional attributes should be selected based on edit code or lines of business specifics. For example, if the edit code is a membership-related edit, then it may be desirable to look at some of the membership attributes to help determine the most common reasons for the pended claims.

Additional critical claims processing attributes commonly include:

- Edit Code 1 - 4
- Line of Business
- Claim Type
- Subscriber ID
- Provider ID
- Group Number
- COB

Create Filters in PIC

After determining which attributes to use it is necessary to add PIC filters for them. For example, it may be important to know which edit code has the biggest impact to the business by claim type. The chart created in section 0 can show the impact to the business by edit code. However, before any analysis by edit code can be done, it is necessary to create a filter for the specific edit code to be analyzed.

Add Edit Code Filter

Follow the steps below to create the Edit Code filter:

1. Click on the "Add an Attribute" button shown below

2. Select Edit_Code_1 for the Search Attribute as shown below (Figure 3.29)
3. THE TOOLING AND PROCESS FOR PROCESS DISCOVERY

Figure 3.29 - Sample Comprehend Analysis Report

3. Click 'Next' (Figures 3.30)

Figure 3.30 - Sample Comprehend Analysis Report

4. Click on the "Equal to" radio button
5. Enter the edit code to filter on in the text box
6. Click Finish
7. The newly created filter now appears in the Attribute Filter list
8. Single click on that filter to select it
3. THE TOOLING PROCESS FOR PROCESS DISCOVERY

9. Click on the "Add an Or node" button

![Add an Or node button]

10. The newly created filter is shown inside an OR statement

```
  Or
  Edit_Code_1(equal : 0:49)
```

11. Click on that Or statement, then click on the Add an Attribute button
12. Add filters for Edit_code_2 through Edit_Code_4
13. The filter should now look like the one shown below

```
  Or
  Edit_Code_1(equal : 0:49)
  Edit_Code_2(equal : 0:49)
  Edit_Code_3(equal : 0:49)
  Edit_Code_4(equal : 0:49)
```

14. Check the Or check box to select the filter

**Add Date Filter**

Due to size constraints, it is possible, and, in fact, most likely, that the raw data in AC will be limited to one or two weeks while the data in PIC could span four weeks, six weeks or more. Due to that situation, it is necessary to add a filter by time to ensure the session IDs in the chart represent sessions still retaining in AC.

Before creating this filter it is necessary to know the data retention policy set in AC. If you do not have administrator rights for AC it will be necessary to ask the AC administrator for that information.

Next, it is necessary to determine how much data PIC retains. It may require adding activity properties to show the maximum and minimum times using the following steps:

- Click on the Map Display Values button at the top of the window

![Map Display Values button]

[119]
3. THE TOOLING AND PROCESS FOR PROCESS DISCOVERY

- Add the time attributes as shown in Figure 3.31
  - time - min
  - time - max
  - Click on the 'OK' button

The process map will update and display the date and time range of the data in PIC. Then, compare the range with AC's data retention policy. In this example, the data retention policy is 14 days. There is also a minimum date/time in PIC of 1/23/09 4:04 PM, and a maximum date/time of 2/12/09 2:26 PM. This tells us that the latest date in PIC is 2/12 and the data retention policy in AC is 14 days, showing that the filter should be set for 14 days back from 2/12. So, the filter should be Greater than 1/30/09 12:01:00 AM as shown in Figure 3.32.
3. THE TOOLING PROCESS FOR PROCESS DISCOVERY

All data displayed in the charts will now be updated to display only the previous 14 days which matches the amount of raw data in AC. We only need the time filter when we are retrieving session IDs for AC session analysis. For all other types of analysis make sure you remove this filter.

