Chapter 14
Analyzing "Spaghetti Processes"

Spaghetti processes are the counterpart of Lasagna processes. Because Spaghetti processes are less structured, only a subset of the process mining techniques described in this book are applicable. For instance, it makes no sense to aim at operational support activities if there is too much variability. Nevertheless, process mining can help to realize dramatic process improvements by uncovering key problems.

14.1 Characterization of "Spaghetti Processes"

As explained in the previous chapter, there is a continuum of processes ranging from highly structured processes (Lasagna processes) to unstructured processes (Spaghetti processes). In this chapter we focus on Spaghetti processes.

Figure 14.1 shows why unstructured processes are called Spaghetti processes. Only when zooming in one can see individual activities. Figure 14.2 shows a tiny fragment of the whole process. The fragment shows that activity "O_Bloedkweek 1" (a particular blood test) was scheduled 412 times and 230 times followed by "O_Bloedkweek 2" (another test). These activities are frequent. However, there are also several activities that are executed for only one of the 2765 patients.

The process model depicted in Fig. 14.1 was obtained using the heuristic miner with default settings. Hence, low frequent behavior has been filtered out. Nevertheless, the model is too difficult to comprehend. Note that this is not necessarily a problem of the discovery algorithm. Activities are only connected if they frequently followed one another in the event log (cf. Sect. 7.2). Hence, the complexity shown in Fig. 14.1 reflects reality and is not caused by the discovery algorithm!

Figure 14.1 is an extreme example used to explain the characteristics of a Spaghetti process. Given the data set it is not surprising that the process is unstructured; the 2765 patients did not form a homogeneous group and included individuals with very different medical problems. The process model can be simplified dramatically by selecting a group of patients with similar problems. However, also for more homogeneous groups of patients (e.g., people that had heart surgery), the resulting process model is often Spaghetti-like.

W. van der Aalst, *Process Mining*, DOI 10.1007/978-3-662-49851-4_14

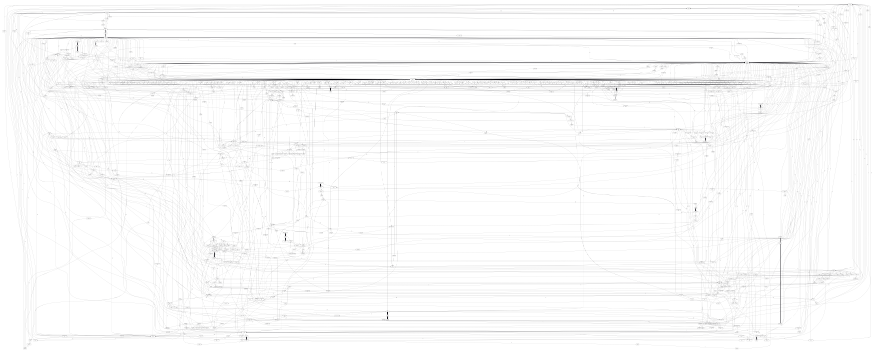


Fig. 14.1 Spaghetti process describing the diagnosis and treatment of 2765 patients in a Dutch hospital. The process model was constructed based on an event log containing 114,592 events. There are 619 different activities (taking event types into account) executed by 266 different individuals (doctors, nurses, etc.)

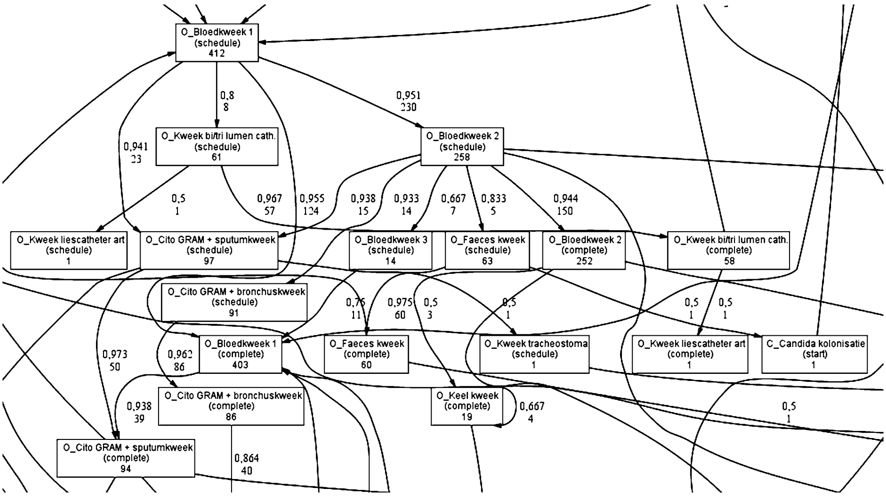


Fig. 14.2 Fragment of the Spaghetti process of Fig. 14.1 showing 18 activities of the 619 activities (2.9%)

Let us consider another, less extreme, example. Figure 14.3 shows the dotted chart for a process of one of the largest Dutch housing agencies (see also Figs. 9.3 and 9.4). Each case corresponds to a housing unit (accommodation such as a house or an apartment). The process starts when the tenant leasing the unit wants to stop renting it. The process ends when a new tenant moves into the unit after addressing all formalizaties. In-between, activities such as “registering the new address”, “first inspection”, “final inspection”, “finalize contract”, “return deposit”, “sign contract”, “repair”, and “update price” are executed. Figure 14.3 is based on an event log containing information about 208 units that changed tenant. There are 74 different

