



Why Clarity and Holism Matter for Managing Human Service Information (And How the Sector Can Achieve Them)

by [Derek Coursen](#)

Published: April 1, 2012

(Article URL: <http://www.tdan.com/view-articles/15967>)

The human service sector's chronic weakness in managing information is partly the result of loosely defined and myopically framed concepts that lay a poor foundation for designing information systems. The problem can be solved within software development projects by modeling the domain with stakeholders in a constructively critical spirit as the first stage of eliciting requirements. This approach can unexpectedly clarify and expand stakeholders' understanding of their work. It can produce powerfully reformulated human service concepts that point toward strong domain models for the sector.

©2012 by Derek Coursen, all rights reserved; published by permission.

ACKNOWLEDGEMENTS: Thanks to Rachel Miller for thoughtful comments on this article, to Megan Golden and Chris Stone for years of encouragement, and to the stakeholders of many human service programs who insightfully participated in piloting the methods presented here.

The human service sector, both governmental and nonprofit, is chronically weak in managing information. Symptoms include—to name a few—expensive and ignominious software project failures, widespread deployment of systems that serve only limited constituencies (e.g., operations or performance measurement but not both), and an absence of systems able to fully integrate efforts across the spectrum of practice areas (e.g., substance abuse, homelessness, job placement, domestic violence and child welfare). People working in the sector become used to a background buzz of uncomprehending frustration as stakeholders stand around wondering why—with so much effort being put into software development and so much data being collected—their needs are still unmet.

Some contributing factors are easy to identify. Compared with for-profit enterprises, technology investment in the human services is low. Practices and policies vary widely and change frequently across a labyrinth of agencies and funding streams. And because metrics of success are not straightforward, stakeholder groups with different roles and interests jostle to advance their various agendas for what data ought to be captured to answer what questions.

But there are deeper and less obvious factors in the mix as well. A human service environment is a complex and fluid social reality that is inherently difficult to model. That difficulty is exacerbated by the sector's core concepts and terminology, which are so imprecise and myopic that they provide a very poor foundation for designing information systems. Established software development methodologies neither recognize nor address these problems.

To solve them, it is necessary to organize software development projects in a new way, first analyzing the specific weaknesses of current concepts and reformulating them so as to build clearer and more holistic models of human service environments. Doing this can have surprising benefits not only for

software development, but also for the way that human service professionals think about and run their operations.

The Weaknesses of Human Service Concepts

Widgets are a convenient counterpoint. Factories obtain parts from various vendors, build widgets, and pass them on to networks of wholesalers and retailers. These operations are easy to describe, and the software systems that support them are organized around very strong concepts: product, component, supplier, sales order, quantity, price, etc. Because the concepts are so straightforward, the boundaries and intended functionality of a manufacturing inventory or commercial ordering system are also straightforward. Different enterprises can use the same kind of system to deal with any kind of widget. Exchanging data is straightforward too. When you are dealing with widgets, you know what you are talking about.

In comparison, what happens in human service environments is less tangible and messier to describe. The human services exist in order to solve, ameliorate or control a set of problems that adhere to individual people but have dimensions relating to family and interpersonal relationships, economic activity, physical and mental health and the social integration of disadvantaged groups. How to characterize the problems is deeply contested, and so are the appropriate ways of addressing them.

To complicate matters more, the terms that human service professionals use for operational purposes, which were inherited from paper-based systems, have grave defects. The most central concepts are program, client, and case. Other key concepts are screening, intake/enrollment, issue/problem/need, encounter/session, service, referral, outcome and closure/discharge. These terms, individually and as an interdependent set, are vague and ambiguous, interpreted differently in different contexts, and have fuzzy, arbitrary or unstable boundaries.

The notion of a program is based on its meaning of “a definite plan or scheme” (Simpson and Weiner, 1989). The case originated in the traditional multi-leaved folders of the legal and medical professions, and has been rightly called “an egregious transfer of an analog concept to the digital environment” (Fitch, 2007). These terms merely conjure up the image of definite plans for managing cardboard. Yet they do have functions. The program is meant to delineate a clearly intentional set of practices and a theory of change for a particular problem (as opposed to offering undifferentiated casework) and to allocate staff and resources toward that purpose (Kettner, Moroney and Martin, 1999). The case, then, can be understood as an individual instance of a programmatic intervention.

In real life, though, the boundaries of both the case and the program are fuzzy in many ways:

- Although a case is usually centered on one index client, services are often designed to involve and benefit family members and others in the client’s life. Those collaterals are thus not formally clients yet in some senses are treated as though they are. To muddy the waters further, some of them could be index clients in independent cases of their own.
- The temporal boundaries of the case are partly an arbitrary convention: if a client leaves a program and returns, the program will need to decide whether to treat the situation as a single case or sequential ones. Furthermore, outcome measurement often requires follow-up after a case has supposedly ended; when performed by operational staff, such follow-up easily shades into continued service.
- Multi-service agencies often refer clients from one internal program to another, and staff members may allocate their time between multiple programs having adjacent models and

overlapping clienteles. This can make a given client's (theoretically separate) cases in different programs hard to disentangle in practice.

- Pressure to assemble grant funds from different sources can motivate an agency to stretch a program model so that it will accommodate a particular funder's priorities; that can cause gaps between funders' and grantees' notions of the program's purpose and boundaries. Such situations can erode the meaning of the term program until it merely denotes a funding stream, not a separate set of practices and resources.

Other concepts around clients' problems, programmatic interventions and results or outcomes have similar weaknesses. What exactly is an encounter and what is a service? (Can an encounter contain several services—or could a single service be delivered incrementally across several encounters?) And what sequence of actions and responses must two service providers and a client perform before it can be said that a referral has occurred?

The relationship between concepts and what they refer to is far less precise, less complete and less stable in the human services than in enterprises dealing with widgets. At best, these concepts function as a simplifying overlay that facilitates operations and rudimentary analysis while ignoring the reality of complex relationships and fuzzy boundaries.

Furthermore, by focusing attention within too small a circle, human service concepts can promote a sort of myopia. Of course, it is necessary to pay attention to nearby objects—but it is also necessary to be able to step back from them to consider larger wholes. The concept of the program, for example, draws a relevant boundary around a human service endeavor. But a world of necessary information falls outside of it. To understand a program's reach and impact it is critically important to analyze the pool of potential participants who touch the intake process but, for various reasons, are never enrolled. And each program is just one part of the whole human service endeavor in a given locality. Clients participate in other programs and there is an increasing need to assimilate information among them in order to integrate services (Ragan, 2003), to understand the interplay between different social problems, and to share measurement across multi-program efforts aimed at collective impact (Hanleybrown, Kania and Kramer, 2012).

