SSA-2020: Vision and Strategy

By Ephraim Feig, Ph.D.
Former Associate CIO for Vision and Strategy, SSA

Abstract

This paper does not address the issue of Social Security solvency. It deals with the operational aspects of the agency that administers Social Security services. For more than a decade now, the Social Security Administration's (SSA's) IT spending has ballooned while overall cost-efficiency has declined and its ability to execute its services has deteriorated. SSA's current approach to modernization will not yield the services that Americans expect and deserve, and will result in serious cost increases. There is a better way. This paper shows that we can deliver superior Social Security services at reduced costs.

1. Introduction

I joined the Social Security Administration (SSA) in March 22, 2010, as Associate Chief Information Officer of Vision and Strategy, a political appointee of the Obama Administration. Early on, I observed that none of SSA's long-range plans discussed issues of budgetary constraints. I found it peculiar, because I have never imagined planning in a budgetary vacuum. To me, this is not planning; this is wishful thinking. I found it especially troublesome because spending issues are nowadays at the forefront of most people's concerns regarding government. I quickly learned that lack of budget awareness in strategy is not just an SSA issue; it is endemic in the Federal government. Of the 126 federal agencies' High Priority Performance Goals for FY2011[1], only one relates directly to administrative costs; OPM's Teleworks goal strives to "reduce management costs"[2].

Another thing that struck me is SSA's marketing message to its customers, the US citizens, and more directly, to Congress, which represents them, and to the Executive Branch, which oversees the overall service delivery of government. Its message is that they are a hugely complex enterprise and that they deal with massive amounts of data, and yet they deliver their services very cost-effectively. Initially, I was indeed overwhelmed by the complexity and size of the enterprise. But what I was observing day to day all seemed to contradict the claims of efficiency. I decided to investigate these claims, and coming fresh from the private sector, I started looking at the money trail.

The Social Security Administration is an information enterprise, so it is not surprising that Information Technology (IT) plays a key role here. It is also not surprising that technology modernization and innovation are critical to its continued success. I asked the obvious question, how effective are SSA's

[1] http://www.whitehouse.gov/sites/default/files/omb/performance/high-priority-performance-goals.pdf

[2] Several DOD goals deal with acquisitions and contracts, but do not specifically call for reduction of overall costs.

modernization and innovation initiatives? What I learned, as I will show in Section 3, is that for more than a decade now, these initiatives have been generating negative results.

Next, I asked, what could be done about this? As I will show, continuing on the current path is not sustainable. We have to find another strategy. I propose that, instead of starting with what SSA is doing today, we look at what SSA is required to do, which is (1) to assign people numbers and gather information about them; (2) to decide whether to pay people based on the information gathered; (3) when we have to pay, to then compute how much to pay based on the information gathered; (4) to initiate timely and accurate payment transactions (by the Treasury); and (5) to communicate with people regarding current and future payments and matters that relate to these payments. That is essentially what SSA is required to do as an enterprise.

SSA does not sell products or services. It does not enter new markets or leave old ones, except as Congress dictates. It does not have to worry about competitors. In other words, SSA is not burdened by the major complex challenges of regular businesses. However, SSA does have to deal with very arcane processes that are the evolution of over 75 years of accumulated legal baggage. The rules regarding the decisions whether to pay and the amounts to pay, while indeed mind-boggling, are deterministic, and in a modern IT environment, should be handled much differently than they are today (think of TurboTax handling income tax rules). Bottom line, relative to most large businesses, SSA is not complex.

As for data size, as I will show later, the amount of data SSA needs to deliver its mandated services is not at all massive in today's IT landscape, and certainly not in the landscape we anticipate in the coming decade.

How can one explain the disparity between SSA's self-image and the realities that I have observed? It turns out that SSA's legacy systems constrain its ability to innovate its processes. As SSA has been upgrading and innovating for several decades, it got to a point (around 1999; see Figures 1 and 2) when, rather than making SSA more efficient, modernization began to have the opposite effect. This is analogous to maintaining a very old car for critical needs. The car has to function at some minimum level of reliability and we want the car to conform to modern driving and maintenance paradigms. At some point, the cost of maintaining the old car becomes prohibitive and it is time to buy a new car.

I should point out that the planned new data center will not alleviate this problem. On the contrary, it will increase SSA's IT costs and all the Agency will be doing is building a replica of the old car using modern parts.

