A Brief Introduction to Workflow Mapping and Analysis

excerpted from

Managing Knowledge and Mapping Workflows,
Chapter 2. Workflow Mapping and Analysis:
Workflows and Processes in Organizations

Fundamentals of Workflow Mapping and Analysis

This chapter provides a standardized workflow mapping and analysis (WFMA) technique for diverse sets of users, ranging from those who need to study and understand complicated problems bound up in a flow of work, to those needing to capture and codify the knowledge used in those processes, to those needing only to diagram a well-understood procedure for purposes of training new employees or documenting a process for an ISO 9000 or other certification. In most cases, the applications will be somewhere in a middle ground between these extremes, and most WFMA tasks contain some parts of routine diagraming and some parts of diagnosis and discovery.

WFMA meets these objectives by showing how the "stuff" of any organization's work, whether tangibles liked manufactured materials, or intangibles like information or changes in patients' states of health, moves through that organization. In today's economy, the "material" that most organizations work on is information, and for the time being I will also refer to this as material, even though it is not. Any of the things that organizations do to accomplish their objectives involve both the processing of their materials and the processing of information about what to do with them. To begin at the beginning, then, we need to have some basic terms and concepts to describe what these things are, and what is going on in the workflow we want to map.

While there is evidence of WFMA as an increasingly important tool in Six Sigma programs and other aspects of quality management and performance improvement, there is no standard format for doing workflow maps; not uncommonly, these are referred to as "flowcharts" or "flow diagrams." I use the term "workflow mapping" to differentiate what is being done here from these other methods, and to reinforce the idea that what we are creating with this method really is a "map," more often through uncharted territory than we might imagine.

What Is Workflow Mapping and Analysis?

Workflow Mapping and Analysis (WFMA) is a graphic method of completely describing the materials and information flows necessary to accomplish one or more specific objectives of work, in their correct sequence, in a single job, a process, an organizational unit, or an entire organization.

WFMA has a number of important properties. First, it is *graphic*—it shows workflow processes visually, diagraming them as a flow of activities and information. Second, it is *standardized*, meaning that it uses a specific symbol set for all workflows or processes being mapped, and that these are used with a specific set of rules—a discipline, and this is as much part of WFMA as the symbols. Third, it is *scalable*, in that it can encompass all parts of a process at whatever level of detail is selected by the user. Fourth, it is *robust*—WFMA can be applied to any flow of work in virtually any kind of organization. Fifth, it is *verifiable*—it can describe existing processes as they are, and any map can be audited or checked against the actual flow of materials and

information and the behavior of jobholders to determine its accuracy. Finally, these properties make WFMA an important precursor for *process improvement* and workflow measurement—all workflow activities, flows, and decisions can be measured in a variety of terms that support improving the process.

What WFMA creates is a "static model" of a business system. When a workflow map is developed and verified the user has a graphic depiction of everything needed to do the work that process covers. If a manufacturing process or an insurance claim process is mapped, people can "walk through" the process as if they were on the plant floor or at the claim site.

What is a "Process?"

"Process" is one of those words that gets tossed around very loosely and in a lot of different contexts in the world of work. Over time, everyone uses it and understands it, at least for themselves; whether that meaning is the same as another user's, however, is often another matter.

To be sure we are all on the same footing, I want to take a few paragraphs to define what I mean by a "process," and also set the stage for understanding some of the properties of processes and the environments in which they occur. With that background, we are prepared to dig a little more deeply into what we mean by materials and information, our primary concerns in mapping and analyzing workflows.

We need just a bit of a model to follow here, and a useful tool for many organizations is to consider them as "systems." Any organization, or part of an organization, can be considered as a "system" in and of itself, or as a "subsystem" of a larger system. Any level of a system can be viewed as a continuous interaction of that element with its environment, where inputs are taken from the environment, transformed into outputs, and returned to the environment, as we see in Figure 2.1.

Examples of this kind of model are everywhere: a human being takes in air, food, water, etc., from the environment, processes them internally, and returns a variety of outputs, including items with added value along with used or waste products. Companies take in raw materials of many kinds (capital, unfinished materials, employee talents and skills) and combine these into new products and services. These are sent to the "market," an open and largely independent forum where potential customers can take or leave what companies have to offer. If customers take it, they give back money, the common medium of exchange, and the process repeats; if not, the firm loses its capital and fails. Interaction with the environment in both of these examples can be seen to take many forms, and in the case of companies and organizations, the relationships become very complex. Nearly any organization tries to influence what the "environment" does, through advertising, public relations, gaining favorable regulation, and the like. If we get cold because we are losing too much body heat to the environment, we try to regulate the exchange by putting on warmer clothes or through

other steps. So our model is a very simple version of what happens, but it captures the essence of the transactions that go on in a "system."