When stakeholders embark on developing a human service information system, the most basic building blocks they start from are these vague, simplistic and myopic concepts.

Reinterpreting the Causes of Human Service Software Project Failures

Software development methodologies tend to assume that subject matter experts already possess a professional terminology capable of guiding the way toward clear system requirements, and that satisfying those requirements (on time and within cost, of course) can be safely construed as success. In the human service sector, those assumptions can be fatal.

Since the core concepts have such loose definitions and vague boundaries, each software design team must make sense of them in the particular context in which the particular stakeholders are expressing their particular requirements. At the same time, the terminology itself undermines stakeholders' ability to understand their environment and frame their needs well. This frequently results in definitions, requirements and a system design which, if poked and prodded in a skeptical spirit, could not stand their ground for very long. Unfortunately, many projects fail to build in enough poking and prodding to surface such weaknesses. Instead, reality does it for them in a stereotypical sequence. It goes like this:

In the beginning, business analysts collect requirements and draw use cases, workflow diagrams, and mockups of interfaces and reports. When they notice vague definitions and unclear boundaries, the analysts look for authoritative answers. The answers they get come from current procedures, clinical protocols, administrative regulations or legislation, or simply from custom, further influenced by the perceived immediate costs of software development and data entry. In this grab-bag, arbitrary and ephemeral answers tend to crowd out meaningful ones.

The requirements then go to the database designer. If the designer has a decent background in data modeling—and the sad truth is that many do not—then a part of his or her mind will try to infer a logical and stable order from them. But human service concepts, so light on intrinsic meaning, provide limited guidance. Yet the database must be built, so the designer soldiers on. The goal becomes merely to create an artifact that will support all of the interfaces, workflows and reports that the analysts have documented. If the designer is a truly poor modeler, he or she may shroud the already hazy meaning of the concepts under a further layer of obscure terminology referring to bureaucratic processes and forms. Developers then proceed to build an entire edifice of software components on top of this dreary substructure. A human service information system comes into being.

And then, gradually, the crows come home to roost. Unforeseen happenings begin to challenge conceptual definitions and boundaries that had been thought fixed. But modifying or expanding the concepts would require changing the database design—the physical foundation of the system—and much of the software edifice that depends upon it; and managers must confront the cost of doing that. A large proportion of information management problems in human service settings conform to this pattern. Where they differ is in the nature of the unforeseen happenings and the consequences for the human service agency.

Often, the unforeseen happenings are changes in the programmatic, legislative, regulatory, clinical or evaluation environment. This is recognized as a serious problem. Examining how the federal and state governments spent an estimated US\$20 billion over twenty years on information systems to administer seven of the largest human service programs, the General Accounting Office found that changing needs and program requirements often made systems obsolete before they could be completed (Nathan and Ragan, 2001). Conventional software wisdom would characterize this kind of change as a project management challenge. True enough, but database design is the most powerful factor determining an information system's capacity to be adapted. Widget management databases can accommodate an infinite and ever-changing variety of widgets; human service databases have not achieved that kind of flexibility.

Other unforeseen happenings originate in tensions between operational and analytic needs. Some stakeholder groups (front-line caseworkers and their supervisors) are primarily concerned with how the software can streamline their workflow and give them immediately relevant information. Others (performance measurement staff and evaluators) want the data to answer larger questions. Any group whose needs cannot be met by the existing system design will eventually make itself heard.

Conventional software wisdom again files this kind of problem under the heading of project management, asking: Why were all the requirements of all the stakeholder groups not elicited? That is a valid perspective, but it ignores the effect of weak concepts on database design. Again, widgets are an instructive comparison. A software system built purely to track widget sales can provide good data for marketing analytics too; but data from a system built purely to serve human service operations is rarely of much use for performance measurement or evaluation. Why? It is not merely that the analytical stakeholders want to collect some extra data elements. Just as often, the problem is that operational and analytical stakeholder groups frame the same subject areas in different ways.

These contradictions suggest that the conventional software development paradigm is itself a part of the problem. Projects are understood as attempts to delineate and satisfy user requirements, so unforeseen happenings tend to be explained either as failures to elicit existing requirements or as the emergence (inconvenient but unavoidable) of new ones. These habitual explanations serve to mask fundamental problems of meaning, structure and design. For many projects in trouble, an equally accurate explanation is that the conceptual framework was so faulty that it led to a data model which did not completely, coherently and flexibly represent the environment.

Furthermore, the focus on user requirements channels stakeholders toward unrealistic notions of how stable their environment is. Managers who must advance their project by signing off on a specification are naturally under pressure to believe that it will satisfy their needs; the situation does not encourage them to pursue long-term thinking about possible change. But organizational theorists recognize that human service agencies have lifecycles (Brothers and Sherman, 2012), as do programs and their evaluation efforts (Trochim, Urban, Hargraves et al, 2012). Early stages of a program's lifecycle involve rapid change. Information systems are an integral part of the cycle. When staff collect and then analyze data in an information system, they create a feedback loop that leads to changing the program—and eventually, therefore, to requiring changes in the kind of data that need to be collected.

A particularly difficult kind of change occurs when the boundaries of an agency's work shift. Organizations often develop new programs or realize that they need to integrate data between programs or with external agencies. Where a data exchange protocol can be envisioned to support a defined business process, the National Information Exchange Model (NIEM) can bridge boundaries (Coursen, 2011). But for an integrated service environment, there is no substitute for a completely integrated information system. Few agencies attempt that, and even fewer have achieved it well.

These headaches add up to a syndrome of frustrated information management throughout the sector. Unforeseen happenings during the software development process cause delays and cost overruns, and may lead to cancelling a project. Unforeseen happenings after software has been officially completed often result in a system being abandoned after a few years, or allowed miserably to limp along, unable to be modified, one isolated silo among many as the agency supplements it by adding other systems with other narrowly defined functions. Of course, these woes are not universal: when talented designers and committed stakeholders work with adequate resources in a relatively stable environment, they can create successful systems. But such bright spots are rarer than they should be.

A Way Forward: Developing Strong Domain Models

What can the sector do about weak concepts that sabotage information systems? A stock answer would be that agencies need to build good enterprise data models. But on the basis of what? In an enterprise model, "business entities... are named using business terms." (Mosley, Brackett et al, 2009, p. 76) The problem is that human service business terms are unsatisfactory material to start with!

Here the notion of a domain model is helpful. The term contains a whiff of aspiration to represent or build consensus about what is common among all settings of a particular type. For example, the domain model of sales orders—used for widgets and other things—would define certain entities (buyer, seller, product, sales order, and detail line), how they relate to each other and their most important attributes. Sales orders have an extraordinarily strong domain model. It is based on a preexisting consensus, as evidenced by the paper forms used by millions of vendors before the computer era. With miraculous conformity, all follow the same semantic convention.