So here we are, early on in the second decade of the second millennium. Maintaining and enhancing SSA's operations is getting more and more expensive. We are expecting a large growth in service demand because the baby-boomers are entering retirement age and as the population ages, the number of people claiming disability will grow. All this is happening right after our greatest economic downturn since the Great Depression, while unemployment is still intolerably high (this, too, leads to more disability claims), and SSA's funding is likely to face enormous pressures for quite a long time.

But, there is a way to turn things around at SSA. This document posits that innovation is not enough to deal with SSA's current problems. What it need is entrepreneurship and creative disruption. This happens in the private sector, but there are no incentives to make this happen in government. On the contrary, inertia works hard to quash creative disruption. But we must not allow SSA to continue upgrading the old car at the detriment of the public. It is time to get a new car.

Getting a new car will be quite a challenge. We will not find such a car ready-made; we will have to build it. More dauntingly, we will have to carefully move the driver and the passengers from the old car to the new one while driving, without slowing down.

In Section 2, I will describe my vision for the Agency in 2020. This vision is totally aligned with the one published in March, 2011, by the Social Security Advisory Board[3]. I will give my assessment of what is possible if we create a modern information infrastructure. Section 3 describes current IT landscape as it relates to SSA's needs, and what we can expect in the coming decade. My IT projections do not rely on any invention, are on the conservative side, and only assume a continuation of the IT performance gains that we have been seeing for the past several decades. Section 4 describes how SSA has been modernizing for the past few decades and highlights what I have asserted above, that beginning around 1999, SSA's IT efforts have been generating negative results. Section 5 critiques SSA's current approach. Section 6 presents my proposed approach to modernization. As part of the strategy, Section 7 describes how to transition from the old to the new while continuing to deliver required services. Section 8 gives concluding remarks. Finally, an appendix (prepared by my former staff of the now defunct Office of Vision and Strategy) lists desirable service features that will come out of the new SSA, but which would not be possible if SSA continues in its current trajectory.

2. Operational Vision

In 2020, I see SSA field office customer representatives (CRs) being considerably more efficient and less pressured than today, being better supported by information technology tools that guide them through their processes. They use information interface paradigms that are familiar. Their main workflow tool is the SSA CRM (Customer Relationship Management) system. Their home page provides a glimpse of what to expect immediately (the day they are working), what is pending and what are the due dates, and personalized performance outcomes (CRs will see how they are personally contributing to SSA's strategic goals). Data entry is fast; users enter data once, and the data is there anywhere you need it, for any application. Likewise, users make a change of the data in one place, and it is used in any other future application; older data is archived, not discarded. All data and rules are tagged with time intervals, so users can re-compute results as they would have happened at any time in the historical record of a customer (this will greatly enhance the quality of SSA's notices and the process to generate them).

[3] http://www.ssab.gov/Publications/Miscellaneous/SSAB_Vision2011_FINAL.pdf

When a customer visits, the CR has a 360-degree view of the person, including what services are already availed to him; what were his most recent interactions with the Agency; what, and at what stage of the workflow, are his pending issues. If the customer wants to see retirement benefit scenarios, a convenient graphical user interface allows input of expected future earnings, if any, and then conveniently shows benefit amounts for various election months. Behind the scenes, a rules-engine based application is running, which leverages all relevant existing information, asks all and just appropriate questions, and then proceeds according to the answer it receives with further questions, until it has everything it needs to generate accurate benefit amounts. In almost all cases, the process is complete before the customer leaves the office. This rules-engine based application is reminiscent of modern tax-preparation software applications; it is easy to learn, it checks for errors and guides end-users as to how to correct them, it suggests alternatives when available, and it makes sure that applicants provide all the data needed to make determinations and compute correct payment amounts. This process will radically change how SSA educates its field workers. They will spend relatively little time formally learning the arcane rules for navigating screens and making determinations and computations and become productive much quicker. The process will alleviate the growing concern, that as SSA's older workers retire and new, young workers arrive, it is getting harder and harder to pass on its institutional knowledge. This process will also drastically reduce the workload now assigned to Processing Service Centers.

This same rules-engine based technology is used for online self-service claims applications. This will make the online claims applications simpler, enable online completion of many more applications, and eliminate the need for callbacks and other manual processing in almost all cases.

