# Deriving Service Principles Relating Lean in-the-large Systems to Autonomous in-the-small Entities

Jay Ramanathan,
jayram@ohio-state.cse.edu
Rajiv Ramnath,
ramnath@ohio-state.cse.edu
Randall Glassgow
GlassgowRandallS@JohnDeere.com
CETI (CERCS for Enterprise Transformation and Innovation)
Work sponsored by IBM faculty Innovation Grant
OSU-CISRC-5/07-TR37
The Ohio State University
Columbus
Ohio 43210
2/8/2007, Revised 5/22/2007

## 1. Abstract

Existing many-to-many relationships between Business and IT services are complex to manage and challenge the agility of an enterprise. While many best practices for services support and delivery exist, they are limited methods for *service improvement*. In addition, existing improvement methods are not designed for Business and IT alignment. In this paper, a case study in IT for the healthcare industry is used to illustrate a widely applicable Adaptive Complex Enterprise framework for improving the deliverables of a complex organization of entities (humans, processes, software systems, assets etc.). The underlying representation scheme also provides a structure that permits the application of the Lean methods by quantifying and simulating the interactions between the *global* system and its *local* autonomous (self-managed) entities. We show examples of how the method for system improvement provides information for Lean decision making and to identify and prioritize service changes to accomplish performance objectives. The framework is applied to develop a roadmap to reduce the time to install new PCs from 17 days to 1 day. The method application also illustrates how to define business SLAs and relate them to OLAs. Framework-based principles for decision-making and management illustrate the wide applicability and an engineering approach to the improvement of complex systems.

## 2. Business and Service Improvement Challenge

The IT (Information Technology) department of a health-care provider with a billion plus in operating revenue has the goal of reducing the personal computer (PC) deployment response from 17 days to 1 day. Eight of the 240 fulltime employees in IS are dedicated to new installations and replacements. Over 100 new PCs are installed each month. The PCs serve numerous different purposes in the hospital – from use at a nurse's station to equipment monitoring. Thus many variations are deployed. Finally, there are enterprise software systems in place – a customer relationship management (CRM) is used for managing the Customer Service Center (CSC) tickets. Asset management, work order, and imaging systems are all utilized. As is typical with most institutions, these systems are not integrated.

Separate teams provide the following services: service desk, request management, inventory management, approvals, billing, deployment, assembly, and imaging/engineering. Each plays a *role* in the deployment of a PC request. The partial flow for the deployment of a PC can be seen in Figure 1.

The case study and project objectives were to:

- Identify and prioritize the specific steps to improve service quality and to reduce the time to deploy from 17 days to 1 day
- Develop a repeatable, teachable and scalable methodology for the co-engineering (simultaneously engineering) business, organizations and IT services for reducing the overall cost of other service improvement programs within the organization
- Enable continuous incremental improvement through a monitoring and management framework
- Leverage existing best practices (Lean, ITIL) and technology (CMDB[i], autonomic computing) as appropriate
- Clearly identify the benefits of a systems engineering approach for the business.

As with most IT service organizations, there were several challenges to overcome. The large variation in requests and related problems had caused the existing process to evolve over time to execute the worst-case scenario each time! Examples of this include waiting for approvals (even for routine requests), traveling to the site every time to survey it for proper wiring, and so on.

Decision-making is made all the more challenging as improvement projects vie for resources. And, last but not the least; externally-driven trends have to be considered. Examples include suppliers offering competitive PC refresh services at a fraction of the cost and site monitoring technologies becoming more viable.

## 3. Research in Service Improvement Methods

In this section we summarize our findings of background research *both* in academic publications and industry best practices. Specifically, we focus on design and improvement of processes serviced by human and non-human agents. We define such systems to be *mixed mode.* Mixed mode systems and process are made of:

° services provided in the physical world (e.g. inventory management, assembly, installation),
° services provided in the electronic IT world (e.g. help desk, inventory software) and
° services provided by human decision-making using the IT services (e.g. applying processes and rules to triaging who to escalate to).

Mixed mode systems have to consider differences between the physical and the electronic world. For example, in the physical world, a resource (that is not a utility) is not likely to be shared "simultaneously" across work centers. In the IT world "virtual entities" (e.g. a CRM system, a server) can be shared across different "**e-Workcenters"** for cost effectiveness. We define an *entity* to be a process, a human resource, a software system, or an asset. We also think of entities as being locally managed and *autonomous* – that is all decisions for improvement etc. are made locally its management agent.

While both practitioners and researchers have identified the need to align business, organizations and IT [15], the development of engineering methods to achieve better performance in mixed mode systems is a new research area.

### Academic research

Much of the research in process improvement aligning business and IT services remains disciplinary focused. Recent works in IT architectures, patterns and autonomic systems (see for example [3,11,9] focus primarily on technology implementation rather than the behavior of the eventual process. While autonomic concepts are a promising way to reduce management complexity, we need architecture principles for building complex systems [10, 17] within which human agents and components interact.

Work in Industrial Systems Engineering, specifically research in developing Lean *resource effective* systems [4,7] is very relevant. However this body of work is focused on improving the behavior of the physical shop floor processes without offering a prescription on how to incorporate the use of IT services and technologies to make the system more effective.

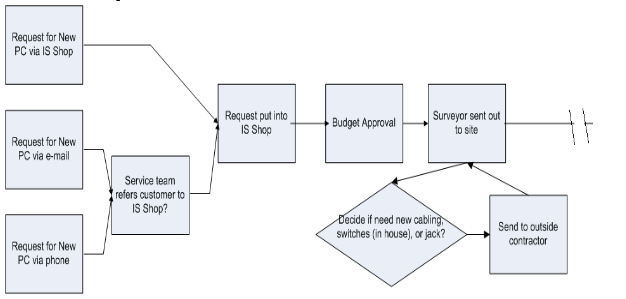

Request for New PC via IS Shop
Request for New PC via e-mail
Request for New PC via phone
Service team refers customer to IS Shop?
Request put into IS Shop
Budget Approval
Surveyor sent out to site
Decide if need new cabling, switches (in house), or jack?
Send to outside contractor

**Figure 1: As-is (partial) PC install process flow illustrates non-value add (e.g. budget approval) steps. This also illustrates that existing processes are complex and have often evolved to handle the most difficult cases. At the same time they also continue to handle the simplest cases, leading to inefficiencies for simple cases.**

### Relevant Best Practices

Best practices [5,6,8,12] are typically presented as process step and do not dictate a specific representation scheme of the object that is to be improved. (This is rightly so, in order to remain widely applicable). This is true of the two recent examples gaining momentum - ITIL[ii] and TOGAF[iii]. These practices do not prescribe the representation of the specific enterprise. This also leaves much to the individual's interpretation in a system improvement program and, thus, each project is done on a case by case basis, often by different system integrators.

The third relevant best-practice is Lean. A common starting point is a *value-stream* representation. This is the representation of the process made of physical work centers and the possible flows/routings of requests (work orders, incidents etc.) so that work centers add value.