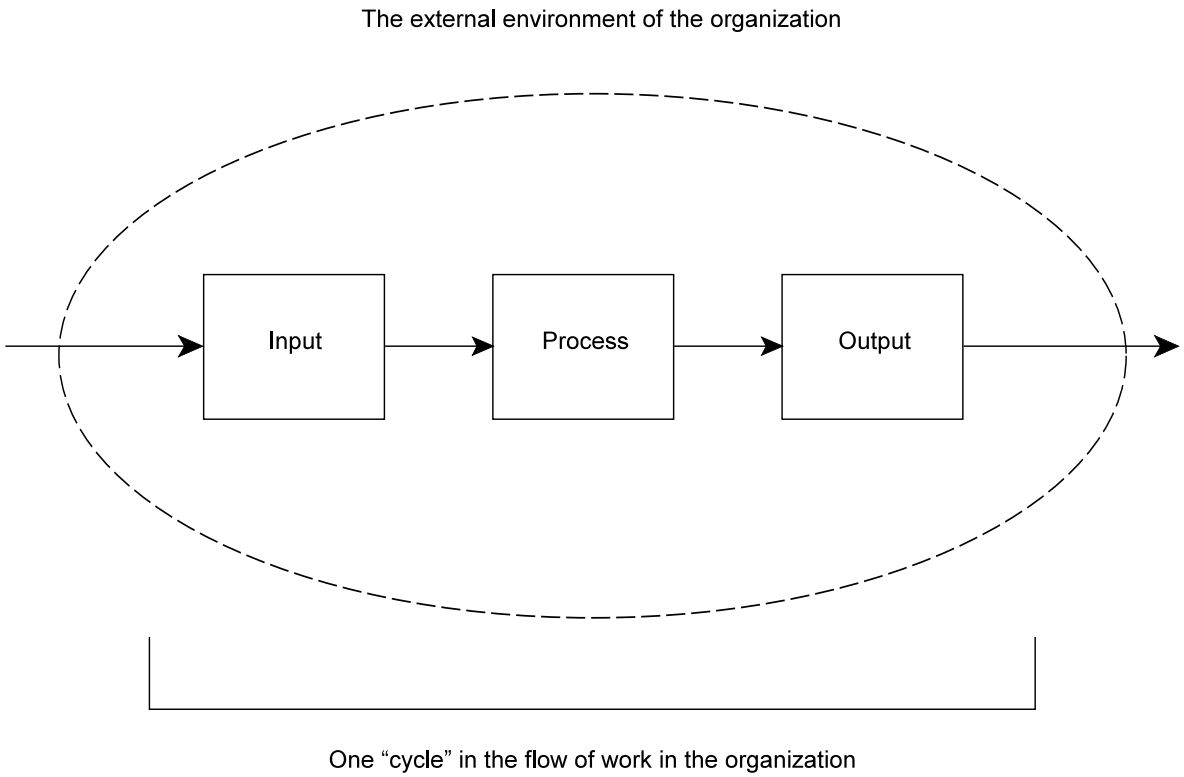


Figure 2.1 The basic system model

Source: John L. Kmetz, The Information Processing Theory of Organization: Managing technology accession in complex systems (Aldershot, England: Ashgate, 1998), p. 38.

The "process" in this model can be defined at two levels. The first is usually focused on how the transformation of inputs is carried out; the second (system-level) process also concerns itself with the exchanges with the environment that acquire inputs and place outputs. These processes may be a transformation of materials, of information, or both The majority of processes involve both, and it is hard to find examples of any process where information is not involved (even "automatic" ones like shivering if we are cold—this involves an internal flow of information about the body's heat level and its changing status over time, and if we are losing too much heat we shiver to increase our warmth, even though we never think about doing it). Processes in organizations thus consist of one or more "workflows," in which associated streams of information and/or materials are transformed over time into outputs, and these are returned to the environment of the system.

Processes are *dynamic*—they occur over time, and are *linked by information*. Most of this information is usually created by specific activities in the flow of work. This information takes two generic forms: (1) "feedback," where some information or output is consumed to maintain control and evaluate performance against expectations—some of the work done by the system is consumed by the system for self-regulation—and (2) "feedforward," which sets targets and establishes criteria used to guide future action and decisions. Figure 2.2 illustrates the idea of these two types of information in the work of an organization.

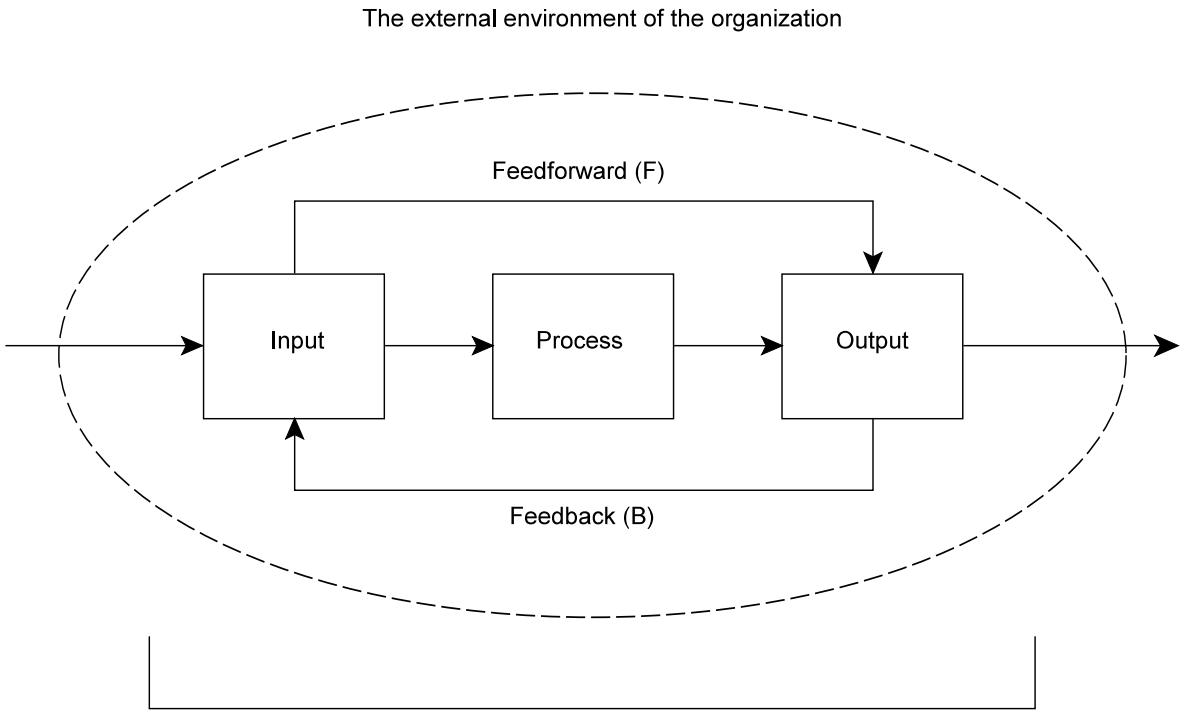


Figure 2.2 The basic system model with dynamic information flows

Source: John L. Kmetz, The Information Processing Theory of Organization: Managing technology accession in complex systems (Aldershot, England: Ashgate, 1998), p. 39.

Most of us are very familiar with the idea of feedback, and we use that word all the time in many aspects of life—feedback from customers, significant others, suppliers, the government, and virtually everywhere else in the world. In companies and organizations, feedback includes financial control data, performance appraisals, assessment of responses to customers, quality measurements, and the like. We realize how important it is and collect and think about feedback all the time. Feedback

about processes is often obtained from internal sources, but many types are also externally derived.

"Feedforward" is a less common idea, but just as important as feedback. Feedforward consists of information like project plans, performance projections, consumer confidence surveys, budgets, benchmarking, and similar types. Feedforwards establish the plans and criteria that guide our actions, and give us a road map to follow, if we do it well. Many feedforwards are from external sources as well as internal ones.

Organizations as "Workflows"

Let's return to the idea of a "workflow" again. Understanding workflows, and the ability to map them, requires understanding of the "raw materials" the company brings in from the environment and the "technology" used to turn them into outputs (and everything, no matter how everyday it may seem, involves a technology of sorts). Those processes usually consist of many steps and activities, many of which are specific to the organization, and are often quite intricate. While knowing about these is necessary to map a workflow, the technical part of the process alone is not everything—we also need to know about the feedback and feedforward used to control the processes, and this also can become very intricate.