**Identify Power Users**

Power claims users can often be identified by the number of claims processed in the shortest amount of time. While that is not always true, it is a good starting point. To find the power users, create a chart showing user ids by count and user time in the following manner:

1. Right click on the main claims processing activity and select 'New Chart'
2. Fill in the information as follows
   a. Chart Name: Power Users
   b. Chart Type: Line/Bar
   c. Floating: Checked
   d. Click on 'Next'
3. Select UserID for the X-Axis attribute
4. Set Max Distinct Values to 50000
5. Click 'Next' (Figure 3.33)
3. THE TOOLING AND PROCESS FOR PROCESS DISCOVERY

6. Select DCN for the first Y-Axis attribute
7. Select count for the function
8. Click the 'ADD' button
9. Select peUserTime for the second Y-Axis attribute
10. Select sum for the function
11. Select Line - Right Y for the Series Type
12. Click on the 'ADD' button
13. Click on 'Finish' (Figure 3.34)
The result is a chart showing users by the count of claims worked and the amount of time used to work those claims (Figure 3.35).

While the chart is not very useful to look at, it contains the necessary information. The next steps are to export the data, filter user ids by the power user index, then import the filtered user ids to use as a filter in PIC. The steps below illustrate the method:

1. Right click on the chart and select Export -> Comma Separated Values…
2. Select a location to store the file
3. Open the file in Excel
4. You should see a file similar to the one below in Figure 3.36
5. Next, sort the data by DCN
6. Delete all rows where DCN is 0
7. Find a DCN cutoff number - the number of DCNs that really matter
   a. The number that matters will be dependent on the data
   b. This example uses 10
8. Delete all rows where DCN is lower than the cutoff number
9. Next, create the power ranking
   a. In the cell to the right of the user time create the formula
   b. \((B2/C2) \times 1000\)
10. Sort the data by the power index column
11. The result should be similar to the following output in Figure 3.37
12. Save the file with a different name for future use
13. Next, select the user ids from that file to use as a filter
   a. Select the user ids
   b. Select copy
   c. The user ids are now ready to be imported into PIC
14. The next step is to import the user ids into PIC so we can use them as a filter
   a. Select the 'Add an Or node and Attributes from a list' button
   b. The following dialog appears:
   c. Select User ID for the Search Attribute Figure 3.38
d. Click 'Next'
e. Paste the user ids into the import box (Figure 3.39)

![Figure 3.39 - Power User Filter Import](image)

f. Click 'Finish'

15. At this point, an OR filter with all of the user ids should exist

16. Click on the Or check box to select the filter
17. That completes building all filters needed to create the session id chart, described next

**Create Session ID Chart**

It is then necessary to create a chart to get a list of session IDs based on the filters set in previous steps. The following steps describe how to create the required chart:
• Right click on the main claims processing activity and select 'New Chart'
• Fill in the information as follows
  o Chart Name: Sessions by Session Count
  o Chart Type: Line/Bar
  o Chart Group: Edit Codes
  o Floating: Checked
  o Click on 'Next'
• Select peSessionID for the X-Axis attribute (Figures 3.39)
• Set Max Distinct Values to 50000
• Click 'Next'

![Create Chart](image)

**Figure 3.40 - X-Axis Attribute Definition for Session IDs**

• Select peSessionId for the Y-Axis Attribute (Figure 3.41)
• Click on 'Add'
3. THE TOOLING AND PROCESS FOR PROCESS DISCOVERY

• Click 'Finish'

That produces a chart of session ids based on the chosen filters.

Export Data from PIC

The chart of session ids are now exported for use in AC.

• Right click on the chart and select Export -> Comma Separated Values
• See a file dialog box appear
• Navigate to the desired folder, provide a file name and click on 'Save'

There is now a file with ids for sessions matching the criteria set in PIC. Next, import those session ids into AC.

Import Data into AC

Open the csv file created in 0. See a file similar to that listed below:
3. THE TOOLING PROCESS FOR PROCESS DISCOVERY

The file should have a column with session ids and a column with counts. Sort the data by count in descending order so that the session ids with the highest count are displayed at the top. That sets the most relevant sessions at the top of the list.

Next, copy the top session ids. A good rule of thumb for choosing the actual number of sessions to walk through in AC is to take the top 25 percent or the top 300 sessions, whichever is higher.

Open the Eclipse interface to AC and open the base report project.

Click on the View Sessions (Figure 3.43) button on the tool bar. Place the cursor in the text box and paste the session ids into the text box.
Click on the 'OK' button to get a prompt for the Security Key. Enter the key and click on the 'OK' button to view the sessions. If you do not have the security key, please contact the AC administrator.