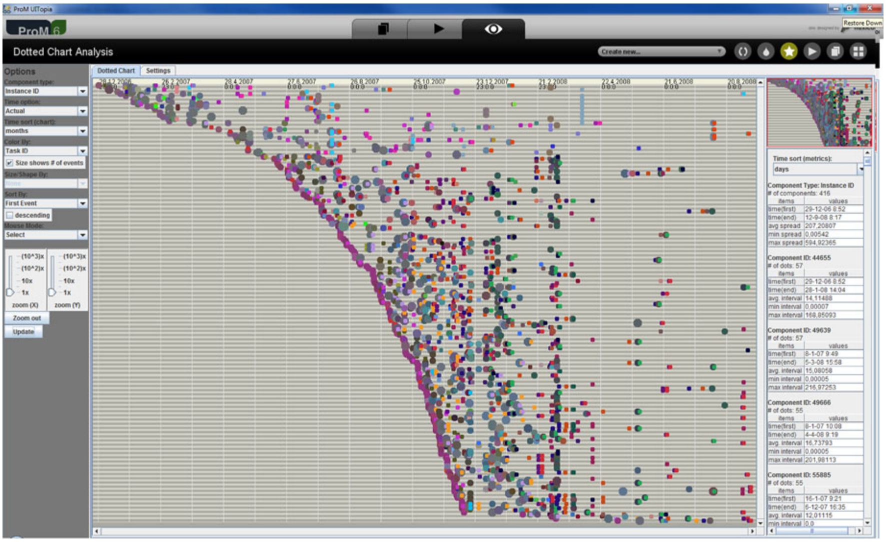


Fig. 14.3 Dotted chart created using an event log of a Dutch housing agency. Each line corresponds to a case (house or apartment). The event log contains 208 cases that generated 5987 events. There are 74 different activities

activities. In total 5987 activities were executed for the 208 units. As Fig. 14.3 shows there is a huge variance in flow time. For some units it takes a very long time to change ownership (sometimes more than a year) for others this is matter of days. The initial events of the 208 cases do not form a straight line; the curve shows that the arrival rate of new cases is increasing during the period covered by the event log.

Figure 14.4 shows a process model discovered using the heuristic miner. Although the model does not look as Spaghetti-like as Fig. 14.1, it is rather complicated considering the fact that it is based on only 208 cases. The 208 cases generate 203 unique traces, i.e., almost all cases follow a path that is not followed by any of the other cases. This observation, combined with the complexity of the model suggests that the log is far from complete thus complicating analysis.

The processes of the Dutch hospital and housing agency illustrate the challenges one is facing when dealing with Spaghetti processes. Nevertheless, such processes are very interesting from the viewpoint of process mining as they often allow for various improvements. A highly-structured well-organized process is often less interesting in this respect; it is easy to apply process mining techniques but there is also little improvement potential. Therefore, one should not shy away from Spaghetti processes as these are often appealing from a process management perspective. *Turning Spaghetti processes into Lasagna processes can be very beneficial for an organization.*

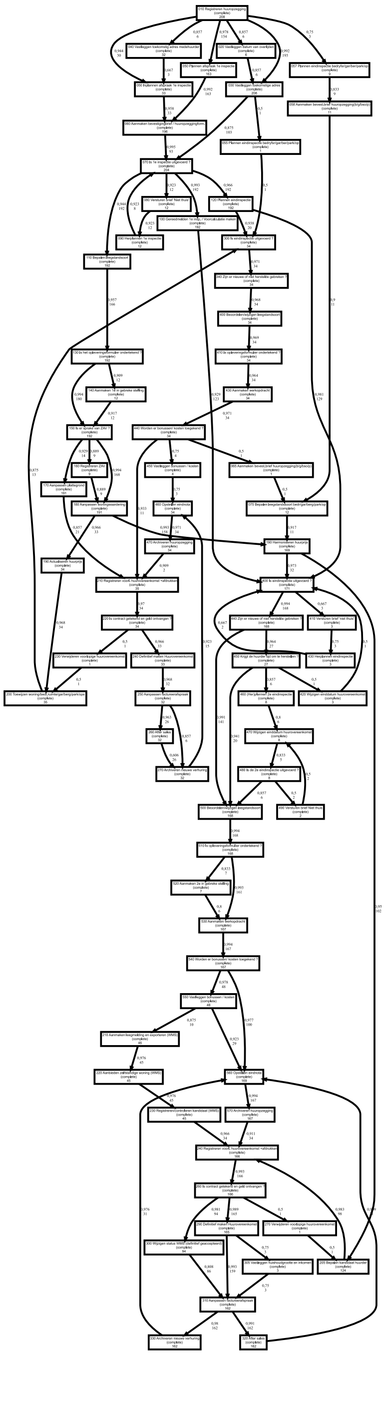


Fig. 14.4 C-net for the event log of the housing agency. The model was obtained using the heuristic miner (with default settings). The model was discovered based on an event log with 5987 events. All 208 cases start with activity "010 Registreren huuropzegging" (register request to end lease). Some of the activities are relatively infrequent, e.g., activity "020 Vastleggen datum van overlijden" occurred only 6 times (this activity is only executed if the tenant died)