All organizations require two basic types of flows to accomplish their objectives. First, there are flows of *materials* in manufacturing, or "quasi-materials" such as documents and information in many manufacturing and service companies. Since the majority of the US economy is no longer manufacturing or agriculture, intangibles are the largest element of our production, and most of those intangibles have a large information content. For that reason, I am going to create the word "informaterials" to designate information as production; most of the time, I will use "material" to designate both physical material and informaterial.

Second, there are flows of *information* in both manufacturing and service organizations; we use the term information flows here to designate these as information about the primary work of the organization, but not the substance of the work itself (i.e., the informaterial). A marketing research firm provides information about markets to its clients as its product; as we use the terms here, the product is the informaterial flow, and its creation is guided by information flows.

Both types of flows must be understood to fully describe a process. To attempt to map the flow of physical material or informaterial through a process without understanding the role of feedback and feedforward is basically a waste of time. An excellent example of this is in the way that we handle things that go wrong in our day-to-day work—what we often refer to as *exception handing*. Unloading and reloading a container ship in a modern port is a complex process. Containers must unloaded in

such a manner as to minimize delays in moving from the ship to local transportation or storage, and reloaded so that the container order is correct for the receiving port. However, this can never be done perfectly for a number of reasons, one major one being that if the ship's load becomes too unbalanced at any time in unloading or reloading, she might simply roll over and sink! So despite the best planning and the use of intensive automation, exceptions and problems will occur, schedules will slip, and information must be processed to deal with these events.

Other exceptions are the "bolts from the blue." An unloading crane breaks down, completely upsetting the schedule for a ship; a package gets lost or delayed; a check isn't cut because of a clerical error by a trainee; a critical meeting is missed because the presenter (a) got caught in traffic, (b) got sick, (c) went into labor, (d) had to deal with a higher-priority emergency, etc. While exceptions such as these are often caused by events beyond our control, they must be dealt with nevertheless. It is not uncommon to find that significant amounts of managers' time goes into this one activity, and that in combination, exceptions may take more of any given day than the so-called "routine."

Flow relationships. A simple way of thinking about flow relationships is whether they are *convergent* or *divergent*. In many cases of process improvement and work simplification, we do want to make the flows of materials and information converge as much as possible. Separation of work from decision-making about the work, for example, is often a principal contributor to delay, error, and the perception of red tape (often the reality), as well as a demotivator for the people who do the job.

This is a case where the "yarn ball theory of organization" should be introduced. In many consulting and training cases, I have used a ball of yarn to illustrate a few simple truths about the way that organizations grow and how their structures change. The exercise is simple—I tie a loop in the loose end of a ball of yarn, put that over my wrist, and toss the ball to someone anywhere in the room. The instruction to everyone from that point on is the same—put a loop around one wrist, and toss the ball to someone else. This process continues with random tosses until everyone has at least one yarn loop around their wrist, or happens to be the last to get the yarn ball.

This exercise illustrates two useful things. First, if "a system is a thing made up of other things and connected to all other things," there is no better way to see this than to be webbed into the "Yarn Co." Second , no one anywhere can move without affecting at least several other people. The yarn linkages between people are the paths that information (movement) and materials have to follow for any work to get done. Rationalizing these relationships, meaning that we want to reexamine them and how they work after they have been put in place, is often the expressed or implied reason for any kind of WFMA project. Following the paths that materials and information follow in reality is at the heart of WFMA, and if we were to "rationalize" the Yarn Co., we would have to make many of the relationships that converge diverge, and vice versa.

Convergence and divergence may be adequate to describe workflows for many purposes, but in other situations it is helpful to break these down into two components. First, with respect to material flows, information flows may be *synchronous*, meaning that they operate simultaneously or in a very tight serial sequence, with the absolute minimum of delays in accessing or using information to support material flow. In synchronous flows, materials and information follow the same paths and are processed in parallel and usually in the same location. However, flows may also be *asynchronous*, in that they are nonparallel, follow different paths, cut through unit boundaries, follow different sequences, or a combination of these. Timing is less likely to be tightly linked between the flows, and tight linkages may be undesirable.

Typically, synchronous flows are preferred, as they keep materials and information about the process in close proximity, so that decisions happen quickly, exceptions are handled on the spot, and the most relevant expertise on the workflow issue is brought to bear on issues needing it. “If you want to dig a better ditch, ask a ditchdigger how to do it” applies here. Asynchronous material and information flows are typically considered undesirable, in that the lack of synchronization creates demands for additional information processing without adding value in return.

This is not always the case. Either type of flow can be described in a second way as *integrative* or *disruptive*, with respect to organizational objectives. Integrative relationships are those in which organizational objectives are served and value is added by following the process as it is designed. Disruptive flows are those which increase delays, costs, the probability of error, or a combination of these, and for which little or no value is gained in return. Because of these differences, the general bias toward convergence in process improvement may not always hold. Examples of asynchronous flow relationships which are integrative are not at all difficult to find—think of cases such as those where people who handle money are not allowed to reconcile accounts; where people with vested or financial interests in companies are not allowed to make public-policy decisions about these firms; or where those who claim to have made scientific breakthroughs are subject to independent replication and verification of their findings. Even though there is designed separation of functions and behaviors in these flows, they are ultimately integrative and add value. Scientific scandals such as “cold fusion” or the recent stem-cell fraud in South Korea were revealed as integrative benefits of asynchronous information flows. No drug is ever cleared for general public consumption unless it has gone through a double-blind review process during its testing phase. Much experience, some of it fatal, has shown the integrative effects of such double-blinding, where neither the doctor administering doses nor the patient receiving them knows whether a particular dose was an experimental medicine or a placebo. Asynchronous double-blind procedures protect the objectivity of the observers and the recipients, ensuring the most accurate possible data on drug effects.

In short, do not make the mistake of assuming that because material and information flows converge, they are necessary integrative or synchronous. In the same way, flows which diverge are not necessarily disruptive or significantly

asynchronous. The examples of relationships gone awry just above should make it clear that accurately mapping a workflow, and then using that knowledge as a way to improve it, requires somewhat deeper understanding of how we use information to control the flow of work.

Much of the purpose of WFMA is to describe, and in some cases to discover, exactly this kind of deeper understanding. The understanding begins with a *valid* map of what is actually being done, and this is sometimes a very difficult map to create, for reasons to be discussed shortly. Having this, however, enables properties like the integrative or disruptive nature of the workflow to be seen as well as matters of synchronization of actions and decisions.