**Analyze Sessions**

Analyzing sessions is an art and takes practice. The more one knows about the claims processing system, sub systems and the specific edit to be analyzed, the easier it is to analyze the sessions.

The first step is to select the first panel in the currently displayed session. Next, select the search button on the toolbar. Enter the search criteria, such as the edit code to be searched, and select String search. If that search does not return relevant information, the Regex search may be useful.

Walk through the user's session from this point on by clicking on the next panel name in the list on the right (Figure 3.44).
Keep track of the claim number to continue looking only at screens related to the same claim. Pay close attention to the AID key used to move to the next panel and the fields modified by the user. The modified fields are highlighted with a yellow box and appear in the second panel.

Walk through the session to find the end of the claim. Note the final outcome to working the claim (finalize, save, etc.). After determining that the claim was actually worked for the desired edit code, go back and take screen shots of the panels used in the workflow. Walk through as many sessions as possible and take note of the various paths the users took to work the claims, the lines of business, and the results of working the claims.

**Build Initial Requirements Document**

The Initial Requirements Document (IRD) has two purposes. First, it documents the various workflows used to work a specific edit code. Second, it determines the most accurate and efficient workflow to use as a standard for process automation or improvement.

The various workflows should be documented in an IRD. If any of the workflows are specific to a line of business that should be noted. Also take note of the results of each workflow and the data elements involved.

Walking user sessions helps to understand what the users do to work claims but does not provide the reasons that the user made certain decisions. For example, some workflows may show users entering an 'S' in a particular field and other workflows show the user entering an 'N'. It is easy to document those occurrences but still not know why the user chose an 'S' or an 'N'. It is a best
practice to make a note in the document about that situation. The SMEs can provide the business rules behind the users' choices later in the improvement process.

It is probable that there are not enough sessions nor time to see all of the potential values in a field. Take note of the fields that have multiple values so that additional analysis can be done in PIC to determine all of the values used. This analysis should also be provided in the IRD.

**Subject Matter Expert Sessions**

The next step in the process is to meet with SMEs for the specific edit code, or group of edit codes, and walk through the IRD. If the IRD provides different paths based on line of business or claim type, then the SMEs should be from the specified line of business or be familiar with the claim type.

Walk through the IRD and ask for specifics not fully explained by walking the sessions. Document the answers given by the SMEs and complete the requirements document.

**Build Business Requirements Document**

After working with the SME to understand the business rules it's time to create the BRD. The BRD should include the screen shots from the IRD and the business logic from the SME. The format of the BRD can be different for each customer but should contain an outline of the workflows. For example:

1. If line of business = Senior
   a. Enter an 'S' in the LOB field
2. Else
   a. Enter an 'E' in the LOB field
3. Press [Enter]

All 'IF' statements must be closed with an 'ELSE' statement. The idea is to make sure all logic paths are accounted for so the Configure developer can create "bullet-proof" transactions.

**Finalize Requirements Document**

Additional analysis might be required to help fill in the gaps. After all requirements are documented and it is believed to be complete, schedule a meeting with all participants to do a final walkthrough of the BRD. All lines of business associated with the specific edit code should be represented in this meeting.

Take notes and update any information appropriate based on team consensus. After all changes are made and everyone agrees on the contents, that constitutes a final version. Then, deliver the final BRD to the Business Approval Committee.

The BRD should provide a recommendation to either automate the process or improve the process. The recommendation should be based on the percentage of claims that can be worked by
a robot. That percentage can be determined by analyzing the various paths and determining if all
data is available to the robot through the standard methods (3270 screen, database access, web
service, properties file, etc).

**Business Approval**
The Business Approval Committee reviews the BRD to verify that it meets the needs of the
business. The Committee determines whether the process should be automated or improved and
documents its decision by signing the acceptance page.

If automation is chosen, the document is delivered to a process automation team. If the process is
to be improved, then the document is delivered to a process improvement team.

3.3.4 Build
In the Build stage, the process automation team creates Configure transactions based on the BRD
created in the Design stage of the improvement process.