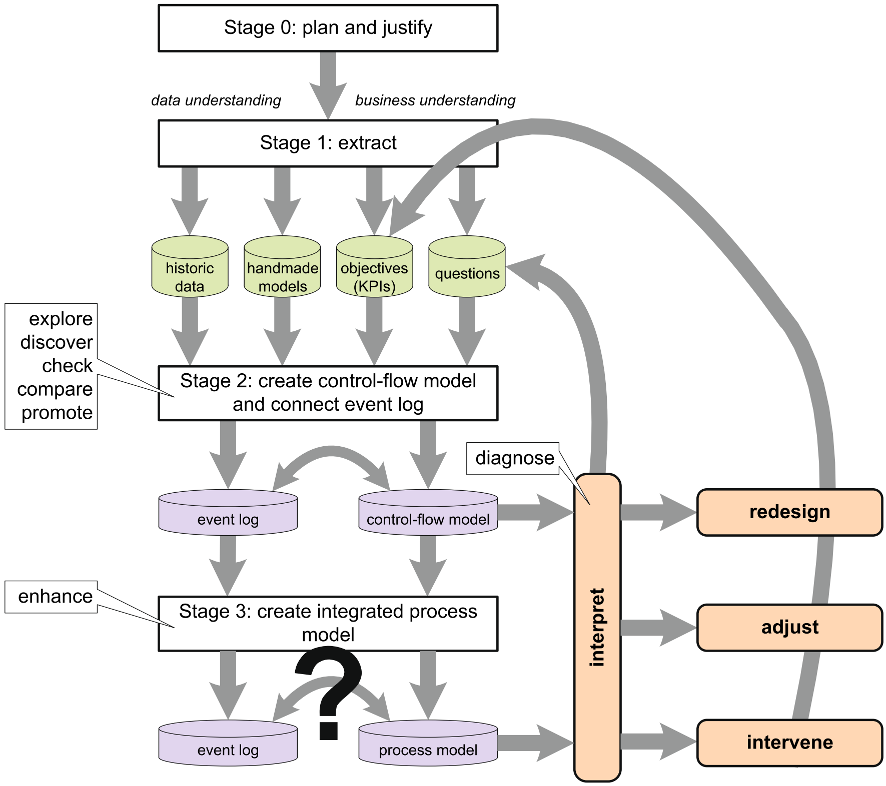


Fig. 14.5 The part of the L^* life-cycle model applicable to Spaghetti processes: stages 0, 1 and 2 are also possible for unstructured processes. However, creating an integrated process model covering all perspectives (Stage 3) is often not possible. Instead separate models are generated for the other perspectives, e.g., a social network

14.2 Approach

In Sect. 13.3 we introduced the L^* life-cycle model describing an idealized process mining project aiming at improving a Lasagna process. Only the initial stages are applicable for Spaghetti processes. Figure 14.5 shows the most relevant part of the L^* life-cycle model. Note that Stage 4 has been removed because operational support is impossible for the processes just described. To enable history-based predictions and recommendations it is essential to first make the "Spaghetti-like" process more "Lasagna-like". In fact, Stage 3 will also be too ambitious for most Spaghetti processes. It is always possible to generate process models as shown in Figs. 14.1 and 14.4 (Stage 2). Moreover, it is also possible to view dotted charts, create social networks, etc. However, it is very unlikely that all of these can be folded into a meaningful comprehensive process model as the basis (the control-flow discovered) is too weak.

In Sect. 6.4 we discussed the challenges related to process mining. They are of particular relevance when dealing with Spaghetti processes. Event logs do not contain negative examples, i.e., only positive example behavior is given. The fact that something does not happen in an event log does not mean that it cannot happen. For example, Fig. 14.4 is based on an event log in which almost all cases follow a unique path (the 208 cases generate 203 different traces). Therefore, the discovery algorithm needs to generalize. For more complex processes, i.e., processes that are large and that allow for many behaviors, the event log is typically far from complete (cf. Sect. 6.4.2). To further complicate matters, there may be noisy behavior, i.e., infrequent behavior that the user is not interested in. Because of these complications, a discovery algorithm needs to carefully balance the four quality dimensions introduced earlier: *fitness*, *simplicity*, *precision*, and *generalization* (see Fig. 6.22). The process models shown in Figs. 14.1 and 14.4 illustrate the relevance of these considerations. For the characteristics of the different process discovery algorithms we refer to Part III of this book. Here, we only stress the importance of carefully *filtering* the event log before discovery.

Let us first consider the *filtering of activities* based on their characteristics, e.g., absolute or relative frequency. Figure 14.6(a) shows a filtering plug-in selecting all activities that occurred in at least 5% of all cases. This ProM 5.2 plug-in is applied to the event log used to construct Fig. 14.1, i.e., activities that do not appear frequently are removed from the event log. As a result, the process model will be simpler as fewer activities are included. Figure 14.6(b) shows a filtering plug-in in ProM 6 applied to the event log used to construct Fig. 14.4. In this case the top 80% of activities are included; all other activities are removed from the log. The effect of filtering is shown in Fig. 14.6(c). This C-net was obtained by selecting all activities that occur in at least 50% of all cases handled by the housing agency. A comparison of the process model obtained using the original event log (Fig. 14.4) with the process model obtained using the filtered event log (Fig. 14.6(c)), demonstrates the effect of filtering. The discovered model shows only 28 of the 74 activities appearing in the event log of the housing agency.

In principle, any model *can be made as simple as desired* by simply abstracting from infrequent activities. In the extreme case, the model contains only the most frequent activity. Such a model is not very useful. However, it shows that filtering can be used to seamlessly simplify models. Interestingly, it is sometimes useful to also abstract from very frequent activities that are interleaved with other activities (e.g., some system action executed after every update). These clutter the diagram while being less relevant. Note that there may be multiple criteria for selecting/removing activities (e.g., average costs, duration, and risks).

Besides the simple activity-based filtering illustrated by Fig. 14.6 there are more advanced types of filtering that transform low-level patterns into activities [77]. Moreover, the cases in the log can be partitioned in homogeneous groups as shown in [13, 62, 78]. The basic idea is that one *does not try to make one large and complex model for all cases, but simpler models for selected groups of cases*. Here, one can use the classical clustering techniques described in Sect. 4.3 and adapt them for process mining. To apply these techniques, feature extraction is needed to describe