If the human services had semantic conventions as satisfying and universally accepted as those of the commercial sales order, the problems described above would not exist. Programmatic staff and evaluators would share a common and stable terminology; database designers would have a secure basis for sound decisions; there would be mature software products that seamlessly spanned the work of heterogeneous practice areas in multi-service agencies; the kinds of flexibility needed for inevitable change and innovative new programs would be well understood; and data exchange between cooperating agencies would be as robustly defined as electronic medical records are. Moreover, stakeholders approaching information systems could reasonably expect to find data constituting a usefully interrogable model of the human service environment itself.

Alas, there are no strong human service domain models. Major resources (e.g. Hay, 1996, 2011; Silverston, 2001a, 2001b) suggest models for commerce, accounting, manufacturing and other domains—but not the human services. And no professional institution appears to have attempted to build domain models in this sense. (The National Information Exchange Model offers a framework for defining exchanges of human service data, but it does not seem to critique or prescribe core concepts.)

This is a chicken and egg situation: strong domain models promote clarity and consensus, but a reasonably high level of clarity is a prerequisite for constructing the domain models in the first place.

There is, however, a way to break through the impasse. It begins from the recognition that human service concepts are socially constructed. If inadequate or inconvenient for organizing data, the concepts themselves can be reformulated.

Established software development methodologies rarely contemplate the fundamental restructuring of stakeholders' terminology. Yet that is what the human service sector needs. And by reordering software development stages and roles, it can be done. The process of eliciting software requirements can be transformed into an opportunity for more accurately modeling the deep structure of the human service environment. By embracing insights that arise while modeling data, stakeholders can arrive at new conceptual frameworks that provide stronger bases for managing information.

Bridging the Gap between Requirements Elicitation and Data Modeling

The software development industry pays considerable attention to requirements elicitation, recognizing that errors in requirements typically have serious consequences downstream and are far cheaper to fix when caught early (Christel and Kang, 1992). At the same time, data modeling theorists note that the impact of data modeling on the quality and success of an information system—including its cost, ability to meet stakeholders' needs, flexibility and capacity to be integrated with other systems—is greater than any other phase of software development (Moody and Shanks, 2003). These have been, for the most part, two separate conversations. It is as if they had to do with two separate problems.

The divide seems to have its roots in a widespread belief that systems analysts collect requirements from stakeholders, while the job of data modelers is simply to embody those requirements in a database. According to that view, the data model is an arcane technical document with which most project stakeholders never interact. That is an accurate account of how most projects are carried out, but it is based on ignorance of the kind of thought that good data modeling involves. As a result, it fails to take advantage of potentially powerful insights.

Expert data modelers do not merely reflect stakeholder requirements. Because a misnamed entity or

attribute can foul the model, they are extraordinarily careful about the clarity and scope of terminology. They will examine a concept's implications and consider alternatives, trying to discern the possible downstream effects it might have on design. They are deeply concerned with flexibility, i.e., the model's ability to accommodate variations over time or between different instances of a system without the model itself needing to be changed. In pursuit of flexibility, modelers often use abstraction and generalization to introduce concepts into the model that were not directly mentioned in stakeholder requirements. Empirical research has found that experienced modelers introduce useful new concepts more frequently than novices do; they also create models that are more flexible (and thus sustainable), more easily comprehensible to stakeholders, and of higher overall quality (Shanks, 1997). They do this by spending more time holistically understanding problems; categorizing problems into standard abstractions; and drawing on their previous experience of similar situations (Batra and Davis, 1992).

In fact, to do their job well, experienced data modelers intentionally avoid being too completely drawn into the worldview of documented user requirements. The secret heresy of expert data modelers is that their most important thinking is done while not modeling the requirements at all. Rather, the modeler is looking through the requirements to perceive the structure of the environment itself.

As the data modeler clarifies concepts or introduces new ones that were not directly mentioned by stakeholders, what effect will that have on the requirements? The answer depends on the way the project and roles within it have been organized. If all requirements have already been finalized without the data modeler having interacted with stakeholders, then the modeler's subsequent work will not affect the requirements. If, however, the data modeler's thinking is deliberately fed back into the requirements elicitation process at an early stage, it can transform the stakeholders' understanding of their business and needs.

That may sound strange. Using data models directly with stakeholders is a minority approach. Popular resources (e.g., Hossenlop and Hass, 2008) omit data modeling from the requirements elicitation process entirely, and in a study of a small group of internationally recognized experts on requirements elicitation, only one of the nine participants mentioned using data models (Hickey and Davis, 2003). Conventional wisdom suggests that data models are too abstract for non-technical stakeholders to digest, and general domain models are more commonly recommended for educating analysts—not users—about the domain (e.g., Christel and Kang, 1992). But on the opposite end of the spectrum, David C. Hay (2011) names communication between users and analysts as one of the primary purposes of data modeling, because it can focus the attention of both on the most fundamental issues of the business—in other words, because it is a good way of modeling the domain.

For the human service sector, which faces fundamental problems of conceptual clarity and unstable system boundaries, the integration of domain modeling into the requirements elicitation process can be an extraordinarily powerful tool.

A Method for Reformulating Core Concepts

This author has employed this approach in the design of over a dozen human service and justice sector information systems. Together, the experiences suggest a coherent method based on three principles.

First, insofar as possible, project managers should assemble representatives from all the far-flung constituencies that may depend on the system—front-line workers, supervisors, performance

measurement staff, evaluators and funders—at the table. This may seem obvious but many human service software projects fail to do it.

Second, these stakeholders must agree that the first stage of eliciting requirements will not be to document the requirements per se, but rather to arrive at a domain model.

Third, the group process needs to deliberately create a dynamic tension between the known and the possible, and between the specific and the general. Rather than focusing too narrowly on the particular setting for which they are building software, the group must, as it were, try to look through the local situation toward the commonalities that exist in human service settings of that general type.

For example, in creating the domain model for Bosnia-Herzegovina's court automation system, participants worked within rules including:

- refer when possible to published data model patterns;
- focus on the meaning of entities, relationships and attributes, discussing work processes only insofar as they directly affect those data model components;
- focus on the local reality first but only as a springboard to broader generalizations;
- focus on the current reality first but only as a springboard to discussing potential needs for flexibility. (Coursen and McMillan, 2010)

How does this happen in practice? A conference room with a whiteboard is a good venue. Working outward from the most central concepts, the modeler helps the group examine their meaning. As the stakeholders describe requirements and mention entities, the modeler proposes—and verbally articulates—a fragment of a domain model that can represent that area. The fragment may come from an existing reservoir of patterns (such as a publication or the modeler's prior experience), or it may be a new proposal not tried before.