Here *value-add* is defined from the customers' perspective. Flows are improved by identifying the value-added and the non-value-added time. One example of non-value added time is when requests are queued at work centers waiting for resources. *Lean Principle* dictates that wasted time can be systematically eliminated resulting in better use of resources. The representation is also used by decentralized multifunctional teams to identify local improvements.

Since Lean has been primarily applied to physical systems we need to generalize [13,14,16] the well-established methods for the physical factory design to methods that work for mixed mode systems. More generally, we need a precise way to represent an enterprise so that we can apply different related business and IT best practices (e.g. Lean and ITIL) to the *single, shared representation*. We also need a *structure* for performance measurement and analysis which is *universal* to interdisciplinary (i.e. Business and IT) practices. We propose such representation concepts and patterns next and later illustrate it use with the PC Install case study.

## 4. ACE (Adaptive Complex Enterprise) Représentation Patterns

While it may seem reasonable to transfer successful system engineering principles applicable in the 'physical' world to the 'electronic' world[1], there are some questions that have to be answered. For example, what are universal patterns that can be used to represent a business so that the value of IT services can be identified and incrementally improved? We next present interrelated performance-focused patterns and propose a representation scheme using these for Lean service-oriented enterprise architectures.

### Triage Pattern: Separates routine from non-routine requests for efficient processing

*Problem:* The problem addressed is the separation of the routine from the non-routine requests for the most effective assignment of resources for each request.

*Analysis:* Service-oriented organizations typically have a large *variation* in incoming *requests* (orders, work orders, complaints, status, quotes, confirmation etc.). Also, incoming requests often have associated service requirements that are not fully understood until they are examined by the knowledge workers.

*Solution:* Apply '*Triage*' rules to characterize the incoming request and then assign the most cost effective resources through routing. The *request classification* characterizes a request as belonging to a particular *request type* and assigns associated initial resources. A common initial classification is as ***routine** (i.e. assigned to level I) **or non-routine routing** (assigned to level II).*

The initial classification can be enhanced and a *routing* (that is additional classifications that are based on needed transactions) is developed for the request. Routing requests to transactions is discussed in the next pattern. Classification and routing can occur throughout. Also there is central log of request and its current status.

*Consequence:* In accordance with best-practices (e.g. ITIL, Lean) triage provides request log, status and separates the routine requests from the non-routine ones (that require more transactions) so that the most cost effective and appropriate processing skills and services can apply.

### RED Pattern: Milestones for Performance Measurement and Managing Customer-Provider Service Level (SL) Perspectives

*Problem*: Service delivery involves dynamic discovery making performance measurement difficult, the definition of SLA goals difficult, and operational improvement difficult to manage.

*Analysis:* Within a services environment processes vary and resource usage varies because of unavoidable requirements discovery. This is a reason why SLAs are difficult to define and deliver on. Static process models *do not* serve well the purpose of capturing the variations that occur in these environments. We need a model to focus on service performance milestones achieved for each group of stakeholders (customer, business operations, execution and IT) enabled by varying underlying process. The milestones provide the invariant structure as they focus on the results of applying different processes, rather than the process specifics.

*Solution:* When a *custome*r's request of a particular type is assigned to a *provider* organization, there is a specific *customer-provider transaction* that is executed to produce the associated service deliverable**.** This is a ***RED*** transaction (see Figure 3) because of its three milestones.

| Milestone | SL Perspectives: measures |
|---|---|
| **R**equirements: Successful deliverable and transaction closure are mutually understood by both the customer and provider and eWorkcenter resources are available for execution. | *eWorkcenter OLA*[iv]:<br>° the needed entity usage times (planned/ actual),<br>° wait time for eWorkcenter role assignments, travel or other non-value added time |
| **E**xecution: Customer value-add is completed with provider-supplied services | *Customer SLA*[v]:<br>° span time from start to finish of RED<br>° 'E' span time (also TAKT time[vi]),<br>° satisfaction (high, medium, |

[1]'eWorld' includes manifestations of ***service providing entities*** such as humans, resources, assets, organizations, processes etc. Often these are both in the physical world and within IT software systems. In addition there are active agents such as automated response systems.

| | |
|---|---|
| | low) |
| **D**elivery: Service to customer is completed and accepted, consequent cost account closure occurred. | *Business:*<br>° Cost accounting,<br>*Improvement:*<br>° Defects and corrective actions. |

**Figure 2: RED transaction milestones, perspectives and measurements.**

The RED transaction focuses on the service usage and performance achieved for service level management, but not on the details (e.g. how a process works, the resource details, or how data is represented) of the underlying shared infrastructure. In general examples of RED transactions include incident-to-resolution, order-to-cash, procure-to-pay etc.

Often a deliverable may require sub-deliverables. In this case a primary transaction is often achieved by sub-transactions that can be initiated by an internal customer on behalf of an external one. To address new requirements discovered during execution, requests can initiate sub-transactions and unanticipated sub-deliverables can be completed. In the case example, this includes needed cabling, additional PCs, etc.

*Consequence:* The dual customer-provider conceptualization of a RED transaction provides the structural points of measurement for detecting qualitative and quantitative performance *contributions* (△). Figure 2 illustrates how the transaction's milestones provide infrastructure use △, customer value △, business value △, and improvement △, from multiple stakeholder perspectives, simultaneously. Thus this is related to the balanced scorecard [8].

## eWorkcenter Pattern: Composition of services to meet operating level requirements for RED transaction execution and service level performance.

*Problem:* When infrastructure entities are shared, multiple services must often cooperate and operate to meet the performance requirements of REDs and their SLAs

*Analysis:* High-cost shared resources are not usually dedicated within transactions. Available capacity must be shared, and utilized as fully as possible, across all transactions that need those services. Also, multiple services need to be available to ensure that together they can complete the transaction and meet the RED SLA. Finally, the needed services for a transaction are known but actual time of use is event-based and not predictable. To address these challenges, shared entities form an infrastructure used for processing many different requests and types. This requires us to measure the SLA performance for *each request type* and relate it to OLAs that precisely guide infrastructure performance, improvement, investment and chargeback to ensure satisfaction from all the customer and provider perspectives.

*Solution:* The virtual eWorkcenter – a composition or collection of desired ***roles*** that provide transaction services – is introduced here and is quite similar to the physical work center. Requests are routed for a particular transaction to be applied, using the services of an eWorkcenter.

First, the eWorkcenter is more general than a physical work center as it references *mixed mode* services applied to produce the deliverable. In the case example, this includes all the services to be completed for a batch of installed PCs.

Next, the roles of an eWorkcenter have associated OLAs as determined by the associated RED. The roles are filled by entities in the underlying infrastructure when needed for a RED transaction. That is, when a request is queued by triage for this particular transaction, the roles needed by the request are dynamically assigned based on the *availability* of the underlying entities. (Note, in actuality, the roles can be bound - assigned entities to act as resources - either statically or dynamically.)