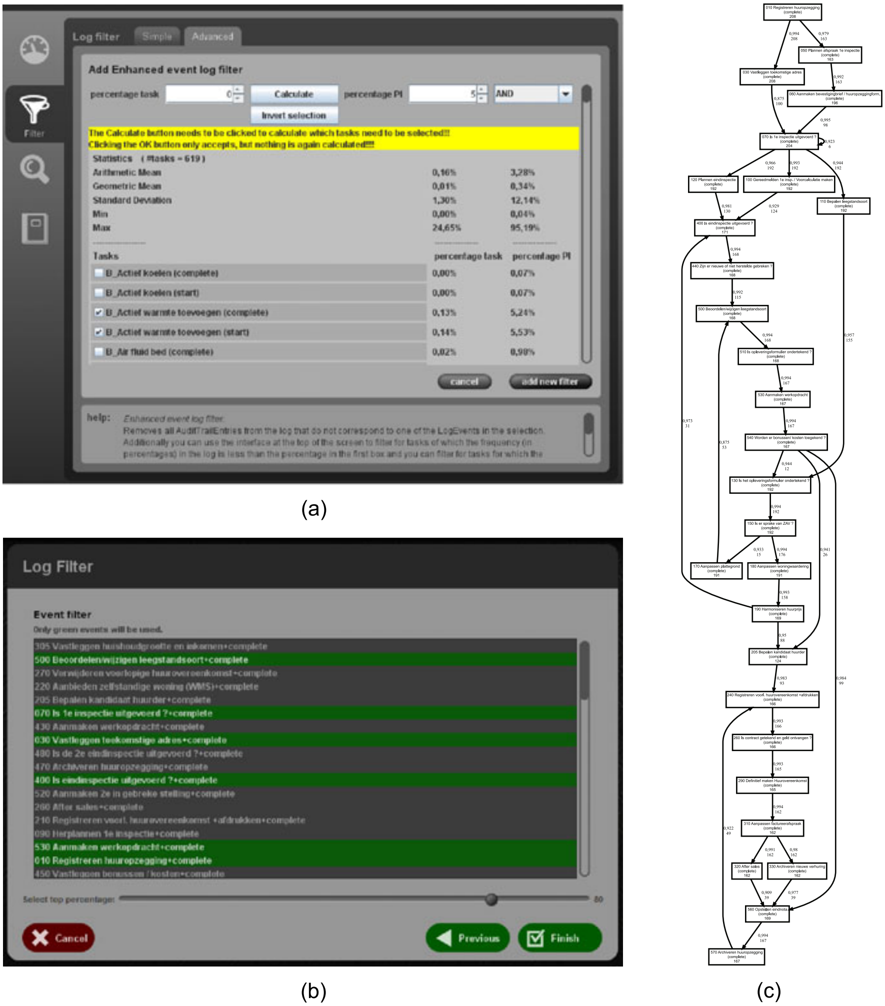


Fig. 14.6 Filtering the event log before process discovery: (**a**) selecting activities that occur for at least 5% of all 2765 patients, (**b**) selecting the top 80% of the 74 activities conducted by employees of the housing agency, (**c**) C-net discovered based on a filtered log (the event log of the housing agency after removing the activities occurring for less than 50% of the units)

cases in terms of a vector of variables (the features). By using a hierarchical clustering technique as shown in Fig. 14.7, one can view the same process at multiple levels. Cutting the dendrogram close to the root results in a few more complex models. Cutting the dendrogram closer to the leaves of the tree results in many simple models.

In the next chapter, we describe an alternative way to simplify process models. In contrast to filtering, simplification and abstraction techniques are directly applied to the process graph. This so-called *fuzzy mining* approach views process models

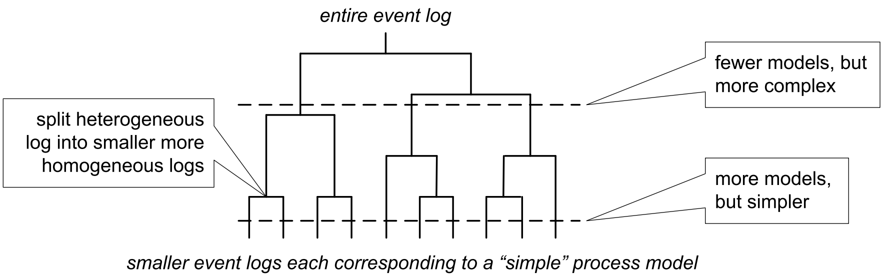


Fig. 14.7 Hierarchical clustering applied to heterogeneous event logs. The whole event log is partitioned into smaller, more homogeneous, event logs. This process is repeated until it is possible to create a "simple model" for each of the smaller logs. The resulting dendrogram can be cut closer to the root or closer to the leaves. This reflects the trade-off between the simplicity of models and the number of models

as if they are geographic maps (e.g., road maps or hiking maps). Depending on the map, insignificant roads and cities can be removed and streets and suburbs can be amalgamated into bigger structures. Figure 14.8 shows the effect this approach on the event log of the housing agency (i.e., the log used to construct the model in Fig. 14.4). Section 15.1.3 will elaborate further on the cartography metaphor used by the fuzzy mining approach.

14.3 Applications

In the previous chapter, we provided a systematic overview of the different sectors, industries, and functional areas where process mining can be used. In this section, we briefly revisit this overview for Spaghetti processes. Moreover, we give some pointers to case studies describing the analysis of highly unstructured processes.

14.3.1 Process Mining Opportunities for Spaghetti Processes

Many of the use cases presented in Sect. 13.2 also apply to Spaghetti processes. However, the "stakes are higher"; it will take more time to thoroughly analyze the process, but the potential gains are typically also more substantial.

Figure 14.9 highlights the functional areas where typically Spaghetti processes can be found.