Accurate data will be available to those who need it, when they need it. There will be one single (virtual) data source. If the same data is needed for various different programs, they will all get it from this database of record. If Congress requests information that is not available in standard reports, SSA will have it for them most typically within hours (often, immediately), or in some cases, after over-night processing.

Data exchanges and analytics will become simpler to deploy and manage, and SSA will be able to run much more sophisticated algorithms. These will enable SSA to catch more anomalies and catch them faster. This will reduce improper payments. We will have more sophisticated tools to semi-automate (and in many cases, fully automate) review processes, again reducing improper payments and enhancing trust in our services.

Similarly, post-entitlement processing will be almost fully automated. This, too, will contribute to an enormous reduction of workload currently done at the Processing Service Centers.

Health IT will advance significantly. More of the health records data will be structured, and automated digital transmission of data together with decision support tools will accelerate the decision process. Determination decisions will become more consistent and integrity reviews will speed significantly. SSA

will have to rethink their reconsiderations processes in light of these new tools; it is hard to predict how they will decide on this issue when it comes up.

Application development will look nothing like it does today. SSA's 2011 Sitar (Strategic Information Technology Assessment and Review) process proposes allocating 2 years and over 170 people-years to modify a few iClaim forms. In our SSA-2020, many forms, whether for online self-service or for field office or telephone center use, are created via configuration, without any code customization at all. These can be done in hours or days, not months and years, with even greater assurances of security and conformity to governance rules. If new data-types must be introduced, then development will take somewhat longer, but still it would be orders of magnitude faster than today.

User interfaces will follow standard, accepted paradigms. The use of SMEs (subject matter experts) and usability testers will be very different from what it is today. For example, web-based self-service applications will start with existing templates and be created via configuration and perhaps some customization (if, say, new data fields have to be introduced), which, because of the new environment, will be much simpler. These could be deployed with limited exposure to study customer reaction, which may lead to some tweaking and further limited exposure. At some point, a decision can be made to completely release the application or simply withdraw it. The cost for creating the application will be so low and the development time so fast, that it will be quicker and cheaper to actually create it and test it than to internally discuss it, plan it, size it, and then decide whether to create it or not.

In my vision for SSA-2020, SSA's workforce is significantly reduced (via attrition, SSA does not have to replace all those who retire) and still performs more efficiently and under less pressure than its total workforce performs today. SSA has the technological and operational flexibility to close field offices and Processing Services Centers (political and union considerations will come into play, of course) and to hire external workers to perform mundane tasks. Total IT costs- hardware, software, contract labor costs and IT staff pay- shrink enormously and in the process, IT deployment becomes orders of magnitude more efficient. In order to understand what drives this optimistic vision, we have to consider capabilities of modern IT, as they relate to SSA's enterprise.

3. Information Technology Landscape

The purpose of this section is to demonstrate that, in today's IT environment, SSA's computational and storage requirements are not massive. This assertion becomes more dramatic when we consider where IT is going in the near future. SSA is about to embark on a $500 million investment for a new data center, and this sum is only the beginning of the project. I have heard estimates of $750 million to finish the project, and then the Agency will be further burdened with continuing massive maintenance costs. As I mentioned in the first section, this data center will be a clone of SSA's current environment, except that the equipment will all be modern. As a clone, it completely misses the reality of IT evolution, that in five years, the same amount of processing, storage and communications power will require less than one-fifth the space. Even more troubling, SSA's historical modernization initiatives have failed to

leverage the exponential growth of IT capabilities over the past several decades. After you read this section, you will have to wonder, is SSA really making good use of our hard-earned tax dollars?

To start with, let us consider the amount of data SSA actually needs to run its enterprise for at least the next decade. We consider two types of datasets- one comprises medical records that are needed for disability determinations, and the other comprises all the rest (personal information, including histories of interactions with the agency, addresses, wages and relationships, to name a few). For the latter, we need to store information on considerably fewer than 400 million people, and the typical size of a record per person is under 50KB. Even assuming an average of 250KB per person, this would translate to a total record size of 100TB of data, which today is not large. Today, one can buy (Western Digital, Hitachi, Samsung or Seagate; at Newegg.com) a 2TB desktop hard-drive for less than $80; in five years, the same amount of money will buy at least 10TB. In other words, in five years, consumer grade storage for 100TB of data will cost under $800.