The modeler then leads the stakeholders through a critique of both the proposed fragment and the stakeholders' expressed requirements. Is the meaning of each entity clear? What about the meanings of its defining attributes? Is the fragment correct, evolvable and sufficiently holistic? Have the users understood their environment in a comprehensive enough way that takes into account possible future change? The model interrogates the requirements while the requirements interrogate the model. If both are found to be satisfactory, the stakeholders provisionally validate them. If there are doubts, then the stakeholders revise their understanding, the modeler proposes a change, and the process continues cyclically.

An essential tenet of this method is that accepted business terms are not sacred. Anyone is free to propose that an existing concept be thrown out and replaced or modified. As a result, the process may require stakeholders to significantly change their terminology and adopt a new view of their work. For some at the table, this may be a stumbling block. But as the following three examples demonstrate, reframing core concepts can provide insights that lead to re-engineering practices in need of change.

Exhibit A: Needs for Navigation and Advocacy

Community Health Advocates (CHA), previously known as the Managed Care Consumer Assistance Program, is a program of the NYC Community Service Society that provides individualized health care assistance through a network of subcontracted nonprofit organizations. Focusing on high-need populations, the program helps people enroll in free or low-cost insurance, maintain their coverage, and access health care services. Organizations in the network record data about clients, cases and

outcomes into a central database so that CHA can oversee and analyze the program.

In 2005, CHA engaged this author to help define the scope and structure of a new information system. The project integrated data modeling with the requirements elicitation process, using an old information system (slated for replacement) as the starting point to identify and critique the program's existing conventions. The core entities were the client, the case(s) belonging to the client, and the service task(s) carried out to resolve each case. At the time of its opening, each case was categorized according to a pick-list of the issue that the client presented.

The first questions for the stakeholder group were: "Each client can have one or more cases; each case is characterized by exactly one presenting issue; each client can therefore have more than one open case simultaneously as long as they are characterized by different questions or problems; each service task addresses exactly one of these cases. Why is it that each case should be characterized by a single question or problem? Is that necessary, optimal, or problematic?" That line of inquiry quickly revealed that other organizations were accustomed to thinking of a case as potentially covering multiple issues; that users found it unintuitive to define each presenting issue as a separate case; and that it made workflows awkward because a single service task might really address the issues within more than one case, and would therefore need to be entered multiple times. The modeler proposed an alternative: a case could have multiple presenting issues, and a service task could address more than one of them. The group agreed.

The group then turned to the list of presenting issues. Users had complained about the existing pick-list, finding that some clients' situations were not covered, while other situations could arguably straddle more than one of the choices. (For a complaint about billing, should Billing or Complaints be chosen?) The pick-list was a confusing jumble. Incoherent taxonomies of this kind are a common problem for information management (Chisholm, 2012).

Thinking about this, the group came to a gradual deep insight: the pick-list had problems because presenting issue was the wrong concept to begin with. The program's purpose was to help clients navigate a complex and confusing healthcare system. Because clients had limited information about how managed care works, they often described their situations in fragmentary and uncertain ways. Caseworkers usually needed to carry out an investigative process leading from the supposed presenting issue to identifying one or more underlying problems in the client's relationship with the healthcare system. Rather than focusing on the presenting issues, a better way to think of a case was as the unfolding discovery of a set of needs that the client might have at different levels over time. The appropriate concept, the group realized, was the identified need.

The next step was for CHA to apply this new model to their work. Because a central concept had changed, a new taxonomy was required. After considerable work, the group arrived at four large categories of identified needs: information about the health care system; medical services; help with routine administrative matters; and help removing larger administrative roadblocks. The group then broke each of these four large categories down into more specific types. Based on the new taxonomy, CHA then exhaustively revised the program's workflows and business rules.

When CHA went to seek a software vendor, the request for proposals included the domain model (see Appendix) and the taxonomy of identified needs. Those helped prospective bidders understand the scope of the work. Conversely, bidders' responses helped CHA gauge how well they had understood it. The chosen firm, Metis Associates, developed a software system over the next year. At the end of the project, CHA's leaders were glad to conclude that the result considerably exceeded their original hopes.

Exhibit B: How Participants Flow Through a Program

The Vera Institute of Justice creates demonstration projects, i.e. programs that test innovations intended to improve the justice system. Like other programs, a demonstration project in the early stages of its lifecycle goes through rapid change. To inform that early evolution, high quality data is critically important. The challenge is to design an information system that can collect that data while also being sufficiently modifiable, because when the program changes, the software must also change. And after the program has finally coalesced around stable operating procedures, the same information system needs to be able to collect data that will be useful for evaluating the program's impact over the long term.

In the early 2000s, the Institute was preparing to launch several demonstration projects with different goals, target populations and programmatic models. For example, one offered community-based supervision as an alternative to juvenile incarceration, another provided substance abuse treatment to court-involved juveniles who might already be detained, while yet another worked to reduce recidivism among young offenders being released from jail. To lower the cost and shorten the timeline of software development, the Institute decided to design a system that could be adapted for use in each different project and could remain adaptable as each evolved.

The key would be to find common patterns that could flexibly represent all of them. But as the Institute's designers shuttled between the different projects, their parallel conversations with stakeholders merely showed what they already knew: each of the projects was developing its own protocol for how to obtain participants, how to work with them, and how to close out their involvement with the project. The design team tried to float sketches of possible data models, hoping to find a common denominator. But as long as the conversation revolved around the concept of a case, none emerged.

Some stakeholders, though, grasped for other terminology. The word transition showed up, and so did the word status: a participant would transition from being in the program to successfully completing it, or from being in one status to another. Homing in on the words status and transition, the design team proposed that those might be the core concepts the group had been looking for. Perhaps any human service program could be mapped as a set of possible transitions between statuses? To some of the group, this sounded odd and a bit abstract. But they were willing. Using a white board, the group began experimenting to create a new kind of flow diagram. To see whether it could be fit into a very simple data model, the group agreed to tightly limit the map to statuses, transitions, and the permissible reasons why they could occur. (Left aside was the kind of procedural information that a flowchart or UML diagram would include.)

Unresolved questions immediately began to surface. The stakeholders had never needed to develop a coherent model of the universe of participants and how one's relationship with the program could change over time. Creating the map forced them to do that. The exercise slowly led the group to clarity and consensus about criteria for admission, stages of participation and major milestones, how to deal with unexpected events in participants' lives, criteria for successful and unsuccessful completion, and the possibility of clients returning to the program. After the first Vera Institute project created its map, the other projects followed suit.