Processes in *Product development* tend to be rather unstructured because they are low frequent (compared to production processes) and rely on creativity and problem-solving capabilities. For example, we have been mining event logs from Software Configuration Management (SCM) systems such as CVS and Subversion. In addition to managing the artifacts created by software engineers, these systems also collect and store information on the software development process to answer

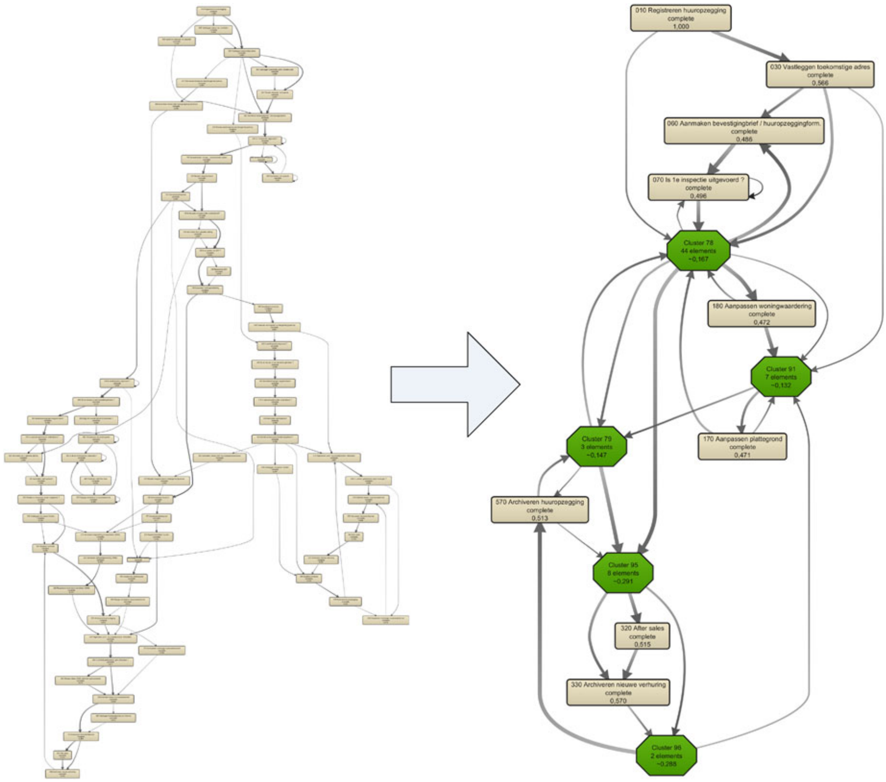
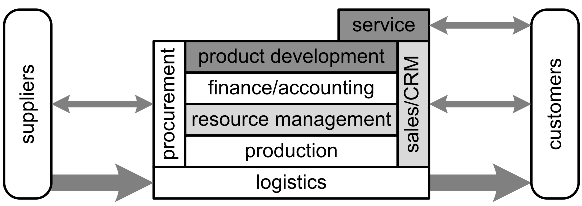


Fig. 14.8 Fuzzy mining applied to the event log of the housing agency. The cartography metaphor is used to support seamless abstraction and generalization. Both models provide a view on the same process. In the right model infrequent activities have been removed or amalgamated into cluster nodes. Moreover, infrequent arcs are removed based on the selected threshold

Fig. 14.9 Overview of the different functional areas in a typical organization. Spaghetti processes are typically encountered in product development, service, resource management, and sales/CRM



questions such as "Who created, accessed, or changed which documents?", "When was a particular task completed?", etc. Process discovery efforts using the event logs of SCM systems as input typically reveal Spaghetti-like processes as shown before.

Figure 14.9 indicates that one can also find Spaghetti processes in the functional area *Service*. An interesting development is that more and more products are monitored while being used in their natural habitat, e.g., modern high-end copiers, expensive medical devices, and critical production facilities collect event logs and can be observed remotely. Later, we will show that ASML and Philips Healthcare already monitor the systems they manufacture. In the future, manufacturers will start monitoring also less expensive goods, e.g., cars, consumer electronics, and heating systems will be connected to the Internet for a variety of reasons. Manufacturers would like to know how their products are used, when they malfunction, and how to repair them.

Resource management and *Sales/CRM* are two functional areas where a mixture of Spaghetti and Lasagna processes can be encountered (cf. Sect. 13.4.1).

One can come across Spaghetti processes in all sectors and industries mentioned in Sect. 13.4.2. However, processes in the tertiary sector tend to be less structured than processes in the other two sectors. For instance, as is illustrated by Fig. 14.1, the healthcare industry is notorious in this respect. In general one can say that processes driven by humans that can operate in an autonomous manner are less structured. Situations, in which expertise, intuition, and creativity are important, stimulate self-government. Doctors in hospitals and engineers in large construction projects often need to deal with one-of-a-kind problems. Consumers that are using products also operate in an autonomous manner. Consider, for example, a television that can be monitored remotely to learn how it is used and when it malfunctions. Some users will watch television the whole day and constantly switch channels whereas other users only watch the news at 8 pm and then switch off the television. Self-directed behavior of consumers and professionals typically results in Spaghetti-like processes.

As mentioned earlier, *Spaghetti processes are interesting from the viewpoint of process mining*. First of all, it is interesting to learn from the amazing capabilities of humans to deal with complex unstructured problems. When automating parts of the process it is important to understand why processes are unstructured to avoid building counter-productive and inflexible information systems. Second, Spaghetti processes have the largest improvement potential. They are more difficult to analyze, but the prospective rewards are also higher.

14.3.2 Examples of Spaghetti Processes

We have encountered Spaghetti processes in a variety of organizations. In Chap. 13, we already mentioned several organizations where we applied process mining. In this section, we give three additional examples: ASML, Philips Healthcare, and AMC. The goal is not to describe the processes of these organizations in detail, but to provide pointers to applications of process mining in Spaghetti-like environments.

14.3.2.1 ASML

ASML is the world's leading manufacturer of chip-making equipment and a key supplier to the chip industry. ASML designs, develops, integrates and services advanced systems to produce semiconductors. Process mining has been used to analyze the test process of wafer scanners in ASML [123].