Symbol and Mapping Rules

In Figure 2.3, the reader will find the basic symbol set used for all workflow mapping and analysis. The usual reaction to first seeing these is, “Is that all there is?” or “Big deal!” That is entirely by design. One of the hard lessons learned in developing this system in a fifteen-year long study of avionics maintenance in the US Navy’s Air Systems Command (NAVAIR), and a similar but shorter experience with the Canadian Air Force, is that the symbols have, in fact, very little to do with WFMA; they are a means to show a few important differences between universal elements comprising any flow of work, but the real objective is to understand the information content of the map, and that has much more to do with the words and expressions in the symbols, along with the relationships between them, than the symbols themselves. The symbols are still important, as we will see, and must be used the right way to achieve their objective. But they cannot dominate the mapping process.

This “right way” is the most critical part of WFMA, and what I am referring to is the *discipline* that must be used to apply WFMA successfully. This discipline must be learned and ingrained in its use at the outset, and required in all applications of it. This discipline is necessary to preserve the simplicity of the WFMA system, and thereby its power to describe a vast array of different kinds of workflows. At the same time, users should recognize that this discipline does not result in a single “correct” map for a given workflow or process; in fact, it makes it more likely that it will result in multiple possible “correct” maps, none of which will necessarily be the same.

How can this be so? I will deal with this issue in more detail in Chapter 4, but for the moment the answer is that any workflow map is a joint product of the flow of work being mapped or described, and the knowledge and perceptual processes of the mapper. There is no way to isolate these or their effects on each other, and so every map produced will be a function of what is done interacting with the eye of the beholder.

The graphic devices on the next page are a very simple set of flowcharting symbols which have been adopted as the entire set for creation of workflow maps. No

other symbols are needed, and as we will see, no others are desired. There is nothing unique or original about these, and many different flowcharting programs and systems use some or all of them as well. The short answer to the question of why these were selected is simply that through multiple trials of mapping workflows, validating these maps with operators on the site, seeking diagrams that could communicate clearly with other analysts, managers, and specialists in the US Navy, and which would do all the things I needed done, these five survived a Darwinian selection process. They work. They have been used in every conceivable circumstance and are so robust that nearly any organization can be mapped using them.

The first is the simple **rectangle** (and this must always be a rectangle—a square is not an acceptable substitute). It can show a huge variety of events and operations, including processes, activities, and location; it can be used to designate delays or holds (the absence of activity over time), such has waiting for paint to dry. It can be used for materials, information, and informaterials, in every organization in which it has ever been tried. It is often referred to as a *process block*.

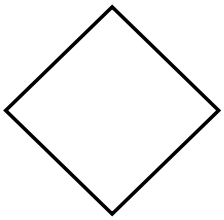
The second is the **diamond**. This is simply a small square rotated 45 degrees, and represents a decision or a branching point in a flow. It is because this is a small square that the rectangle must never be—these symbols must be kept visually distinctive. The diamond is the only symbol that has two exit paths, which must be *mutually exclusive*. It is primarily an information symbol, although it can be used to show how physical items are parsed or divided.

Third, there is the **connector**, which is a small circle. This can be used individually to designate “start” and “stop” points, but is most commonly used where matched pairs enable a large, complex map to be arranged with connectors for clarity. A connector labeled “A” at the end of an arrow designates that another “A” at the head of an arrow leading into the workflow is where the flow continues. These may also be used to indicate where different parts of a path might lead back to a single re-entry point in a flow. (Since many programming purists over the years have harrassed me on the use of connectors for “start” and “stop” events, I have allowed the small circle to be stretched into an oval, without repercussions for those who do this.)

The fourth symbol is the **arrow**. This is a single-headed arrow to be used individually between other symbols. Pairs of arrows from any source other than a decision diamond are not allowed, nor is a double head, i.e., an arrow with opposing heads on each end.

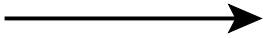
Finally, there is the **document**. This is a bit of an anachronism. The truth is that most of what is shown using a document symbol can be represented just as effectively with a rectangle, but I have found that there are often circumstances where the need to show that a document was produced, or a rule referred to in a process, or a report generated, and the like, were very important. For that reason, it survives.

RECTANGLE:
- Process, Activity
- Location
- Delay, Hold
- Both material and information symbol

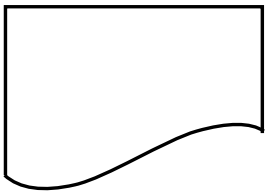


DIAMOND:
- Decision, Branch
- Two *mutually exclusive* exit paths only (yes/no, true/false)
- Primarily information symbol

CONNECTOR:
- Used in matched pairs or groups to connect parts of flow or continue across pages (“A” connects to “A,” etc.)
- Can also be used singly for “start” and “stop” (traditional oval variation OK for these uses)



ARROW:
- Material or information flow in direction of arrowhead
- Always single direction and single arrow
- Material and information symbol



DOCUMENT:
- Paper input or output
- Used for required documentation in or out
- Material (“quasi-material”) and information symbol

Figure 2.3 The WFMA symbol set

Source: John L. Kmetz, *The Information Processing Theory of Organization: Managing technology accession in complex systems* (Aldershot, England: Ashgate, 1998): Adapted from Figure A8.1.1, p. 378.

The selection of these symbols was rather Darwinian. What I mean by this is that they are the "survivors" of a number of different candidates that I tried over time. There are several related reasons for their survival. First, they are all visually distinctive—if drawn as shown, there is no possibility for any of them to be confused with any other symbol. Second, that means they can be learned very quickly, without requiring a user to first learn either a complex set of symbols or the mastery of particular software. Third, they are robust, such that they can be applied to virtually any flow of work. Finally, they are comprehensive—they are able to capture any activities in a workflow or process. Together, they establish a standardized graphic vocabulary for mapping workflows in organizations.

There are a number of rules for symbol use, which are discussed next. I refer to these as "rules" rather than "suggestions" or some gentler term because I have found that violating them, even slightly, leads to still more violations, and that these progressively destroy the standardization and simplicity which is at the heart of WFMA.

The rules are the syntax for constructing workflow maps. When applied correctly, they have the key benefit of making it possible for the symbol set to quickly retreat into the background of any WFMA application or examination. *The objective of such a simple set is to make it disappear in use—if more than two minutes is needed to completely learn the symbols, they are too complex*. Literally, if one flips through a five-slide show, with one symbol and the bullet points explaining it as shown in Figure 2.3, an audience with handouts should need no more than 25 seconds to understand each symbol.