Finally, the assigned entities provide the transformation services that will result in the eventual service (deliverable) of the eWorkcenter as a whole. Thus together, the entities will have to meet the operating requirements of the roles.

- A request starts a customer – provider 'RED' transaction that uses services that are usually mixed mode.
- The services needed to complete the transaction for the particular request type are grouped logically into an eWorkcenter.
- The mixed mode services are provided by software or by humans within the providers' infrastructure.
- When services are applied within a transaction, a deliverable results and metrics are captured.

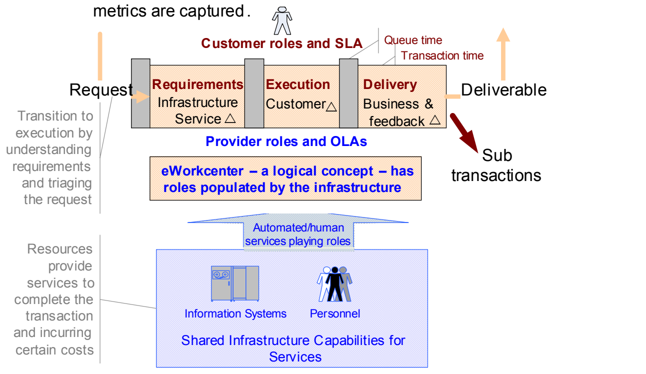

Customer roles and SLA
Queue time
Transaction time
Request
Requirements
Infrastructure Service △
Execution
Customer △
Delivery
Business & feedback △
Deliverable
Provider roles and OLAs
Sub transactions
Transition to execution by understanding requirements and triaging the request
eWorkcenter – a logical concept – has roles populated by the infrastructure
Automated/human services playing roles
Resources provide services to complete the transaction and incurring certain costs
Information Systems
Personnel
Shared Infrastructure Capabilities for Services

**Figure 3: Relating a Request, RED transaction, eWorkcenter Roles and Infrastructure services. The Requests are routed to the RED transactions that represent the customer's SLA perspective. The RED eWorkcenter uses entities from the infrastructure that fills needed roles. These roles provided the**

**services according to OLAs that produce the deliverable at the eWorkcenter.**

eWorkcenters are thus groupings of required roles to be played by *shared (or infrastructure)* entities *as* resources that provide services. Since an entity can fill roles in many different eWorkcenters, available time is more fully utilized. This sharing of entities across eWorkcenters is illustrated in **Error! Reference source not found.**. Together the eWorkcenter and role concepts support *a virtual organization*.

*Consequences:* The many-to-many associations between requests, REDs, services used and entities providing services must be traceable and measurable to define and ensure SLA performance.

The RED-eWorkcenter conceptualization provides crucial traceability between the RED customer facing performance or SLA, eWorkcenter OLA performance, and shared infrastructure-specific OLA performance.

*Traceability* is defined as the cause-and-effect along the whole chain of many-to-many associations: *Request type ⇔RED ⇔eWorkcenter ⇔Infrastructure use.*

By aggregating the RED transaction metrics (Figure 2) we now have traceability along the following *for each request type*:

- *Customer perspective*: Satisfaction and execution cause for dissatisfaction
- *Business perspective:* Investment (cause) and increased business value (effect)
- *Operations and execution perspective*:
    - RED SLA: Average response time, satisfaction, queue times
    - eWorkcenter OLA: Throughput, wait time (i.e. non-value added time spent in specific queues)
- *Shared infrastructure use perspective*:
    - *Entity OLA:* resource times used, defects

*Self-managed entities*: Finally, an entity plays roles in different eWorkcenters based on its *capabilities,* thus achieving maximal use and flexibility. But, then, each entity needs to satisfy the OLAs requirements of each eWorkcenter they participate in. For example, a human resource can be an installer in one eWorkcenter or an assembler in another. But then the resource needs to have both skills. More generally, for each of the entities shared across eWorkcenters $\sum_{eWorkcenter} used\_capacity$ provides the total entity service time used by eWorkcenters. This allows us to identify how much total capacity is needed. In the same manner we can also $\sum_{eWorkcenter} OLA$ to get the OLA that needs to be supported by each entity.

## Lean-in-the-large ACE Pattern: Deriving the routing policy for request execution to ensure Lean use of resources.

*Problem:* With many types of incoming requests that need to be executed and an infrastructure with different entities, costs and availabilities, what is the best way to assign resources?

*Analysis:* Different request types and executions require different processing and infrastructure resources. In this context, we need to define 1) the most resource efficient policy for resource assignments across request types and 2) establish and manage the performance benchmarks (with SLAs and supporting OLAs) for each request type.

*Solution*: Develop an ACE structure by identifying and assembling RED transactions needed for each request type. A *primary* RED transaction is conceptualized so that it completes a service or part of a service whose value is *directly perceptible by the external customer* (or an internal customer who acts on behalf of the external one). *Secondary* transactions are used by the primary one, but only as needed for non-routine requests. Also, there are other associations between transactions – for example the Inventory Shop may be an independent on-going transaction that maintains a buffer of PCs. All this leads to the structure of ACE transactions in Figure 6.

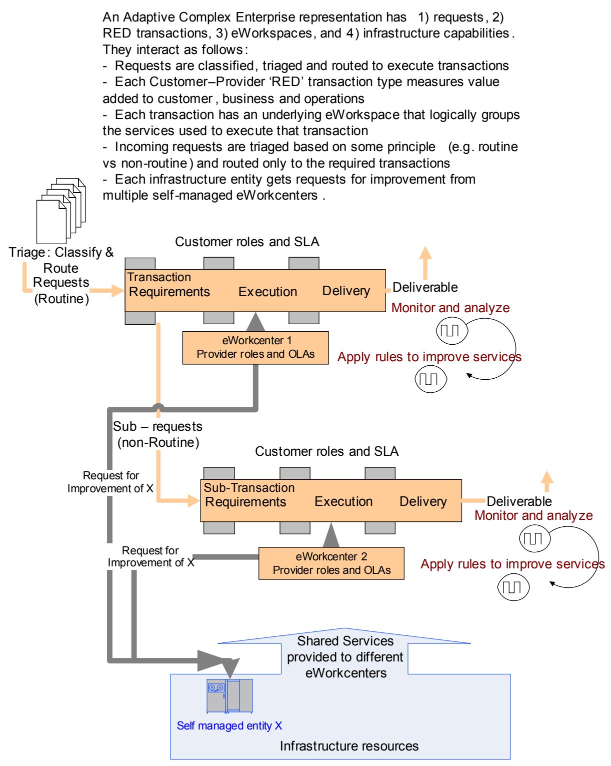

An Adaptive Complex Enterprise representation has 1) requests, 2) RED transactions, 3) eWorkspaces, and 4) infrastructure capabilities. They interact as follows:
- Requests are classified, triaged and routed to execute transactions
- Each Customer–Provider 'RED' transaction type measures value added to customer, business and operations
- Each transaction has an underlying eWorkspace that logically groups the services used to execute that transaction
- Incoming requests are triaged based on some principle (e.g. routine vs non-routine) and routed only to the required transactions
- Each infrastructure entity gets requests for improvement from multiple self-managed eWorkcenters.