The rules and notation for creating this kind of map have been formalized (Coursen and Ferns 2004a, 2004b). Below, Figure 1 is an example of one for a fictional methadone maintenance and detoxification clinic. Each person begins in the status of an unscreened potential participant. Arrows represent permissible transitions, and the numbers within the arrows refer to a list of permissible reasons. A complete sequence of transitions constitutes a cycle: an arrow with a circle opens the

client's cycle, and an arrow with a bar closes it. And potential participants who have not yet been enrolled are formally represented as within the ambit of the program; thus an abortive intake nonetheless constitutes a cycle.

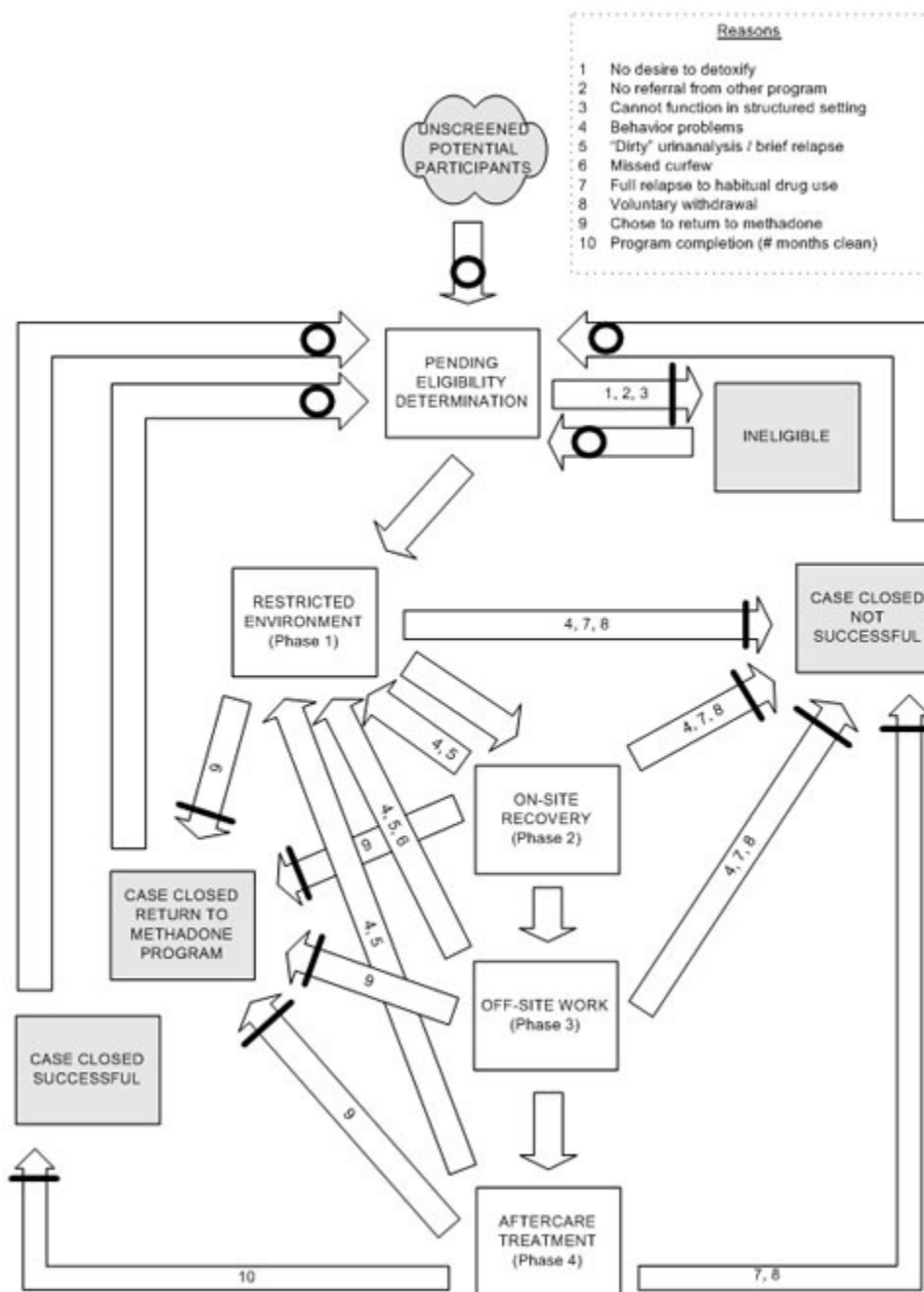


Figure 1: Status-Transition-Cycle Map of La Casita. From Coursen and Ferns (2004b). Reprinted with permission.

The concepts of status, transition and cycle replace existing conventions about the program. They offer a very complete and precise way of modeling a program's participant flows, rather than considering that clients simply enter and exit (cf. Rutman and Mowbray, 1983; Lohmann and Lohmann, 2002). They are akin to the stocks and flows used in system dynamics modeling, and since they are tied to a data model, they offer a standardized way to collect data—whether from a single

program or from many interacting programs—in support of sophisticated planning techniques (cf. Taber and Poertner, 1981; Milstein, 2008).

For managing data about individual participants in an information system, the data model (see Appendix) states that each participant may have one or more cycle, and each cycle consists of more than one transition. These entities thereby replace one function of the case. Whether in medicine, justice or the human services, the case manages three kinds of information: it delineates participation, it aggregates smaller substantive components, and it tracks trajectory (see Figure 2 below). After the Vera Institute embodied this approach in its software, many users of the system stopped even speaking of cases. The new terminology had given them a more powerful set of organizing concepts.

SECTOR	Justice	Medicine	Human Services
FUNCTION			
Delineating Participation	defendant prosecutor/plaintiff court judge witnesses jury other government actors	patient physician other service providers patient's family	client service provider other service providers client's family/friends
Aggregating Substantive Components	charges civil claims verdicts settlements sentences	symptoms diagnoses treatments progress outcomes	client issues service plan progress outcomes
Tracking Trajectory	hearings filings	intake medical appointments referrals discharge	intake service sessions referrals case closure

Figure 2: The functions of the concept of a case. From Coursen and McMillan (2010)

Exhibit C: The Programmatic Ecosystem

The Vera Institute's demonstration projects needed to track interactions involving people and organizations of many types, not just program participants. There were staff, family members, programs to which clients might be referred, agencies in the legal and healthcare systems, staff at those programs and agencies and others. For each project, though, that constellation of actors was unique. Again the projects needed a flexible and holistic data model, and again the conventions of the human service sector provided poor material to work with.