Wafer scanners are complex machines consisting of many building blocks. They use a photographic process to image nanometric circuit patterns onto a silicon wafer. Because of competition and fast innovation, the time-to-market is very important and every new generation of wafer scanners is balancing on the border of what is technologically possible. As a result, the testing of manufactured wafer scanners is an important, but also time-consuming, process. Every wafer scanner is tested in the factory of ASML. When it passes all tests, the wafer scanner is disassembled and shipped to the customer where the system is re-assembled. At the customer's site, the wafer scanner is tested again. Testing is time-consuming and takes several weeks on both sites. Since time-to-market is very important, ASML is constantly looking for ways to reduce the time needed to test wafer scanners.

Figure 14.10 shows that the testing of wafer scanners is indeed a Spaghetti process [123]. The model was discovered based on an event log containing 154,966 events. The event log contained information about 24 carefully chosen wafer scanners (same type, same circumstances, and having complete logs). The number of events per case (i.e., the length of the executed test sequence) in this event log ranges from 2820 to 16250 events. There are 360 different activities, all identified by four-letter test codes. Each instance of these 360 activities has a start event and complete event. Figure 14.10 is based on just the complete events.

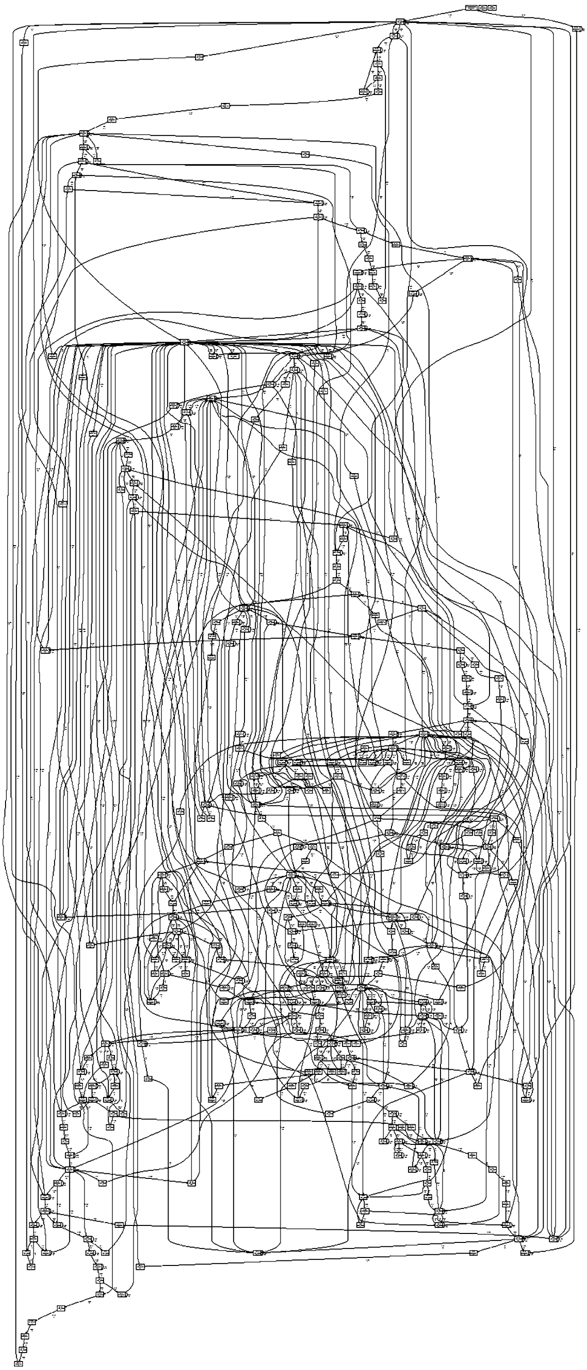
ASML also had a so-called reference model describing the way that machines should be tested. This reference model is at the level of job steps rather than test codes. However, ASML maintains a mapping from the lower level codes to these higher level activities. Comparing the reference model and our discovered model (both at the job step and test code level) revealed interesting differences. Moreover, using ProM's conformance checker we could show that the average fitness was only $\mathit{fitness}(L, N) = 0.375$, i.e., less than half of the events can be explained by the model (Sect. 8.2). When replaying, we discovered many activities that had occurred but that should not have happened according to the reference model and activities that should have happened but did not.

Both the discovered process models and the results of conformance checking showed that process mining can provide new insights that can be used to improve the management of complex Spaghetti-like processes. We refer to [123] for more details.

14.3.2.2 Philips Healthcare

Philips Healthcare is one of the leading manufacturers of medical devices, offering diagnosing imaging systems, healthcare information technology solutions, patient monitoring systems, and cardiac devices. Like ASML, Philips Healthcare is developing complex high-tech machines that record massive amounts of events. Since 2007 there has been an ongoing effort to analyze the event logs of these machines using process mining.

Fig. 14.10 Process model discovered for ASML's test process



Philips Remote Services (PRS) is a system for the active monitoring of systems via the Internet. PRS has been established to deliver remote technical support, monitoring, diagnostics, application assistance, and other added value services. Low level events (e.g., pushing a button, changing the dosage) are recorded by the machine and subsequently sent to Philips via PRS. Using the Remote Analysis, Diagnostics And Reporting (RADAR) system, event logs are converted into an XML format and stored in the internal database of RADAR. Subsequently the collected event data are translated into MXML files to enable process mining.

Process mining has been applied extensively to the event logs generated by Allura Xper systems. These are X-ray systems designed to diagnose and possibly assist in the treatment of all kinds of diseases, like heart or lung diseases, by generating images of the internal body. These systems record three types of events:

- *User messages.* When a message is shown to the user (e.g., "Geometry restarting") this is recorded in the event log.
- *Commands.* Both users and system components can invoke commands. These are all recorded. Commands typically have various parameters (e.g., voltage values).
- *Warnings and errors.* Whenever a problem occurs (or is anticipated) an event is recorded.