**Figure 4: ACE structure illustrating interactions between incoming routine/non routine requests and sub-requests, RED transactions/SLAs, eWorkcenters/OLAs and the shared infrastructure service OLAs.**

We next *associate the nodes - the transactions - into an ACE structure* to implement the key Lean flow principle of "pull instead of push". (This is analogous to organizing the work centers in the factory floor.) To do this we must ensure that only those transactions required for each type of customer request are applied. The implementation of this is to *route* a request to only those transactions or sub-transactions (and their eWorkcenters) as required to deliver on the request.

*Consequence:* Just as the work centers on the factory floor are organized for the efficient application of transformations to the raw materials, the eWorkcenters can be organized into a Lean system of transactions and deployed as discussed next.

The RED-eWorkcenter conceptualizations can be implemented with existing systems and with training and thus serving as a useful tool for quick re-engineering, performance measurement and analysis. However, some differences between the physical and electronic world have to be understood when we apply the patterns to set up a baseline structure for the specific enterprise. As an example, consider that while physical routing and travel takes time, this is instantaneous within software systems. Electronic workflows are already present in existing CRM software and do not have the overhead of travel time. Queue time, however, is there and indicative of waste in both worlds! This can be measured for requests queued within CRM. As noted before, according to ITIL the CSC is a single point of entry for all requests. From here requests get triaged to level I or Level II resources. The virtual eWorkcenter generalizes this concept across functions, systems and organizations through its levels, based on the needs of the request.

## 5. PC Install ACE Model & Analysis

In the as-is PC install process, complexity resulted not only due to the variation in the types and numbers of PCs installed but also due other reasons such as the missing information on the site (e.g. adequate connections, cabling etc.), need for approvals, etc. The requests also came in through different ways and were not always ticketed at the Customer Service Center. Finally, multiple *RED transactions,* or hand-offs between organizations to produce sub-deliverables, are needed to complete the primary service.

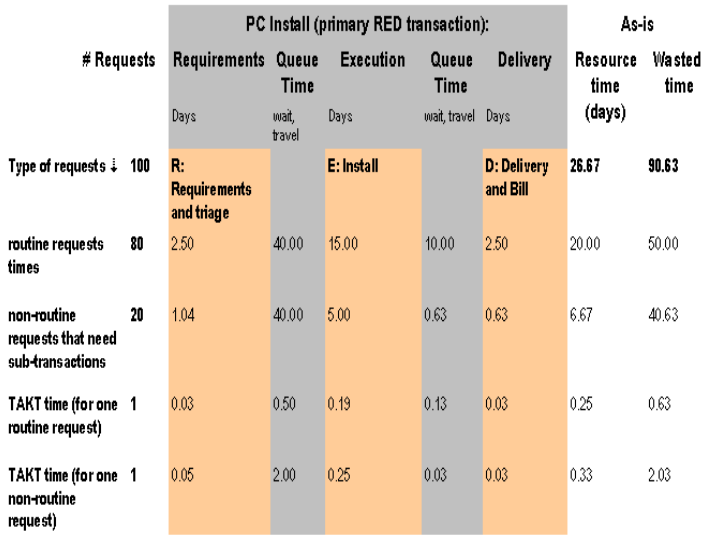
### As-Is Value Stream Mapping & Baseline

To identify non-value-add effort (e.g. queue time), we began with a time and motion study to provide a deeper understanding of where time was wasted. We found that over a hundred requests are handled each month and most are queued for a considerable amount of time. We identified the different types of requests and for each type of request the value-add and non-value-add times. The resulting baseline was as in Figure 5.

### ACE Framework for To-Be Lean Flow

We next show how we applied the above patterns for improvement in the following six deployment steps. The result of these deployment steps is a ***logical ACE architecture that is a planning, execution, monitoring and improvement model*** of an adaptive complex organization.

1. **Define Triage Rules:** The main request types and initial classifications are typically in the organizations service catalog, but the detailed to-be routings had to be created. Identify the different request types and attributes that classify according to the transaction processing needs - as routine/non routine for triage (for initial routing at the CSC). Figure 7 illustrates the enhanced classification examples for PC install in orange and the related routings to transactions in yellow.



| | | PC Install (primary RED transaction): | | | | | As-is | |
|---|---|---|---|---|---|---|---|---|
| | # Requests | Requirements | Queue Time | Execution | Queue Time | Delivery | Resource time (days) | Wasted time |
| | | Days | wait, travel | Days | wait, travel | Days | | |
| Type of requests ↓ | 100 | R: Requirements and triage | | E: Install | | D: Delivery and Bill | 26.67 | 90.63 |
| routine requests times | 80 | 2.50 | 40.00 | 15.00 | 10.00 | 2.50 | 20.00 | 50.00 |
| non-routine requests that need sub-transactions | 20 | 1.04 | 40.00 | 5.00 | 0.63 | 0.63 | 6.67 | 40.63 |
| TAKT time (for one routine request) | 1 | 0.03 | 0.50 | 0.19 | 0.13 | 0.03 | 0.25 | 0.63 |
| TAKT time (for one non-routine request) | 1 | 0.05 | 2.00 | 0.25 | 0.03 | 0.03 | 0.33 | 2.03 |

**Figure 5: Primary PC Install Transaction (RED columns), Queue time (grey columns), as-is TAKT time and totals for routine and non-routine requests.**

2. **Assemble the ACE structure of RED Transactions for Lean Routing**: Next identify and organize the transactions (and sub-transactions) into a structure to meet the processing needs from the customer's value-add perspective. Note as in Figure 6, the PC install main transaction uses sub transactions only if needed for non-routine requests. This is indicated by different color *associations*.

   In this case, there was only one primary transaction leading to the overall ACE structure and juxtaposition of the RED transactions in Figure 6. This transaction structure was created in one meeting illustrating an interesting side-effect of the ACE method that detailed process flows do not have to be created for re-engineering. Transactions and sub transactions are known to all teams and can be easily identified.

   The ACE structure of transactions is augmented with request percentages that flow to each transaction and TAKT times. This provides us with a baseline for identifying improvements. This is accomplished by starting with the as-is value and non-value added times of the main transaction. Starting with the Figure 5 times for the main transaction to deliver value as seen by the customer, we *apportion* the TAKT times for entire sub transactions.

   Figure 5 - the primary (customer facing) PC install transaction – illustrates the as-is metrics for the RED steps (in this case *Requirements, Execution of Install, Delivery and bill*). The information includes the as-is TAKT times for each request type, the resources used, and wait times. Missing as-is TAKT times are supplied through observations. For example we found out how long a survey takes.

   Next, as shown in Figure 7, use the ACE structure to create the underlying routings to these transactions based on the classification of the requirements for each request type. These routings to specific transactions are designed to use only the resources as determined by the routing. The routing can be dynamic – for example the Cabling transaction with a supplier is not needed unless the Survey initiates a sub-request.