Somewhat more time is needed to learn the rules, but these should also quickly become part of the standard approach to WFMA. Learning the rules is best done with hands-on practice, beginning with something relatively simple and then progressing to a more complex process. *Nearly anyone should become an "expert" in WFMA in no more than one morning's work.*

What I learned in my early experiences with development of my approach is that the information conveyed by the map, which is conveyed by the brief descriptors used in conjunction with the arrangement of symbols that describes how something gets done, is the only thing that matters. If the mapping graphics and rules get in the way, they obscure what they should reveal, and they distract from the analysis rather than support it. In no case is this ever a good thing, and in situations where there may be political or other opponents of what is being done with WFMA, these distractions provide ammunition for them to discredit the entire program and its objectives.

WFMA Discipline—the Rules for Symbol Use

1. **Keep It Simple, Stupid (KISS).** The basic WFMA symbol set is based on a sound psychological principle that favors fewer rather than more symbols. Busy people have a hard time keeping large numbers of symbols straight in their minds, and overload is never far away.

Many people suggest that a method that makes workflow mapping easy is to simply write down, in step-by-step fashion, what is done in a process; some even contend that this is more than enough to provide the information a map would give. I generally disagree with both of these arguments. Very few processes are so completely linear as to allow a clear, well-organized written description. What do you do, for example, each time you come to a branch or decision? How do you keep track of where you have been when you're on page 26 and don't quite remember which set of branches you took to get here? When you have to depart from the main process to deal with exceptions, how do you track through what are sometimes complex exception-handling procedures and remember what was happening when you return to the main flow?

2. **Use the symbols correctly.** A major benefit of simple symbols is that they have highly specific meanings. Using them as intended prevents confusion; be especially careful not to violate these rules unless there are truly compelling circumstances:

- Use the symbols only for the purposes shown. For example, never show decisions in process boxes (a fairly common error, actually).

- Use brief, clear text within the symbols (as much as possible) to communicate what is happening at each step; in the case of decision diamonds, the exit path arrows will also have text associated with them. If additional annotation is needed, this can be added in other locations on the page or through notes—this will be discussed in more detail below.

- Use only single arrows, always with the arrowhead to designate direction. Two-way or double-headed arrows are ***never*** allowed.

- Use only ***two mutually exclusive*** exit paths from a decision diamond (unless there is truly no other choice, and that is extremely rare). Mutually exclusive paths mean that there can be no confusion over what each path means or why it would be selected rather than the other one. This is an extremely important rule, and in some cases may make a workflow map segment larger than multiple exit paths would suggest. Here are two examples: if we apply a first coat of paint to a wall and want to recoat it, we might show this as a decision ("first coat dry?") with "Yes" and "No" exit paths. However, which one applies to "touch dry?" We might instead use

three paths labeled “fully dry,” “touch dry,” and “wet.” To make sure that we do not mistakenly apply the second coat too quickly, we may need to add one or more additional conditions (decision diamonds) to the diagram to be unambiguous. In a second actual case, credit-card customers were being reviewed at random, and for those with some questions about their credit histories or experience, a possible outcome of the review was to limit their account or suspend it entirely; most, however, were at least renewed and frequently offered a higher line of credit. The student with whom I had this argument made a logical point—there could be five outcomes from a single review decision. These were (1) raise the limit; (2) renew the account at present limit; (3) renew the account with lower limit; (4) suspend the account for a limited time; and (5) close the account. The student argued that all five exits paths could come from a single “Account review outcome?” diamond, and that there was no need to make this more complicated by the addition of a stepwise series of two-outcome reviews.

In both cases, it sounds logical to use more than two exit paths from a decision, since this reflects the actual world of work. My contention is that if we do not insist on mutually exclusive paths at all branches in the workflow map, it is inevitable that ambiguous conditions for branching will creep into the map, and what seems absolutely clear to the person(s) creating the map will be completely confusing to other users. My personal experience with this issue bears this out. Moreover, the consistent use of two paths becomes habituated for both mappers and users, so that when a decision diamond is encountered, there will always be two paths to evaluate—no more, no less.

The most important reason for consistent use of two mutually exclusive paths, however, is that these require the creators of the map to fully articulate the logic for the branching that occurs at that point. This is often harder to do than we might think, because much of what gets done within many jobs is a process of evolutionary change and internalized learning. Having to spell out all that is done and why it is done involves more head-scratching than is apparent, but the payoff is that a great deal of what we will later discuss as *tacit knowledge* is uncovered this way. I will have more to say on this subject in Chapter 4.

- Put brief labels on exit paths from a decision diamond. This enables the use to follow the map clearly, and helps keep the logic of the map visible and accessible to the user.

- Avoid complex backward flows. These are acceptable on small areas of the map, but can be confusing on large areas; use connector circles instead to break the map into sections and pieces which can more easily be followed.

- Be careful of creating endless loops. This can happen for several reasons, but typically is because of the incorrect use of double (or double-headed) arrows in a flow, or ambiguous criteria for a decision or branch.

- Avoid making a single page too dense, whether with large numbers of symbols and arrows, too much text, small symbols and fonts, or a combination of these. In an earlier lifetime in the printing industry, we used the term "white space" to indicate that there was enough unprinted area on a page to make the printed parts accessible and stylistically appealing. While one always has to struggle with the tradeoff between detail and content per page versus overall document size, adequate white space is beneficial to workflow maps.

3. **Make WFMA your own, but limit "customization."** Every organization needs to make a few changes to the basic symbols to account for things unique to that organization. There will also be some need for an organizational standard format. Do these things, but only as much as really necessary. *Do not* add more symbols (or "modifications" to the symbol set) without *compelling* reason to do so.

This rule is especially likely to be violated by those with flowcharting, data flow diagraming, or similar kinds of experience. The temptation to add a shadow to a rectangle after doing it for years is hard to resist; so is using a round-shouldered as opposed to right-angled rectangle. But doing this now requires the use to learn, and to carefully observe, these subtle differences while also attending to the logic of the flow and the information contained in the symbols; this introduces noise and inevitable confusion to the system, and this is never a good thing.

4. **Start by creating *actual* maps of the workflow, not the maps that "should be!"** A sure way to waste lots of time with WFMA is to let people diagram the way (they think) work *should be* done when you need to know how it is *actually done now*. Most WFMA projects start with the intention of description and diagnosis, which is usually the right idea. Validation, the next step, will help ensure that the map really reflects what is being done, and that is the point of departure for any application of the map. Mapping what should be done is almost always an exercise in fiction writing, to some extent. Fiction writing is *not* the right idea, no matter how good the final story.

This should not be interpreted as meaning that a normative, "should be" map can never be drawn—quite the opposite, a normative map might serve as a valuable straw man for discussion of how one or more jobs might be redesigned, for example. The problem is that when this approach is used for many existing jobs, the version of reality that is reflected in the map will be a single view of it. In mapping the workflows on seven aircraft carriers and seven shore sites, not a single one was the same; nevertheless, I had three experienced chief petty officers provide me with "ideal" workflow and organizational maps. None of those matched, either.