**Design Transaction(s)**
The new transaction(s) should conform to the Transaction Framework developed specifically for
the customer by OpenConnect. Before the automation team can build any transaction, it is
necessary to design it. Follow the step below to design the transaction:

1. Review the BRD
   a. If there are any questions, send them to the owner of the BRD
   b. If there are still questions, ask for sample claims that meet the requirements in
      order to personally walk through the claim to verify the business rules.
2. Determine how the new transaction will interact with other transactions already available
   in the Transaction Framework
3. Determine how to create the Panel Handler
   a. Panel Handler behavior will depend on the panels with which the new transaction
      will interact
   b. It is also necessary to plan how to handle unknown panels
4. Determine whether Perform or Set-state will be used to call other areas of the new
   transaction

3.3.4.1 Build Transaction(s)
The new transaction(s) will be created using the OpenConnect Configure Development
Environment. Open Configure and create your new transaction. Develop the transaction based on the
BRD and the considerations in the Design Transaction phase.
3. THE TOOLING AND PROCESS FOR PROCESS DISCOVERY

3.3.4.2 Unit Test

All unit testing is performed within the Configure development environment. Use the personally created test cases or those created by others. The test cases should match the various scenarios outlined in the BRD.

3.3.4.3 Integration Test with Web Application

After all test cases have been successfully executed and validated, deploy to the development environment to test with the web application. The steps for deploying are shown below:

1. Click on the Deployment Wizard button on the toolbar
2. Click 'Next'
3. Accept the default package name or create a new one
4. Click 'Next'
   a. This will generate the source files for the project
5. Click 'Next'
   a. This will compile the source files
6. Accept the version number displayed, which should be one greater than the current version, or create a new version number greater than the version displayed
7. Enter a meaningful description of this version
8. Click 'Finish'

3.3.4.4 System Test

Before system tests can be executed test claims have to be available. Make sure there are enough test claims to test all of the various transaction paths and scenarios. Configure the web application to execute the test claims. This could mean pointing it to the correct mainframe environment or it could be entering a search string.

Once the claims are available, execute the transactions through the web application. That execution depends on the environment, customer and web application. After execution, run the web application report to validate the results.

If the environment supports the setVisible feature of Connect, make sure to have setVisible set to 'true'. This will allow observation of the emulated 3270 screen as claims are executed.

3.3.4.5 User Acceptance Testing

After all tests are successfully executed it is time to send the project for user acceptance testing. That may require deploying to a different environment (database) or marking the new version available to the UAT environment depending on the configuration.

The most important considerations for UAT are the test claims and the expected results. That should come from the UAT team. Once the test claims become available it is possible to execute the web application pointing to the UAT environment.

As before, create a report from the web application and provide this report to the UAT team for validation. If there are claims that did not get the expected result a ticket will be opened and it will be necessary to work with the UAT team to determine the root cause.

3.3.5 Execute

The Execute stage is used to run the web application in production after User Acceptance Testing is complete and all tests have successfully executed.

3.3.5.1 Deploy Configure Project to Production Environment

The version of the Configure Project that was tested in UAT should now be deployed to the production environment. This could mean to physically deploy to another server or it could mean to just mark the version as current for the production environment depending on the configuration.

Also, depending on the environment, it may be desirable to disable the web application before making this change. In some environments that may require synchronizing the deployment with the mainframe parameters used to provide claims to the web application.

[135]
3.3.5.2 Execute New Edit Code

Add the new edit code to the web application so that claims with this edit code can be worked. The required parameters will depend on the environment and the web application.

If the web application is disabled, make sure all synchronization issues are resolved before enabling it.

3.3.5.3 Run Reports

Depending on the environment one can run reports either at the end of a run, or at any time before or during a run. OpenConnect recommends creating a schedule, having the web application automatically create the report and email it to the defined recipient(s).

3.3.6 Monitor and Measure

The Monitor and Measure stage is used for two purposes. The first is to monitor the execution of the web application and Connect transaction to make sure they are working properly. The second is to measure the solution’s success rate.

3.3.6.1 Monitor

Monitor refers to observing the automated processes to determine if they are executing properly. This is an important step in the improvement process because the underlying business process changes over time. This is called process drift and is caused when business rules change, new functions are required or new data is entered, to name a few examples. If the automated process is not continuously monitored then errors can occur which could result in a large negative financial impact to the organization.