A traditional medical chart belongs to a single patient, and organizes interactions as dyads between the client and physician. That has become the convention used in the design of most human service databases. Typically, the client is a central entity and the staff is another independent entity. Other people and organizations, as well as the services provided, are modeled in a way that depends on the client. This is limiting because it does not allow the system to holistically represent the way that people often appear in different roles (e.g. client and then family member, or lawyer for several different clients) and events over time.

In fact, this current convention is out of step with the broader understanding of the social work profession. The ecosystems perspective considers a person's situation to be a complex system of interconnected phenomena (Meyer and Mattaini, 1998). Practitioners often represent a client's

ecosystem using a diagram called an ecomap which includes all the important people and institutions with which the client interacts (Hartman, 1978).

The current convention is also primitive by the standards of data management in other fields. Enterprise data models have well-established and much more sophisticated conventions for representing this situation: there is a hierarchy that descends from the most generic (party) to more specific (person or organization) and then to even further levels of specificity. Such models represent the fact that particular parties are in certain roles at particular times, and how parties may be related to each other. There are various kinds of events, and multiple parties may participate in an event. These modern conventions are clearly applicable to the human service sector and they dovetail well with the ecosystems perspective.

Software designers at the Vera Institute therefore implemented a version of this data model (shown in the Appendix) which in many respects follows Hay's (1996) work. Then, in order to connect this modern model to the specific concerns of the Institute's stakeholders, the analysts designed a new visualization tool, the programmatic ecomap (Coursen, 2006). In the group setting, stakeholders would do an inventory of the types of person, organization and event that were relevant to their project. Below, Figure 3 is the ecomap of Adolescent Portable Therapy, a project which provides substance abuse treatment to incarcerated youth. The ovals, lines and rectangles correspond directly with the party types, relationship types and event types that will populate the database.

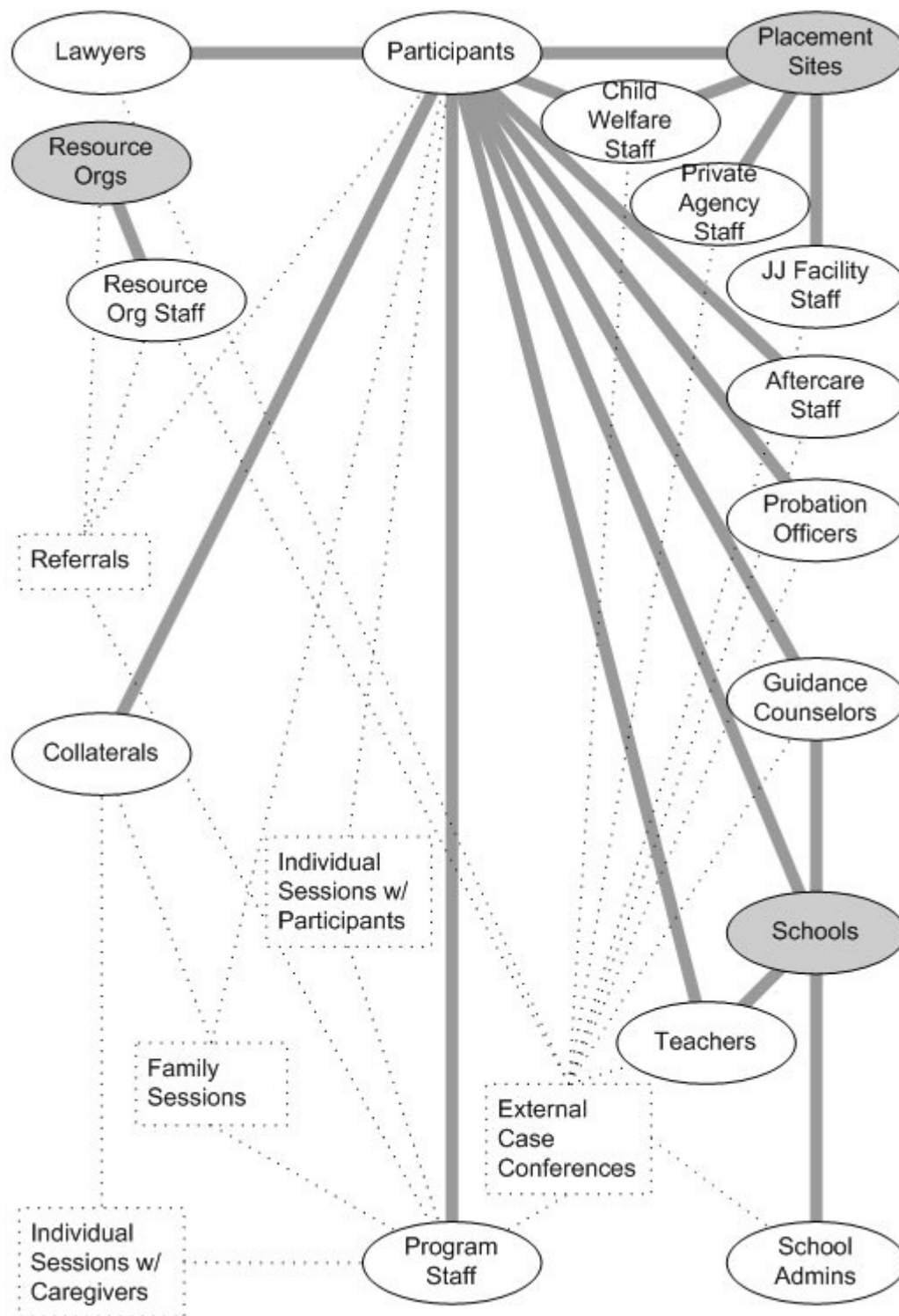


Figure 3: Programmatic ecomap of Adolescent Portable Therapy. From Coursen (2006). Reprinted with permission of the publisher (Taylor & Francis Ltd, <http://www.tandf.co.uk/journals>)

The exercise of creating the ecomap specifies what the data model will represent in a specific programmatic environment and points toward areas that need to be extended for that environment. It stimulates the stakeholders to think about their program's structure and how the software needs to support its workflows. And it helps stakeholders envision how their program interacts with other

institutions and actors in their clients' lives.

Toward a Sector-Wide Conversation

In each of the three examples above, the stakeholders and the modeler worked together to reformulate human service concepts by integrating data modeling with early requirements analysis. Each of the three fragments was created with an eye toward a general domain model, attempting to look through the local situation toward broader commonalities. As a result, each of the three is potentially applicable in a range of settings.