   Finally, to deploy ACE into execution, we created routings (role assignments that can occur within the CRM). For a ***routine*** request the routing is simple (only the main PC Install transaction is required as illustrated in Figure 7). For a ***non-routine*** request additional sub-transactions are required to engage the resources of additional eWorkcenters as needed. For example, a non-routine request for "non-standard PC configuration" is routed to the "Assembly and Image" sub-transaction because the non-standard requirement is not met by the inventory of standard pre-imaged PCs. CRM reporting queries have to be implemented to get the metrics for each transaction and sub-transactions.

3. **Define RED transaction SLAs and Priorities:** For each transaction associate define customer/ business/ infrastructure metrics and the as-is SLA (Figure 5). Note also that a request is queued at a transaction till it is ready to execute using entities. This includes travel time. This time is included in the as-is SLA.

   The as-is and to-be SLA performance of each RED transaction is in the context of the ACE transaction structure. Specifically, we identified the numbers of routine and non-routine requests per month as in Figure 5. We can then see that 80% of the routine requests executed the first PC install transaction and nothing else. For the remaining 20% non-routine we next identified what fraction needed specific sub transactions and the related time metrics.

   Using this ACE structure we can now identify the precise potential to-be SLA for a transaction. For the *routine requests* the as-is SLA was 17 days but the potential was 1 day by eliminating the non-value add queue time! However this potential could *not be reached for the non-routine* requests because of the additional sub transactions needed! Requests classified as non-routine will actually need additional sub-transactions and consequently additional time. Thus, it is important that any SLA identify the request types for which that SLA applies.

Finally, define the improvement opportunity and priority of a RED based on its role in the ACE structure. Higher priorities are given to:

- o customer perceptible value add (for example reducing inventory is not immediately perceptible to the customer),
- o high volume of routed requests (for example focus on the 80% routine requests first),
- o lower throughput rates (poor approval times),
- o high queue times, also,
- o sub-transactions of those that meet the above criteria and have similar characteristics.

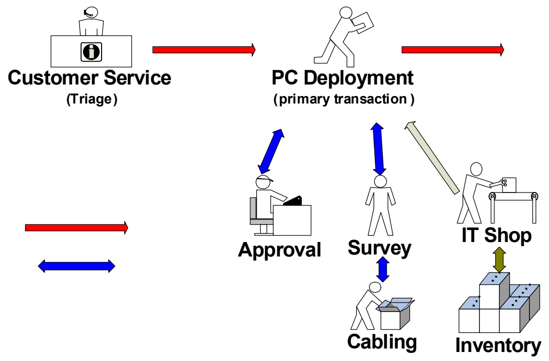
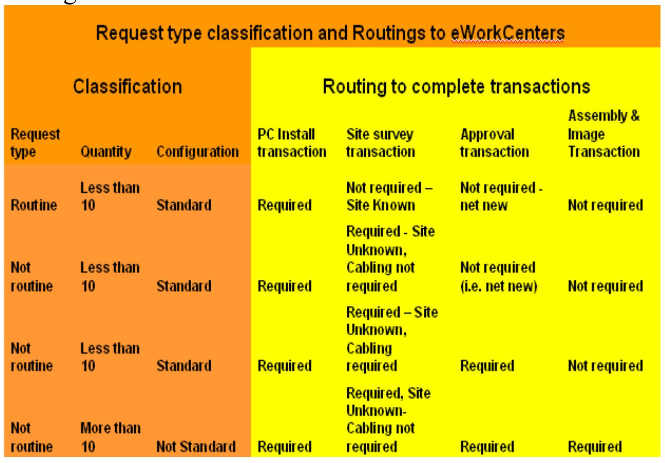


**Figure 6: To-Be PC Install RED transactions - eWorkcenters along with routine (red), non-routine (blue) routings and other associations. Note that routine requests require only the primary transaction (i.e. PC Install) and therefore this path is taken 80% of the time according to Figure 5. Thus, this high transaction volume request has highest priority for improvement.**

4. **Define eWorkcenters & Role OLAs and Priorities**: Example services from the eWorkcenter for the PC Install transaction example (of Figure 5) are in Figure 8. The columns identify the enabling IT services, organization (org) services and the migration services for the to-be. This collection takes a request or sub-request and produces a deliverable (installed batch of PC's) in the customers' environment and completes the billing. These roles have measures (OLAs and any desired △s for performance enhancement). That is, services improvements are locally identified with business transaction and managed for improvement.

   The priority of the eWorkcenter is derived from the transaction that it supports. In addition, for each role used, define the OLA desired for that eWorkcenter. And finally, the priority of the role based on its importance in the eWorkcenter. In other words, if a minor service does not really affect the OLA, it is given a lower priority. An example is space/location for a workstation may not be important but the location of the inventory holding is critical to quick service.

5. **Shared Infrastructure Entity OLAs**: The final step is to consolidate the entity perspective of the services provided. The relationship between an entity and its eWorkcenters is shown in Figure 4 - a server entity in the infrastructure might provide capacity to multiple business transactions. That is a server may play the role of a data base host for one eWorkcenter and the role of the email server for another eWorkcenter. We need to identify all the shared entity services and the roles they play in each eWorkcenter along with the current OLAs. This is accomplished by consolidating all the △OLAs against its roles.

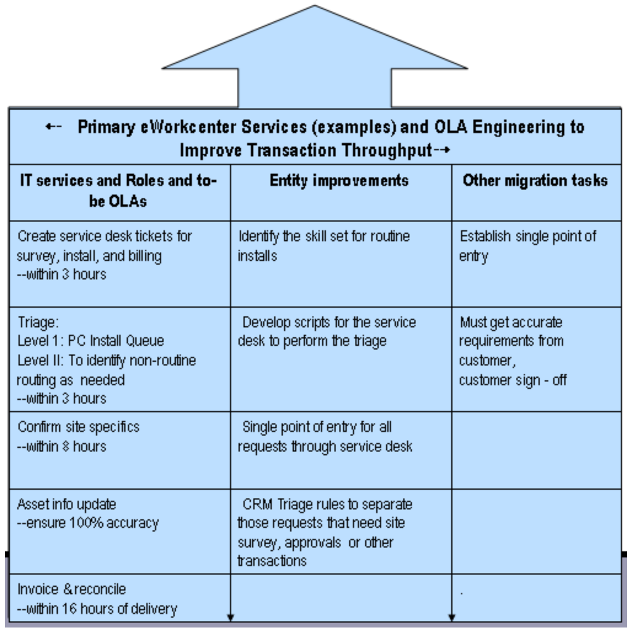


| Request type classification and Routings to eWorkCenters | | | | | | |
|---|---|---|---|---|---|---|
| Classification | | | Routing to complete transactions | | | |
| Request type | Quantity | Configuration | PC Install transaction | Site survey transaction | Approval transaction | Assembly & Image Transaction |
| Routine | Less than 10 | Standard | Required | Not required – Site Known | Not required - net new | Not required |
| Not routine | Less than 10 | Standard | Required | Required - Site Unknown, Cabling not required | Not required (i.e. net new) | Not required |
| Not routine | Less than 10 | Standard | Required | Required – Site Unknown, Cabling required | Required | Not required |
| Not routine | More than 10 | Not Standard | Required | Required, Site Unknown- Cabling not required | Required | Required |

**Figure 7: Routine and non-routine request types with classification (red) and routing (yellow) attributes. Note that the routing indicates whether a specific transaction is required or not required.**

*Self-managed entities*: Consolidate the incremental OLAs against each entity based on services rendered to play roles in each of the different eWorkcenters. Consolidation will often result in different levels and priorities derived from different eWorkcenters. These have to be reconciled by the individual entity as we will show later.