Each event has a timestamp and contains information about the component that generated the event.

It is possible to analyze the processes in Allura Xper systems from various angles. The concept of a "case" (i.e., process instance) may refer to a machine, a machine day, the execution of a particular procedure, the repair of a machine, etc. Figure 14.11 shows an example taken from [67]. Processes discovered for these systems tend to be Spaghetti-like. To simplify diagnosis, the log is often preprocessed as discussed in [77–79]. Moreover, fuzzy mining, as illustrated by Fig. 14.8, is used to further simplify the model [67].

Mining processes from the event logs generated by Allura Xper systems is very challenging. The machines consist of many components and can be used in many different ways. Moreover, logging is rather low-level and changes with every new version. Nevertheless, there are various opportunities for process and system improvements using process mining. These are listed below. Note that opportunities also apply to other types of systems that are monitored remotely.

- Process mining provides *insight* into how systems are actually used. This is interesting from a *marketing* point of view. For example, if a feature is rarely used, then this may trigger additional after sales activities. It is also possible that, based on process mining results, the feature is removed or adapted in future systems.
- *Testing* can be improved by constructing test scenarios based on the actual use of the machines. For instance, for medical equipment it is essential to prove that the system was tested under realistic circumstances.
- Process mining can be used to improve the *reliability* of next generations of systems. Better systems can be designed by understanding why and when systems malfunction.

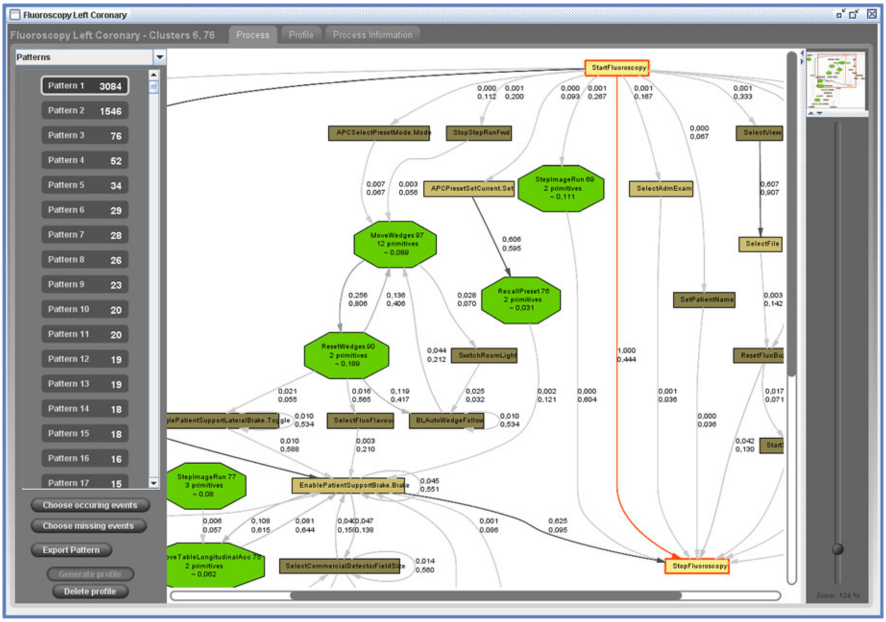


Fig. 14.11 Screenshot of a discovered process model for fluoroscopy runs in the context of the so-called "left coronary procedure" inside Allura Xper systems distributed all over the globe

- Process mining can also be used for *fault diagnosis*. By learning from earlier problems, it is possible to find the root cause for new problems that emerge. For example, we have analyzed under which circumstances particular components are replaced. This resulted in a set of *signatures*. When a malfunctioning X-ray machine exhibits a particular "signature" behavior, the service engineer knows what component to replace.
- Historic information can also be used to *predict* future problems. For instance, it is possible to anticipate that an X-ray tube is about to fail. Hence, the tube can be replaced before the machine starts to malfunction.

These examples show the potential of remote diagnostics based on process mining.

14.3.2.3 AMC Hospital

Hospitals are particularly interesting from a process mining point of view. By law, hospitals need to record more and more data in a systematic manner and all event data are connected to patients. Therefore, it is relatively straightforward to correlate events. For example, by Dutch law all hospitals need to record the diagnostic and treatment steps at the level of individual patients in order to receive payments. This so-called "Diagnose Behandeling Combinatie" (DBC) forces Dutch hospitals to record all kinds of events. There is also consensus that processes in hospitals