Of course, enterprise grade storage is not at all the same as consumer grade storage. To the hardware costs of enterprise grade storage one must add costs of housing, connectivity, redundancy, energy and support services. Google charges $25,600 a year to store 100 TB of data[4]. Amazon S3 Cloud is indeed an enterprise grade service, and for the types of loads on core data that we expect at SSA, charges would be under $12,000 a month[5]. The model here (not necessarily how I would design things, but for argument sake; cost tradeoffs depend on where the database and application servers reside) is that 100TB is stored, and 200,000 complete personal records, averaging at most 250KB each, are transferred in and out every day). Also, overall storage service prices will not go down as fast as hardware prices. But it is safe to assume that in five years, enterprise cloud storage fees for SSA's core data requirements will be under $100,000 a year.

As for medical data, currently the average record size is about 1.5MB. SSA's very few largest records (from the VA) are 375MB. We can estimate that for the coming decade, SSA would need at most an average of 50MB (more likely, around 10MB, which translates to about 200 pages of tif images) per each of at most 100 million (more likely, around 60 million) disability claims related people. This translates to at most 5PB (more likely, around 600TB) of data. Access to this data is not required in even near real-time, which means that a medical records system for SSA can comprise of mostly low-cost storage devices with some enterprise grade high access rate components. As a baseline, we note that in five years, consumer grade storage (just the hard drives) for 600TB of data will cost under $4,800.

Next, let us consider workload support. In the field structure, SSA currently has fewer than 19,000 customer representatives, 10,000 service representatives, 4,000 field office management and supervisory staff, and 10,000 program service center employees. If all of them, simultaneously, use a Customer Relationship Management (CRM) system to manage their task assignments and maintain and access a 360-degree view of our customers, we would require the ability to handle under 54,000

[4] https://www.google.com/accounts/purchasestorage

[5] http://aws.amazon.com/s3/pricing/

concurrent CRM users. SugarCRM reports[6] that "a single high power server with eight 86x64 CPU cores, 16GB of RAM and enterprise storage, running the entire application stack (SugarCRM application, PHP, web server and database server) can support up to 400 of concurrent Sugar users. One such server, Dell's PowerEdge R715 Rack Server, loaded with SUSE Linux, currently costs about $8,000. Microsoft reported[7] already back in 2008, running 500 concurrent users of Microsoft Dynamic CRM. They use one HP Proliant DL 585 (4 cores) database server, on e HP Proliant BL 25 (2 cores) application server, and one HP Proliant DL 325 (2 cores) load generation server. Equivalent servers today cost under $15,000. This implies that in five years, when servers are at least five times more powerful, the server cost for supporting 500 concurrent CRM users will be under $3,000, which in turn implies that standard enterprise-grade rack mounted servers, costing around $325,000, will be needed to support 54,000 concurrent CRM users. Of course, the actual number of concurrent CRM users will be considerably smaller; far from all CRM users use the system simultaneously. The above analysis provides a crude cost estimate, and is intended only to demonstrate that SSA's hardware requirements for CRM are orders of magnitude less that the types of systems they deploy today.

SSA is currently processing about 30,000 claims applications a day. Let us assume that we have to process 100,000 such claims a day; assuming a 10-hour day, this translates to an average of 167 claims a minute. So let us put our peak requirement at 500 claims a minute. Let us also, pessimistically assume that each claim requires 100 invocations of a rules engine; this would translate to 50,000 rules engine calls a minute, or 833 calls per second.

A 2005 benchmark study[8] found ILog Rules Engine running on a dual-core Xeon machine with hyper-threading, 2 GB RAM, JBOS 4.0.1, Linux configuration, doing over 100 transactions per second, invoking a rule set containing 10,000 simple rules. If we allow a factor of 100 to compensate for our potentially more complex rules (again, I am being overly pessimistic), this simple 2005 configuration will handle 1 invocation of the rules engine per second, or a requirement of 833 2005-class commodity servers to meet our requirements. In five years (eleven years after 2005, the year of the study), we expect our requirements to be met with fewer than 40 enterprise-grade rack-mounted servers. Here, again, I am simplistic about the actual architectural configuration, because my point is that the IT costs for implementing a rules-engine based system to handle SSA's arcane decision processes will be orders of magnitude less than what SSA pays today (and will pay even more in the future) to handle such decisions, while providing much better services to the citizens.