6. **ACE Adaptation:** Enter the cycle of continuous incremental performance improvement of the deployed ACE structure.



| ←- Primary eWorkcenter Services (examples) and OLA Engineering to Improve Transaction Throughput-→ | | |
|---|---|---|
| **IT services and Roles and to-be OLAs** | **Entity improvements** | **Other migration tasks** |
| Create service desk tickets for survey, install, and billing --within 3 hours | Identify the skill set for routine installs | Establish single point of entry |
| Triage: Level 1: PC Install Queue Level II: To identify non-routine routing as needed --within 3 hours | Develop scripts for the service desk to perform the triage | Must get accurate requirements from customer, customer sign - off |
| Confirm site specifics --within 8 hours | Single point of entry for all requests through service desk | |
| Asset info update --ensure 100% accuracy | CRM Triage rules to separate those requests that need site survey, approvals or other transactions | |
| Invoice &reconcile --within 16 hours of delivery | | . |

**Figure 8: Example eWorkcenter enabling the primary PC Install transaction with Organization and IT services. The underlying entity improvements and migration tasks to go from as-is to to-be are also indicated.**

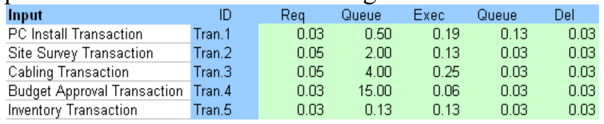
In the next section we show how the deployed ACE structure becomes the basis for adaptation.

## 6. ACE Adaptation

We prefer to use the term ‘adaptation’ rather than ‘continuous improvement’ to connote the incremental improvement that is locally defined and self-managed but is directed by overall global Lean and externally-driven context. We first show how the ACE structure is used for monitoring and providing quantitative input to decision-making leading to various ways of adaptation.

### Monitoring

ACE provides the structure for monitoring and analysis. The monitoring was not ‘real-time’ and does not have to be automated to be useful (observed values and queue metrics from the CRM etc. can be entered manually). The specific numbers for each transaction are in **Error! Reference source not found.**. The transaction metrics ( % routine/non-routine requests routed to the transaction, queue times, TAKT and non-value added times, resources used and transaction throughput) are associated with each node and updated as changes are implemented. Based on this and the ACE structure we have ACE monitoring ‘dashboard’ spreadsheet-based illustrated in Figure 10.



| Input | ID | Req | Queue | Exec | Queue | Del |
|---|---|---|---|---|---|---|
| PC Install Transaction | Tran.1 | 0.03 | 0.50 | 0.19 | 0.13 | 0.03 |
| Site Survey Transaction | Tran.2 | 0.05 | 2.00 | 0.13 | 0.03 | 0.03 |
| Cabling Transaction | Tran.3 | 0.05 | 4.00 | 0.25 | 0.03 | 0.03 |
| Budget Approval Transaction | Tran.4 | 0.03 | 15.00 | 0.06 | 0.03 | 0.03 |
| Inventory Transaction | Tran.5 | 0.03 | 0.13 | 0.13 | 0.03 | 0.03 |

**Figure 9: ACE transactions and metrics data.**

**The spreadsheet model illustrates a simulation - the metrics related across transaction associations of the ACE structure (shown as arrows), thus computing the traceability results shown in**

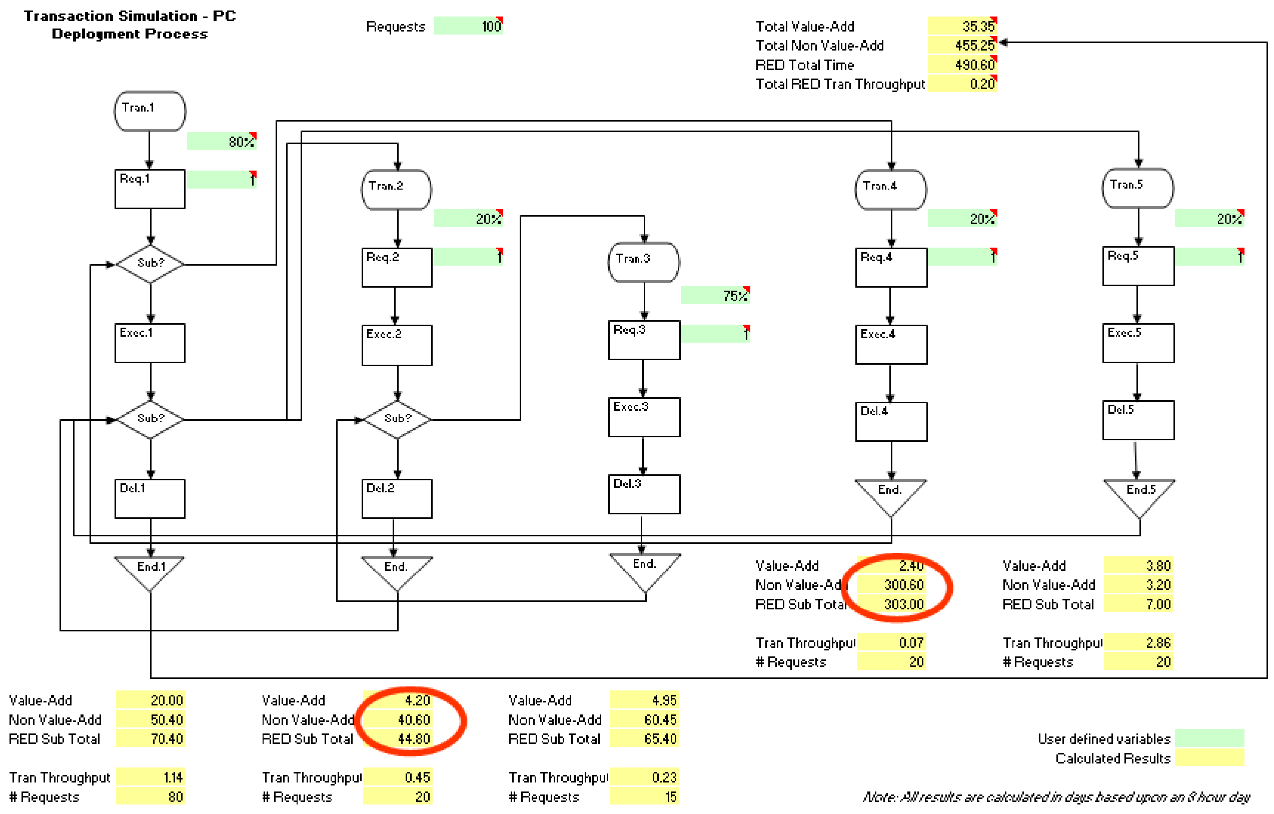
Figure 10. This lead us to some interesting observations:

- The greatest opportunity for improvement is to eliminate the wasted time due to approvals. This is 303 days.
- The opportunity through accurate site information can be addressed though a technology such as a CMDB. However, the potential improvement is only 44 days. Thus a CMDB has to be justified across many transactions.