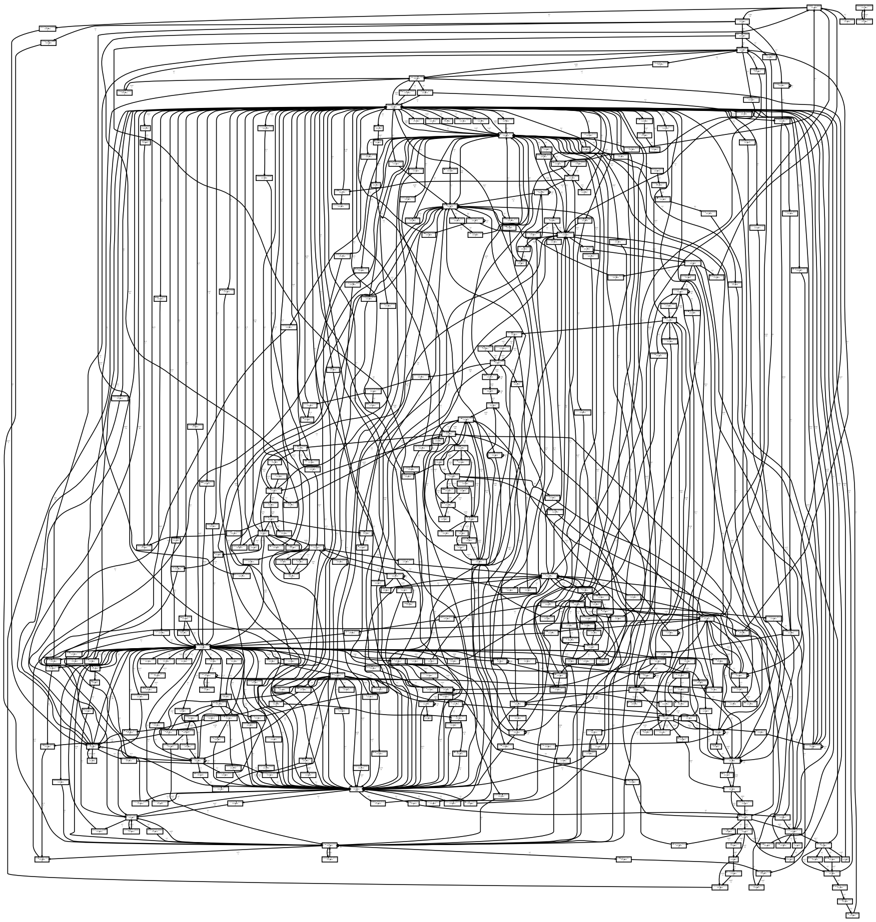


Fig. 14.12 Another Spaghetti process. The model is based on a group of 627 gynecological oncology patients. The event log contains 24331 events referring to 376 different activities

can be improved. Unlike most other domains, operational care processes are not tightly controlled by management. This, combined with the intrinsic variability of care processes, results in Spaghetti.

Some think that care processes in hospitals can be improved by simple principles from operations management or by introducing workflow technology. Process models such as the one shown in Fig. 14.1 demonstrate that this is not case. One needs to better understand the variability, before suggesting solutions.

We conducted several process mining experiments based on event data of the AMC hospital in Amsterdam [95]. Together with people of the AMC we have been investigating the introduction of workflow technology in this large academic hospital. This revealed many limitations of existing WFM/BPM systems when it comes to care processes. The variability in these processes is larger than in most other do-

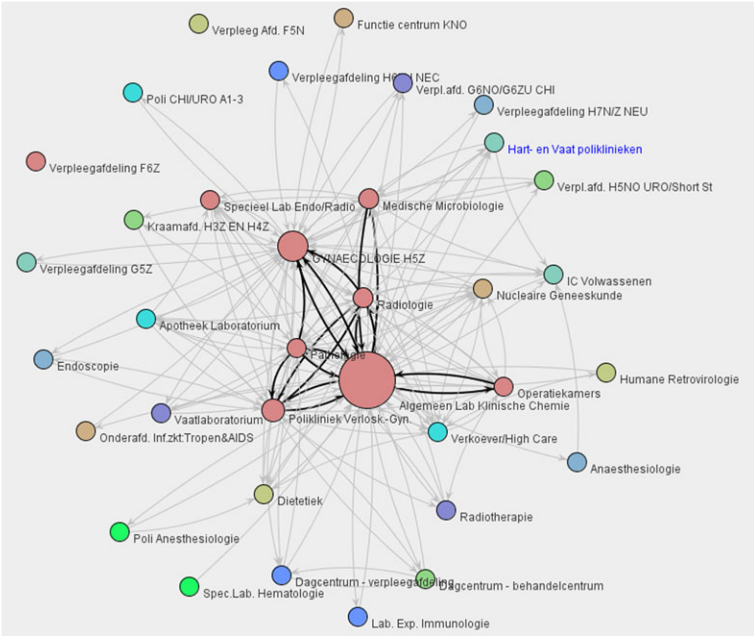


Fig. 14.13 Social network showing handovers between different organizational units of the AMC hospital

mains. This imposes unique requirements with respect to flexibility. Moreover, care processes combine flow oriented tasks with scheduled tasks [96]. As a result, conventional workflow technology is not applicable and a better understanding of the processes is needed.

Figure 14.12 shows an example of a process model constructed for the AMC hospital. The model was discovered based on event data of a group of 627 gynecological oncology patients treated in 2005 and 2006. All diagnostic and treatment activities have been recorded for these patients. Clearly, this is a Spaghetti process. However, as shown in [95] it is possible to create simple models for homogeneous groups of patients using the hierarchical clustering technique illustrated by Fig. 14.7. The same event log also contained information about resources. For instance, Fig. 14.13 shows a social network based on this log. As in earlier examples, the social network is based on handovers of work. However, now we do not look at individuals but at the level of organizational units. Figure 14.13 can be used to analyze the flow of work between different departments of the AMC hospital. For example, the social network reveals that most handovers take place between the gynecology department and the general clinical lab.

Experiences with process mining in several hospitals revealed important challenges when applying this new technology. The databases of hospitals contain lots of event data. Since any event can be linked to a patient, correlation is easy. However, for many events only the date (“31-12-2010”) is known and not the exact timestamp

("31-12-2010:11.52"). Therefore, it may be impossible to deduce the order in which events took place. Another problem is related to the trade-off illustrated by the dendrogram in Fig. 14.7. The process model for a large group of patients is typically Spaghetti-like as illustrated by Fig. 14.12. It is possible to create simpler models by looking at smaller homogeneous groups of patients. However, the drawback is that often the number of cases per group gets rather small. If there are only few cases in such a homogeneous group, the result is not very reliable. Only for homogeneous groups with more cases, the result is more trustworthy.

Despite these challenges, process mining provides a "mirror" for managers, doctors, and IT specialists in hospitals. To improve care-flows and to provide better IT support, it is essential to face the inherent complexity of these Spaghetti processes.