3.3.6.2 Create Attributes

Before it is possible to monitor the automated processes it is necessary to identify which attributes to use. To monitor the web application and Connect transactions it is necessary to ensure that the following attributes are defined:

1. User ID
2. Edit Code(s)
3. Success (this can be a finalized flag or a transition to a finalized activity)
4. Adjustments (this is usually defined as a separate activity)

Sometimes the proper execution of a transaction requires access to specific functions. Those functions can usually be provided either by an attribute on a claim screen or by another activity. If the information can be provided through an attribute then there is no need to create the activity. However, it may be required to create the activity to determine if the transaction followed the correct procedure.
3. THE TOOLING PROCESS FOR PROCESS DISCOVERY

3.3.6.3 Configure and Monitor

Create a perspective in PIC by Claim Number (DCN). This will expose the paths taken for each claim as it is executed. Create a View from this perspective and give the view name of 'Monitor'.

Add the filters for the appropriate attributes. For example, add the filter for the user id(s) used by the Configure transactions. Also add a filter for the specific edit code to be monitored. The method is to create filters for each of the edit codes then only apply one filter at a time.

After all of the filters are applied, PIC shows the paths taken by the automated process for the filtered edit code. The paths will help determine if the transaction is following the correct procedures. In addition to the paths it is desirable to create charts to help understand how the transactions are running. These charts are edit code specific but at a minimum create the following charts:

1. DCN by User Time
   a. This chart helps determine how long a process executes for a specific edit code
   b. Create the baseline right after deploying the edit code to production
   c. Monitor the execution time and keep track to determine trends
   d. If the execution time increases or decreases significantly over a short period of time there may be a problem with the transaction

2. Edit Code by Count
   a. Create this chart and watch the various paths in the process map
   b. Keep track of the edit code distribution for each path
   c. If there is a significant increase or decrease of a specific edit code for specific paths over a short period of time, there may be an issue with the transactions for the specific edit codes

3.3.6.4 Measure

Measure refers to the process to measure the success of the automated processes and to determine how to improve the success rate.

3.3.6.5 Create Attributes

Measuring the success of the automated processes can be accomplished by combining the reports provided by the web application and PIC charts. For PIC charts, the following attributes are required:

1. User ID
2. Edit Code(s)
3.3.6.6 Configure and Measure

To measure the success of the automated processes it is best to use the reports provided by the web application. Each web application is specific to the customer and environment. Refer to web application-specific documentation to determine how to create and read the reports provided by that web application.

PIC can be used to help determine how to improve the execution of the automated process. If, after reviewing the reports from the web application, the success rate for a particular edit code is low because there are other edit codes on the claim that prevent the claim from finalizing, use PIC to help.

First, learn what those other edit codes are. Then, determine if those edit codes can be automated. Then, determine what the success rate can be if they are automated. Finally, create the requirements documents so that they can be automated.

Described above is the improvement process outlined in this document. The difference is that it will be necessary to have the User ID and Edit Code filters defined and selected before starting the process.

3.3.7 Summary

This document described an overall process and a set of detailed steps that are flexible enough for each organization to fit it to their environment and situation as necessary. Each stage can be considered a separate sub-process and can be enhanced to provide results needed for any particular organization.

Experience shows that following the process leads to quantifiable, sustainable improvements to claims processing, including cost savings.

In this chapter a fairly complete example of a toolset that automates the major steps of the C-K-theory based process. The collectors and analytics components build the data in the required Event-Actor-Object format as input for Process Discovery. The analytics components provide the first clustering off coherent event streams into services, to simplify and reduce the input data. A major role is played by the Process Intelligence Cluster which automates the disjunction and the actual analysis of the process conceptual representation (in terms of Hidden Markov Models). An extension of PIC with Formal Concept Analysis is scheduled in subsequent versions of Comprehend in 2013 – 2014. Finally, Comprehend also provides support for conjunction and activation of insights in value leaks and process improvements, by means of its possibilities for simulation and user replay of processes.