These quantitative observations, together with analysis rules given next form the basis for performance improvement and adaptation

### Analysis Rules

Next we explore the rules that identify and typically govern the greatest opportunities for transaction improvement within the context of an enterprise. The main question we address is - how do we identify the transactions and improvements which have the maximum impact to business? As we shall see, one of the fundamental advantages of the traceability introduced above is the ability to answer this question by relating Business to IT performance by considering all the stakeholder perspectives, as shown next.



| | |
|---|---|
| Requests | 100 |

| | |
|---|---|
| Total Value-Add | 35.35 |
| Total Non Value-Add | 455.25 |
| RED Total Time | 490.60 |
| Total RED Tran Throughput | 0.20 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Value-Add | 20.00 | Value-Add | 4.20 | Value-Add | 4.95 | Value-Add | 2.40 | Value-Add | 3.80 |
| Non Value-Add | 50.40 | Non Value-Add | 40.60 | Non Value-Add | 60.45 | Non Value-Ad | 300.60 | Non Value-Add | 3.20 |
| RED Sub Total | 70.40 | RED Sub Total | 44.80 | RED Sub Total | 65.40 | RED Sub Total | 303.00 | RED Sub Total | 7.00 |
| Tran Throughput | 1.14 | Tran Throughput | 0.45 | Tran Throughput | 0.23 | Tran Throughput | 0.07 | Tran Throughput | 2.86 |
| # Requests | 80 | # Requests | 20 | # Requests | 15 | # Requests | 20 | # Requests | 20 |

| | |
|---|---|
| User defined variables | 🟩 |
| Calculated Results | 🟨 |

**Figure 10: Transactions metrics in Figure 9 is propagated along the ACE spreadsheet structure to obtain the traceability results.**

## Analysis Rules for Prioritization and Improvement from Business-IT Perspectives

### Customer perspective: relating the external environment to request types and RED transaction priorities

*Rule:* Prioritize those primary RED transactions across the business that have

- high competitive value to the business, and
- high request volume, and
- lower throughput, and
- high queue times,
- contributions to transactions that have high priority

The resulting priorities set the context for primary transaction execution improvement.

*Case examples:* For example, if 80% of the routine requests are processed by the PC Install transaction its performance has higher priority than the remaining transactions that are the non-routine 20%. On the other hand, within the same medical center, the 80% of non-routine requests (i.e. patients triaged with emergency conditions) will be of higher business value and priority. Even external pressures (such as competitive vendor pricing of $50 per PC installed) might set priorities and performance targets. For example, if the internal costing is above the vendor pricing, a strategic question to consider is why this is so and address the important organizational reasons for it.

### Business perspective relating transaction priorities to investments and constraints

*Rule:* Compare the cost of OLA improvement to the potential value due to improvement of transaction SLA. If needed justify the improvement to the underlying entities by looking at the impact to lower priority transactions.

*Case examples:* We found an example of the impact of an IT service that was less than the impact due to an organizational service improvement. By eliminating the approvals sub-transaction, **Error! Reference source not found.**we shaved off more wasted time than by implementing a CMDB that provides a service that lets the PC Install know whether the site is correctly wired or not.

### Operational innovation and SLA perspective: evaluating changes in the ACE structure

*Rule*: For the prioritized primary RED transactions (and the ACE structure), what sub-transactions can be eliminated, shared as a service, out sourced, in sourced, and/or made more efficient through IT automation?

*Case example*: For the PC Install transaction as-is TAKT for routine requests is about 17 days. The to-be SLA is determined by identifying how much of the non-value added wait can be eliminated and how much is the value-added TAKT. Wasted time can be eliminated by:

- Eliminating one or more sub transactions – in this case approval for most cases and keeping a batch of pre-imaged PCs (see Figure 6).
- Adding more resources to reduce wait time
- Increase training of entities so they can do multiple tasks.
- Eliminating the need for a survey by investing in a site management system.
- Standardizing requests – in this case eliminating special configurations and thus outsourcing to a vendor.
- Transaction interoperability - identify technology to eliminate one or more services, or combine services. For example, a request entered at the website could update the CRM automatically and generate a sub-request to the PC vendor.
- Data sharing between transactions - if a service is performed manually or if information that is created needs to be retained for later use mark it as a candidate for IT automation or interoperability. For example, a CMDB service was identified to provide up-to-date information on the site and status of communications.

Thus in this case, only by eliminating all the wasted time using the above methods we can see from Figure 5 that TAKT can be reduced to a to-be of 1 day. However, note that this can only happen for those requests that do not need any sub transactions.

Also, importantly, for each of the methods above the investment needed for OLA improvement is identified. It is interesting for note from the spreadsheet simulation that the maximum reduction of wasted time came from eliminating the approvals step (which incidentally was also lowest investment).

**eWorkcenter OLA and Infrastructure perspectives:**
*Rule:* Prioritize OLA requirements from multiple eWorkcenters: The △OLAs from different eWorkcenters are ordered in importance based on the originating transaction priority. If the entity has conflicting △OLAs, the entity negotiates with the eWorkcenter manager. For example, the OLA requirement to reduce costs by having the site surveyor also implement the needed changes might not be possible as it means hiring more expensive resources that will increase the cost of the service. Whether this increased cost is acceptable can only be determined within the eWorkcenters and the overall transaction value.

If the cost for an entity to meet the △OLA is high, see if the cost can be shared across additional eWorkcenters. However, also note the eWorkcenter manager can look for alternative services and sub-transactions through in-sourcing or out-sourcing that will reduce the cost of delivering a service. For example, the PC assembly step can be subcontracted to a PC vendor.

More generally, note that the *use* of all humans and systems services need to be managed from the eWorkcenter perspective, separately from the infrastructure perspective. Stated another way, rarely is a service useful in isolation. The eWorkcenter is a collection of services needed to deliver value. Migration tasks to evolve a transaction from as-is to to-be (e.g. scripts, interoperability implementations, skills training) are all identified locally for each eWorkcenter so that the requirements are precisely directed at improvements to the transaction performance.

**Enterprise Architecture (EA) perspective:**
The EA group applies the rules for analysis to improve the performance of ACE. With precise transaction information, the ACE monitoring and metrics enables improvement that is implemented in a highly distributed way by eWorkcenter managers and entities.