5. **Validate the map.** When a process map has been done, trace it through to the final details. If the map doesn't completely match the process, it is not a valid map, and you don't know what is being done in this process. Revise the map and try again.

This is a task that often falls on the manager or originator of the mapping process, and sometimes requires the support of higher management (such as the process owner, to be discussed in the next chapter). In simple terms, that person has to be a pest and do a "walk through" of the map with the person(s) who created it. At every step, that person needs to be prepared to ask questions, clarify terms, correct some of the mapping discipline, add or subtract steps, and the like. Typically, a revised map will be needed, and this may happen several times until the map is fully validated. This is a high value-added step, and if it is not done there are many things that can go wrong with the process.

In the majority of cases, it is most important to get a fully validated map. This means that there will be "little things" that are done that don't match the general flow, and some of these aren't "little things." It is not at all uncommon for the major part of the work done in a process to be the smaller part of a map, while exceptions and deviations from the norm eat up lots of time and energy to diagram—often mimicking the reality of dealing with them. Getting this information will take persistence, but this is where process variation lives, and it is often where the richest opportunities for process improvement are found as well.

6. **What happens before, and what happens next?** If a map can't account for where something came from before it got to this location, or where it goes next, is it really connected to the rest of the organization; is it really connected to critical suppliers; is it really connected to customers? *Again, organizations are systems.* Valid workflow maps must show what the system is actually doing, and without a valid map meaningful process improvement or other change is most unlikely to be possible.

Like the symbol set, the rules for use of them are also simple. The discipline needed to make WFMA effective is to apply these rules strictly, and not to allow variations to creep into either the maps or the procedures that produce them. If this is not done, any moderately large firm, or firm with geographically separated offices, will inevitably begin to develop divergent and incompatible WFMA techniques, and the power of WFMA will be lost.

Basic Mapping Conventions

On the following three pages, Figures 2.4 to 2.6 illustrate the major conventions in constructing a workflow map which have developed over the past several decades. Figure 2.4 illustrates the principal relationships between diagram elements as they might be assembled in nearly any workflow map. Annotation of various parts of the process can be added if they add clarity, as shown, and this will be discussed momentarily.

Figure 2.5 illustrates a repetitive loop in a diagram. While I generally discourage backward flow loops in diagrams, as noted, this is a case where it clearly makes more sense to keep the diagram on one page rather than force some kind of artistic compromise to avoid such a loop. Proliferation of such backward loops, however, can become a threat to the clarity of a map, and one excellent application of connectors is to reduce the extent to which backward (and especially crossing) flows are used in a map.

Figure 2.6 illustrates how processes can be viewed at differing levels of detail. On the left side of the figure the four high-level steps necessary for completion of a transaction in this financial institution are shown. The right side shows the first of these four steps, the taking of the order, in more detail. Here we can see the three different ways that orders can be received (telephone, fax, or via dedicated computer link), and the additional activities needed to prepare them for the next major part of the process.