Other modelers who look at these fragments may perceive weaknesses or limitations and so advocate different solutions. If they do so publicly, then the purpose of this article will have been accomplished. One of the innovations of these fragments is the simple fact that their background and justification have been published. After all, domain models are only possible and useful insofar as they represent the shared thinking of practitioners in a field. In order for the human service sector to have strong models, the deficiencies of existing models need to be discussed openly as a public problem.

These three fragments are just a small beginning. For the sector to arrive at a complete human service domain model, many other subject areas—e.g., services, referrals and outcomes—need to be addressed. Furthermore, each specific practice area such as homeless services or child welfare contains elements relevant to its own context which must be modeled separately. And there is a need for an overarching model that can integrate different programmatic interventions and funding streams within a single system.

The pursuit of a complete human service domain model calls for a kind of conversation that the sector has never had. It requires a willingness to challenge and, when necessary, reformulate or abandon cherished terms of art. The software development profession will need to rethink its focus on stated user requirements and move toward the creation of flexible and interrogable models of the human service environment. There will need to be forums for offering and critiquing alternative models. And all constituencies—funders, managers, front-line service providers, performance measurement staff and evaluators—will need to be at the table.

Appendix: Fragments of Domain Models Discussed

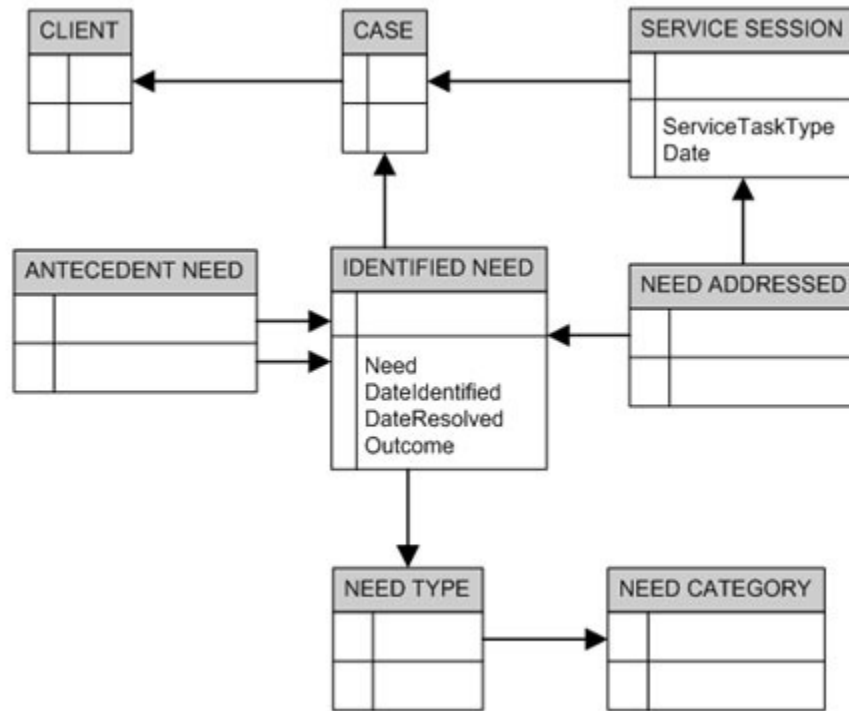


Figure 4: Identification of needs for navigation and advocacy

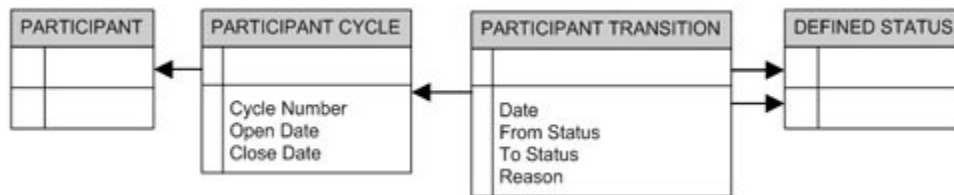


Figure 5: Participant flows

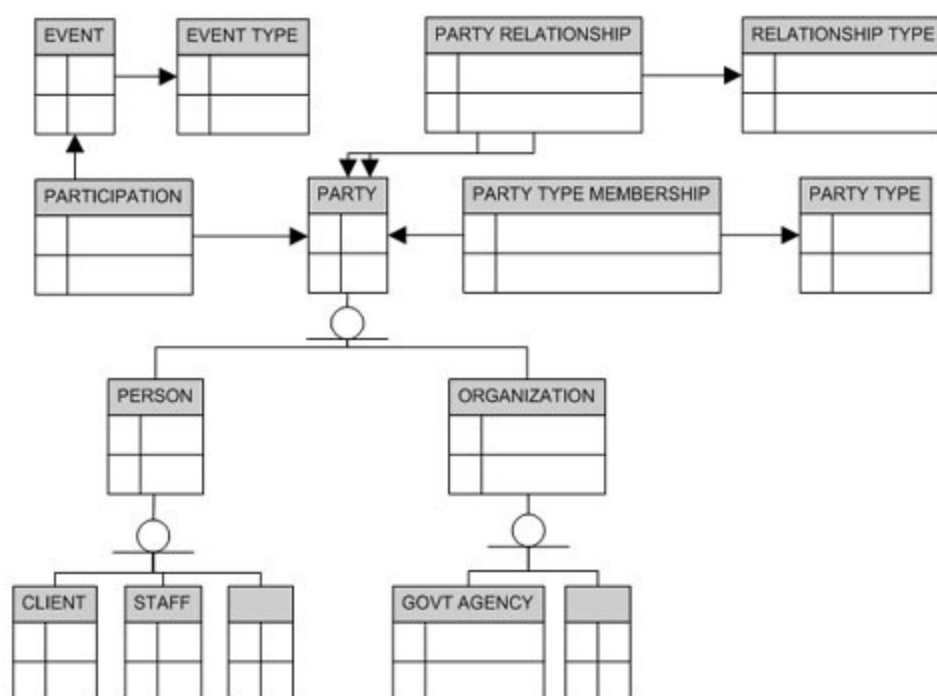


Figure 6: The programmatic ecosystem

References:

- Batra, D. and Davis, J.G. (1992) 'Conceptual data modeling in database design: similarities and differences between expert and novice designers', *International Journal of Man-Machine Studies*, Vol. 37, pp. 83-101.
- Brothers, J. and Sherman, A. (2012) *Building Nonprofit Capacity: A Guide to Managing Change Through Organizational Lifecycles*. San Francisco: John Wiley & Sons.
- Chisholm, M. (2012) 'The Celestial Emporium of Benevolent Knowledge', *Information Management* (17 February). Available at www.information-management.com [Accessed 15 March 2012].
- Christel, M.G. and Kang, K.C. (1992) *Issues in Requirements Elicitation* [Technical Report CMU/SEI-92-TR-12]. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University.
- Coursen, D. (2006) 'An ecosystems approach to human service database design', *Journal of Technology in Human Services*, Vol. 24, No. 1, pp. 1-18. Available at https://files.nyu.edu/dac229/public/Coursen_JTHS_ecosystems_approach.pdf [Accessed 15 February 2012].
- Coursen, D. (2011) 'A Route to Streamlined Performance Reporting and Stronger Information Systems: Nonprofit Human Service Agencies and the National Information Exchange Model', *Data Administration Newsletter* (October). Available at <http://www.tdan.com/view-articles/15551> [Accessed 15 February 2012].