## 7. Conclusions

The motivation for the case study and research herein arises from the observation that no complex system improves from an as-is to a to-be state overnight. Neither do complex systems stay in a static state, especially in the context of on-going improvements and evolving request types. Thus, any system improvement program involving significant resources and investments requires a representation framework that allows us to achieve global objectives through locally and incrementally implementable adaptations. This shared framework would considerably reduce overhead (meetings, mapping efforts etc.) needed to improve overall system performance.

Generalizing from the specific business problem we note that certain characteristics are true of most service–oriented enterprises[2]. The ACE (Adaptive complex enterprise) framework and method provides a standardized representation, monitoring and analysis of a complex enterprise. Using the PC Install case study within a medical center, we illustrate how this

[2] teams, departments, organizations, supply chains etc.

representation allows the application of Lean principles to implement locally managed improvements in an ITIL environment.

### Future Research Issues Identified

The ACE framework is a starting point and theory for *management of complex systems to high-level objectives.* Based on ACE we have identified several areas of future research. The ACE representation and Business-IT performance alignment scheme serves as a basis for analysis techniques that can take as-is metrics and predict the overall system behavior and target to-be improvements. This includes the *development and real-time monitoring environments* that are extensions to commercial products (like Tivoli© and Openperspective©) and the use of standards (like OASIS standards for Web Services). Particular research topics include:

*WS* component:* Existing specifications for Web services describe the indivisible units of interactions. It has become clear that taking the next step in the development of Web services will require the ability to compose and describe the relationships between lower-level services. For example, starting with OASIS standards linking service usage patterns to prototype new Web services needs to be studied [2].

*Practice research:* There is a need to explore different Business-IT scenarios such as capacity management, charge-back, and disaster recovery with the objective of applying policies within the framework proposed here. Management policies for transaction SLAs, eWorkcenters services use, entities OLAs, related performance metrics and rules need to be developed.

*Policy languages*: Formalize the SLA to OLA mapping to explore issues like derivation of OLA from SLA, the "satisficing" of SLA by OLA and the resolution of conflicting OLA directives on a resource [18].

*Expansion of Model-driven environments:* We propose to expand model-driven concepts to integrate development, deployment and monitoring. This would involve assessing state-of-the-art environments to develop ACE-based instrumentation and runtime support for the eWorkcenters life-cycle (not just the creation of service functions but also their non-functional monitoring) to achieve objectives like minimized escalation though self-management. The environment would be for developing and modifying policies interactively.

*Real-time monitoring interface*: Develop Open Standardized Application Monitoring Interfaces (that can be integrated with heterogeneous products) for ACE Visualization and for knowledge-based decision making that can be installed in an evolutionary way with existing technologies. This would build upon previous work on the Kansei testbed [1].

*Curriculum research*: And finally, many of the concepts related here should be further enhanced and assembled as part of an advanced curriculum in IT Services Delivery.

**Acknowledgements**:
We wish to thank many facilitators and contributors -- Pete Shelkin, Mike Ross, Jack Frost at the OSU Medical Center; Randall Glassgow at John Deere; Mark Herriott at OIT; Michael Allan, Joe Bolinger, Michael Hussey, Nate Huffman, and Paul Scudieri at The Ohio State University.

## 8. References

[1] Arora, A., Ramnath, R., Ertin, E., Sinha, P. and others (26) "ExScal: Elements of an Extreme Scale Wireless Sensor Network," 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, Hong Kong, China, 2005.
[2] Efstratiou, C. Friday, A. Davies, N. Cheverst, K Utilising the event calculus for policy driven adaptation on mobilesystems. Policies for Distributed Systems and Networks, 2002. Proceedings. Third International Workshop, Publication Date: 2002. On page(s): 13-24, ISBN: 0-7695-1611-4.
[3] E. Gamma, R. Helm, R. Johnson, J. Vlissides; "Design Patterns, Elements of Reuseable Object Oriented software", Addison Wesley Professional Computing Series. (1995).
[4] M. L. George, "Lean Six Sigma," McGraw Hill, ISBN O-07-138521 (2002).
[5] M. Hammer and J. Champy, "Reengineering the Corporation: A Manifesto for Business Revolution," Harper Business (1993).
[6] Stephan H. Haeckel, Adaptive Enterprise: Creating and Leading Sense-And-Respond Organizations, Harvard Business School Press; (July 1, 1999), # ISBN: 0875848745
[7] D. T. Jones, J. P. Womack, "Lean Thinking: Banish Waste and Create Wealth in Your Corporation," Revised and Updated, Free Press; 2nd edition, ISBN: 0743249275 (2003).
[8] R. S. Kaplan, D. P. Norton, "The Balanced Scorecard; Harvard Business School," Press, Boston Mass. (1996).
[9] J.O. Kephart and D.M Chess, "The Vision of Autonomic Computing," IEEE Computer 36, No. 1 41-50 (2003).
[10] J.O. Kephart "Research Challenges of Autonomic computing", IBM Research Report, 2005.

[11] Magnus Penker (Author), Penker (Author), Hans-Erik Eriksson (Author) "Business Modeling With UML: Business Patterns at Work" John Wiley & Sons; 1 edition (February 1, 2001), # ISBN-10: 0471295515
[12] Michael E. Porter, "Competitive Strategy: Techniques For Analyzing Industries And Competitors, " Free Press, ISBN 0684841487, June (1998).
[13] R. Ramnath and J. Ramanathan, "IT Architecture and the Case for Lean eBusiness Process Management," MKWI, (2004).
[14] R. Ramnath, "Strategic Planning and Execution for Information-Technology Enabled Sense-and-Respond in Complex Public Organizations," CACM, May 2005.
[15] Ramanathan, Ramnath "Co-engineering Business, Information Use, and Operations Systems for IT-enabled Adaptation." Book chapter in "Adaptive Technologies and Business Integration: Social, Managerial and Organizational Dimensions", Publisher IDEAS. 2006.
[16] Ramanathan. Fractal Architecture for the Adaptive Complex Enterpise," Communications of the ACM, May 2005.
[17] L. Stojanovic, J. Schneider, A. Maedche' S. Libischer, R. Studer, Th. Lumpp, A. Abecker, G. Breiter, J. Dinger. "The Role of Ontologies in autonomic Computing Systems," 2004 IBM Systems Journal, Vol 43, No 3, 2004
[18] Zhile Zou, Zhenhua Duan, Building Business Processes or Assembling Service Components: Reuse Services with BPEL4WS and SCA European Conference on Web Services (ECOWS'06) pp. 138-147

---

[i] Configuration Management Data Base
[ii] ITIL: http://www.itil.co.uk/
[iii] TOGAF: http://www.opengroup.org/
[iv]Operating level agreements – we use the term agreement to mean both formal and informal agreements.
[v] Service level agreement - we use the term agreement to mean both formal and informal agreements.
[vi] **TAKT time** can be defined as the maximum time allowed for producing a product to meet customer perceived value. It is derived from the German word taktzeit which translates to clock cycle (Wikipedia).