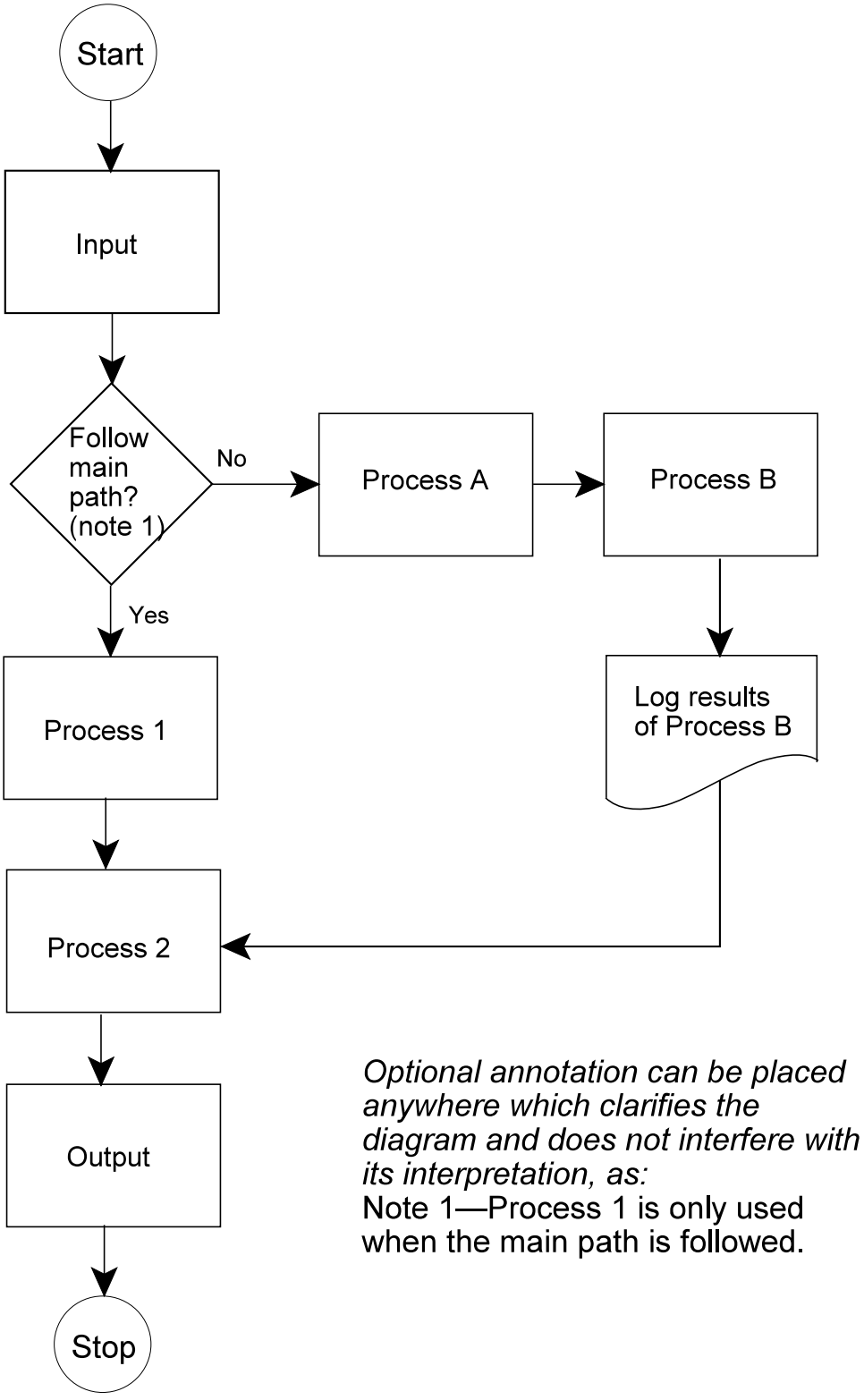


Figure 2.4 Single-cycle Process Flow

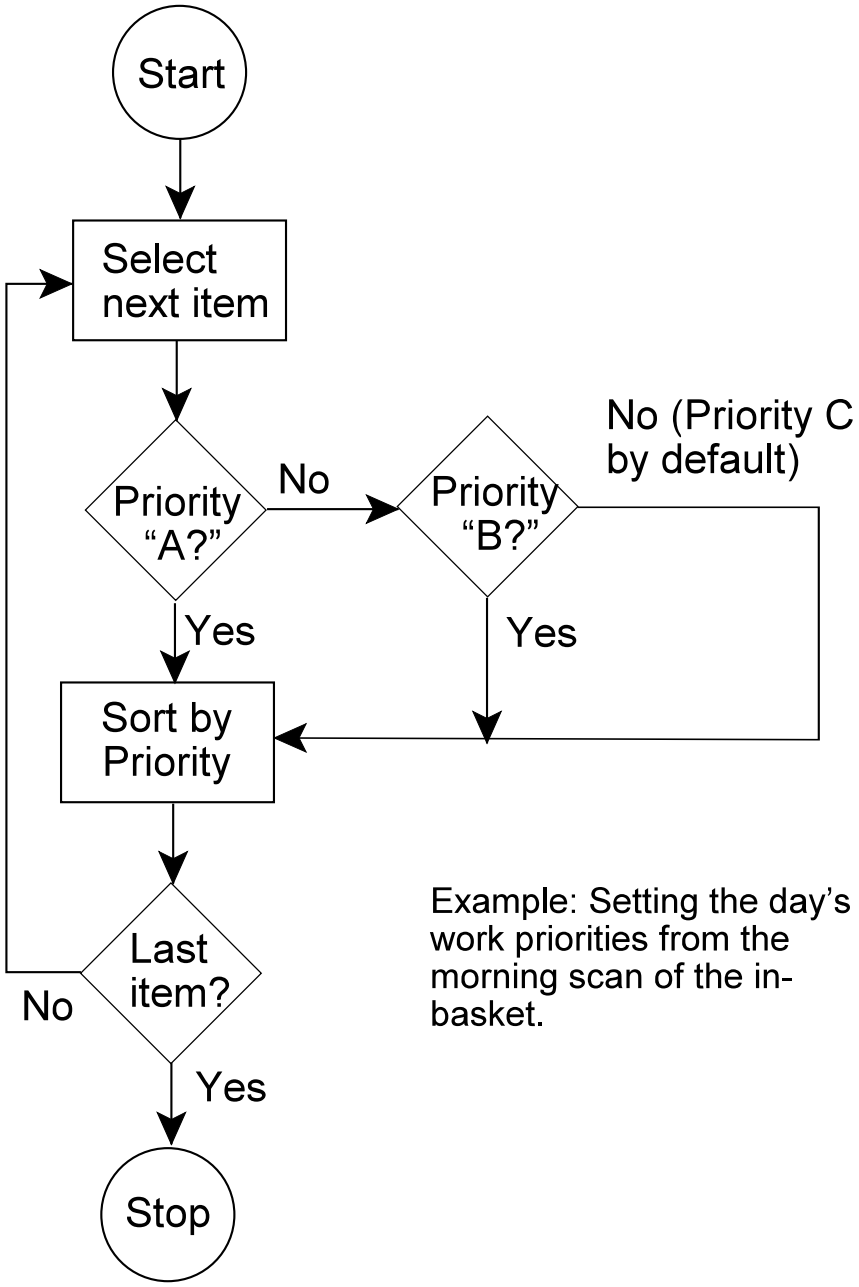


Figure 2.5 Multiple-cycle ("looping") Process Flow

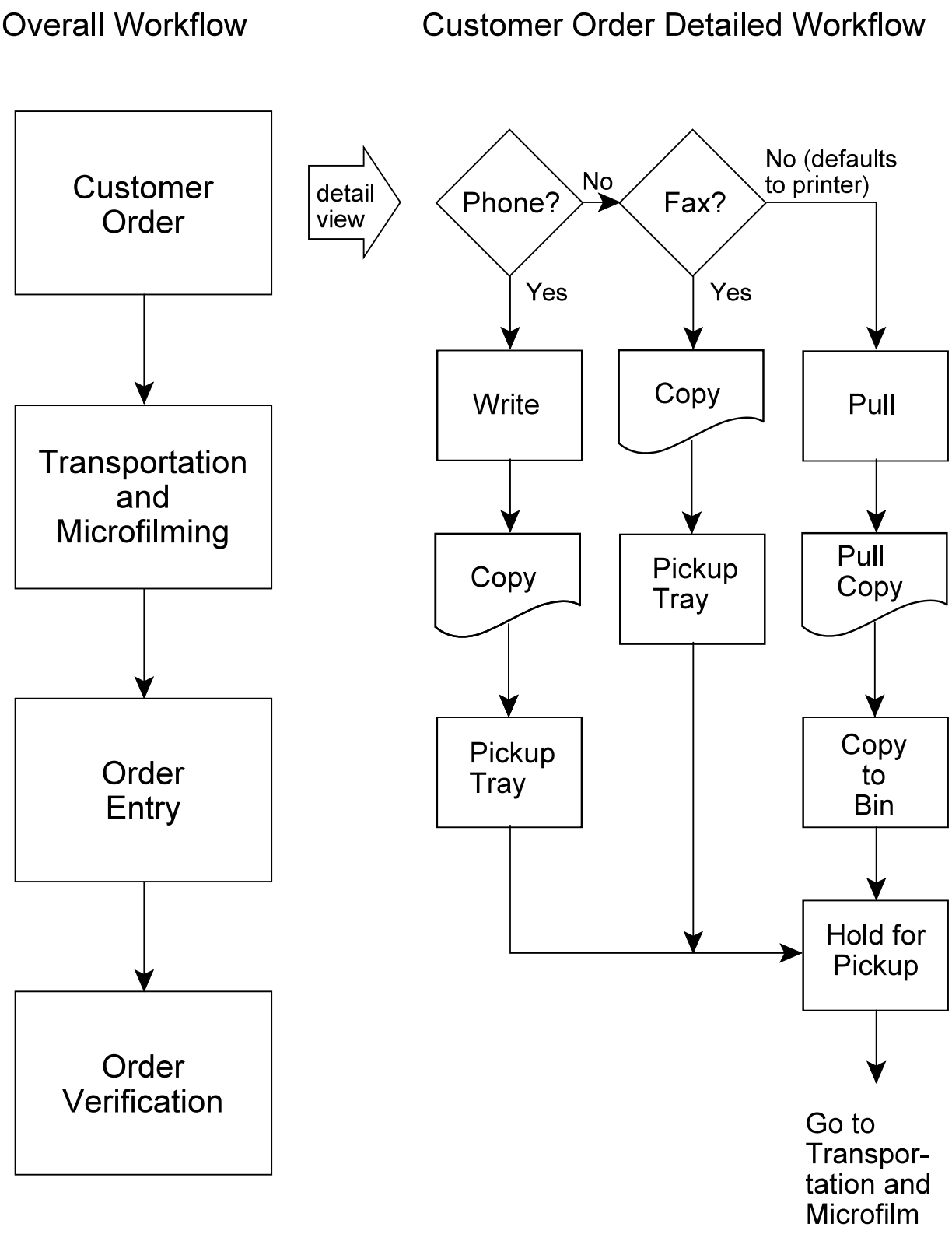


Figure 2.6 Two levels of detail in WFMA (“drill-down” or “layer”)

Using Annotation in WFMA Diagrams

The general rule for WFMA is that we want to keep as much of the relevant text and information about a process flow inside the process symbols as is possible. This is part of the WFMA discipline—when the user follows the diagram, there will be an automatic association between the symbols (for which, again, we want the minimum user processing) and the information in them. Together, these fully define the workflow we are interested in mapping.

Occasions do arise where annotation apart from the symbols is necessary, and there are many such circumstances. In making a training map, we might note to the trainee that things done in one step may be referred to another office under some conditions; that there may be supplier or other external relationships that could affect a part of the flow; that random samples of work may be taken for quality assurance; and the like. It is also possible that questions might be anticipated, as in Figure 2.4 above.

Whatever the situation requiring annotation, there are several rules of thumb that should be followed to the greatest extent possible:

1. Try to keep annotation within the area of the relevant symbols. This simply associates the annotation with the process steps it refers to, and prevents the user from having to go somewhere else in the document to get the desired information.

2. Do not mix annotation directly with process information in the process symbols. Annotation implies separate information, and it is best to keep it that way. When annotation is used, it should be apparent that it is annotation.

3. Avoid using straight arrows for annotation; if possible, connective symbols in general should be avoided. I generally discourage the use of symbols to associate comments and annotation with the elements of the map—introducing the famous engineer's "curving arrow" may be appropriate in some cases, but too many of them clutter the diagram and make misinterpretation of it more likely rather than less. My preference is to use relatively standard typographic tools such as asterisks, parenthetic note references, footnotes, and endnotes to associate supplemental information with the diagram. Footnotes are often a very effective way to make this association, and easy to fit onto a page with modern word processors. This is a convenient association for making printed copies of a map, and is also useful for electronic documents, since relatively large displays with high resolution have become common.

4. Use annotation appropriate to the level of detail in the workflow map. Annotation is sometimes used to address matters at levels much higher or lower in the organization of work than is necessary, and this can be confusing and distracting. If one is looking at the high-level (left side) view of Figure 2.6, for

example, it might be appropriate to note why the sequence is what it was at that time. (This is an historical case where time is money, and these are high-value transactions; thus, nothing was done until the order was logged and officially in the system. Needless to say, this was automated a long time ago.) It would not be appropriate to explain at the high level that pulling a copy from the computer printer meant separating five-part forms with carbon sheets between (as was the case then); this would be appropriate to the detail view on the right side.

This is a case where the layering capabilities of many drawing and workflow-mapping software packages can be used to excellent advantage. Figure 2.6 would be separated into two layers (the high level and detail views), and each could be annotated as needed. A user wanting to have easy access to information at different levels would only need to change to the desired layer to see the alternate view.