Coursen, D. and Ferns, B. (2004a) 'Modeling participant flows in human service programs', Journal of Technology in Human Services, Vol. 22, No. 4, pp. 55-71. Available at https://files.nyu.edu/dac229/public/CoursenFerns_JTHS_modeling_participant_flows.pdf [Accessed 15 February 2012].

Coursen, D. and Ferns, B. (2004b) 'Modeling participant flows as a basis for decision support in human service programs', Proceedings of the 10th Americas Conference on Information Systems. Available at <https://files.nyu.edu/dac229/public/SIGDSS03-1476.pdf> [Accessed 15 February 2012].

Coursen, D. and McMillan, J. (2010) 'A framework for logical data models in courts', Data Administration Newsletter (December). Available at <http://www.tdan.com/view-articles/14749> [Accessed 15 February 2012].

Fitch, D. (2007) 'Designing Databases Around Decision Making'. In M. Cortes & K. Rafter (Eds.), Nonprofits and Technology: Emerging Research for Usable Knowledge (pp. 135 – 147). Chicago: Lyceum Press.

Hanleybrown, F., Kania, J. and Kramer, M. (2012). 'Channeling change: making collective impact work', Stanford Social Innovation Review (26 January). Available at http://www.ssireview.org/blog/entry/channeling_change_making_collective_impact_work [Accessed 15 March 2012].

Hartman, A. (1978) 'Diagrammatic assessment of family relationships', Social Casework, Vol. 59, No. 8, pp. 465-476.

Hay, D. C. (1996) Data Model Patterns: Conventions of Thought. New York: Dorset House.

Hay, D. C. (2011) Enterprise Model Patterns: Describing the World (UML Version). Bradley Beach, NJ: Technics.

Hickey, A.M. and Davis, A.M. (2003) 'Elicitation technique selection: how do experts do it?', Proceedings of the 11th IEEE International Requirements Engineering Conference.

Hossenlop, R. and Hass, K. (2008) Unearthing Business Requirements: Elicitation Tools and Techniques. Vienna, VA: Management Concepts.

Kettner, P., Moroney, R. and Martin, L. (1999) Designing and Managing Programs: An Effectiveness-Based Approach, 2nd ed. Thousand Oaks, CA: SAGE.

Lohmann, R., and Lohmann, N. (2002) Social Administration. New York: Columbia University Press.

Meyer, C. H., & Mattaini, M. A. (1998) The Ecosystems Perspective: Implications for Practice. In M. A. Mattaini, C. T. Lowery & C. H. Meyer (Eds.), The Foundations of Social Work Practice (pp. 3-19). Washington, DC: NASW Press.

Milstein, B. (2008) Hygeia's Constellation: Navigating Health Futures in a Dynamic and Democratic World. Centers for Disease Control and Prevention, Syndemics Prevention Network. Available at http://www.cdc.gov/syndemics/pdfs/Hygeias_Constellation_Milstein.pdf [Accessed 15 March 2012]

Moody, D. and Shanks, G. (2003) 'Improving the quality of data models: empirical validation of a quality management framework', Information Systems, Vol. 28, pp. 619-650.

Mosley, M., Brackett, M., Early, S. and Henderson, D. , Eds. (2009) *DAMA Guide to the Data Management Body of Knowledge*. Bradley Beach, NJ: Technics.

Nathan, R. P. and Ragan, M. (2001) *Federalism and the Challenges of Improving Information Systems for Human Services*. Albany, NY: Nelson A. Rockefeller Institute of Government. Available at <http://www.rockinst.org/> [Accessed 3 May 2011].

Ragan, M. (2003) *Building Better Human Service Systems: Integrating Services for Income Support and Related Program*. Albany, NY: Nelson A. Rockefeller Institute of Government. Available at <http://www.rockinst.org/> [Accessed 3 May 2011].

Rutman, L. and Mowbray, G. (1983) *Understanding Program Evaluation*. Beverly Hills, CA: SAGE.

Shanks, G. (1997) 'Conceptual data modeling: an empirical study of expert and novice data modelers', *Australasian Journal of Information Systems*, Vol. 4, No. 2, pp. 63-73.

Silverston, L. (2001a) *The Data Model Resource Book, Vol. 1: A Library of Universal Data Models for All Enterprises*. New York: John Wiley & Sons.

Silverston, L. (2001b) *The Data Model Resource Book, Vol. 2: A Library of Universal Data Models for Specific Industries*. New York: John Wiley & Sons.

Simpson, J.A. and Weiner, E.S.C. (1989) *Oxford English Dictionary*, 2nd ed., Oxford: Clarendon Press.

Taber, M. and Poertner, J. (1981) 'Modeling service delivery as a system of transitions: the case of foster care', *Evaluation Review*, Vol . 5, No. 4, pp. 549-556.

Trochim, W., Urban, J.B., Hargraves, M., Hebbard, C., Buckley, J., Archibald, T., Johnson, M. and Burgermaster, M. (2012) *The Guide to the Systems Evaluation Protocol*. Ithaca, NY: Cornell Digital Print Services. Available at <http://core.human.cornell.edu/research/systems/protocol/index.cfm> [Accessed 10 March 2012].

[Go to Current Issue](#) | [Go to Issue Archive](#)

Derek Coursen - Derek joined Public Health Solutions in 2007, and currently directs planning and informatics around the administration of New York City's portfolio of HIV care and prevention contracts. He was previously Director of Information Management at the Vera Institute of Justice. His research focuses on the holistic modeling of public service settings with the goal of designing more evolvable information systems that support performance measurement and evaluation as well as operations. Derek earned M.S. degrees in management (New York University Wagner School of Public Service), information systems (Pace University) and information and library science (Pratt Institute). He may be contacted by email at dcoursen@healthsolutions.org or by phone at (347) 401-3649.

Quality Content for Data Management Professionals Since 1997

© Copyright 1997-2012, The Data Administration Newsletter, LLC -- www.TDAN.com

TDAN.com is an affiliate of the [BeyeNETWORK](#)