5. Hyperlinks. Hyperlinks, the familiar blue-colored underlined connections we use to browse the Internet, are easily created with nearly every major software package, and can be a very useful form of annotation. As an example, suppose that a particular step in the administration of an experimental drug to a patient requires the consultation of an attending physician, specific attention to several important patient-health details as well as other matters, and that a form must be signed by the appropriate physician upon giving a dose. Trying to annotate this much supporting information at a single point in a workflow map might be dysfunctional given the length of the procedures and forms to be completed. However, this requirement could be met with annotation such as, “Only attending physicians are allowed to give this drug and they must document their actions every time.” The hyperlink associated with “document” may connect to a lengthy and complicated research protocol; the workflow map, on the other hand, can immediately and clearly go on its way. We will also discuss hyperlinks in the correct use of software in Chapter 3.

Workflow Mapping versus Flowcharting

Workflow mapping is *not* flowcharting, although the outward similarity at first makes it seem so. Most methods of charting or mapping workflows are derived from standard flowcharting symbols, as mine has been, but there are no truly standardized methods for mapping (Harris, 1999:155). While well-known flowcharting symbols are used to create workflow maps, there are a number of important differences between WFMA and flowcharting as the latter is applied in programming and information technology environments:

1. Only a limited set of flowcharting symbols and rules are used for workflow mapping, and these are distinctive and easy to learn and apply. This is quite

important for workflow maps to be useful to those not trained in flowcharting, and makes the symbols a means to the desired end, which is understanding of the workflow. Most flowcharting symbols, in fact, are far too specialized for workflow mapping.

2. Workflow mapping captures and describes an actual process used to create or contribute to an output, whatever steps may be involved. Flowcharting works from a logical program structure and follows exact rules of computer language. In that sense, workflow maps are *descriptive*, where flowcharts are *prescriptive*.

3. Workflow maps can vary enormously; it is unlikely that two different workflow mappers would create the same map for any job or process; at the same time, many different versions of a workflow map can be "correct." Flowcharting will produce a "right" answer that can vary only to the extent the programming language allows. It is likely that two skilled programmers would solve a programming problem in relatively similar ways.

4. Levels of detail and presentation format can vary enormously in workflow maps, whereas flowcharts must provide sufficient detail to write code, and cannot ignore rules and syntax the computer language requires. Workflow maps can and should be tailored to the level of detail most useful and informative to the immediate user. Software "drill-down" capabilities can be used to excellent advantage to embed varying levels of detail within the process map as shown in Figure 2.6 above.

5. Workflow mapping is an iterative process of discovery and investigation, not of application of externally-determined coding rules; the objective of workflow mapping is to develop a valid map of a process or workflow. The objective of flowcharting is to develop an efficient, correct answer to a programming problem, and by definition that means it has to run.