As for Health IT, SSA does not need real-time or near real-time performance. SSA is not handling emergency room medical procedures. SSA can schedule medical data delivery well in advance of its use for disability determinations and reconsiderations. During the coming decade, SSA will process, annually, fewer than 5 million disability claims; this translates to an average of fewer than 20,000 claims per workday. Assuming, pessimistically, that each claim translates to 10 transfers of electronic health records and that all transfers are done within a 10-hour window during the day, we will be expecting no more than 20,000 transfers per hour, or fewer than 6 non-real-time transfers per second. In five years, this will not be a processing or bandwidth challenge, even if average file sizes are 25MB.

4. Is SSA really modernizing IT?

[6] http://media.sugarcrm.com/datasheets/SugarCRM_Sizing_Guidelines.pdf\

[7] http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=20122

[8] http://logic.stanford.edu/POEM/externalpapers/iRules/jrules_cap_wp.pdf

SSA claim to be continually modernizing its IT environment. They are certainly augmenting to it and making changes; but are they actually modernizing?

SSA is in the midst of a huge database migration initiative, involving first moving its old MADAM database to a flat file inside DB2. Once this is done, they plan to convert this flat file to a relational database schema in DB2. The current first phase of the program is now in its sixth year; SSA would like to finish the entire migration within a years. During the past decade, SSA has enabled online interactions with its constituents, and is continually upgrading its online capabilities. It has been either retiring or converting some of its mainframe assembly code, and it continually migrates various parts of its COBOL software to modern languages, most typically Java. There is no plan to eliminate very old COBOL applications that form the core of its arcane decisions and computations. A good critique of the Agency's IT modernization initiatives can be found in a 2009 report by The Computer and Communications Industry Association.[9]

The goal of modernization is to get more bang for the buck; that is, to be able to provide more and better services for relatively less money spent. So, how well is SSA doing?

First, let us look at SSA's total IT spending (including contractors and salaries) for the past two decades. Figure 1 below gives values for the past two decades. During the decade of the 1990's SSA's total IT spending was relatively constant, around $600 million (inflation adjusted 2010 dollars). However, something happened in year 2000, and during the next decade, SSA's IT spending has ballooned to nearly 2.5 times the amount that it was in 1999. Rising costs in IT are not necessarily a bad thing, if in turn, the costs of delivering services goes down or the quality of services gets better, or ideally, both happen simultaneously.

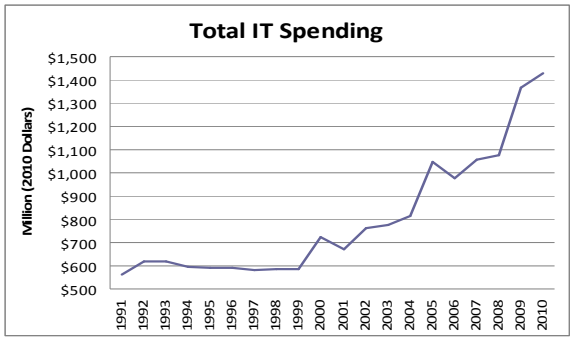

Figure 1: Two Decades of IT Spending at SSA

[9] http://www.ccianet.org/CCIA/files/ccLibraryFiles/Filename/000000000296/CCIA%20SSA%20Citizen%20Data%20Paper%20Nov%205%202009.pdf

There is no evidence that the quality of SSA's services is significantly improving because of IT investments in the past decade; there are areas where we know that customer satisfaction is actually down (customer satisfaction with our 800-number phone service dropped significantly). For a while, SSA has been reducing its disability determination backlog and speeding up the average time to decision, but this has been achieved primarily by adding staff and changing policy. Reducing the backlog is a high priority performance goal for the agency; it is the first strategic goal listed in its latest Agency Strategic Plan (ASP)[10]. Unfortunately, even here the Agency has begun to slip.

A June, 2011, TRAC (Transactional Records Access Clearinghouse, Syracuse University) report[11] tells the story. In May, 2007, Commissioner Astrue testified to Congress about a plan to eliminate the backlog of hearing requests by 2012 and to eliminate its recurrence. In March, 2011, the Commissioner again testified to Congress, saying, "We reversed many negative trends, most notably with the hearings backlog…" But the study found that "the overall number of individual claimants awaiting a hearing has not in fact gone down but up, climbing to 728,012, higher than it was when the SSA launched its expensive rehabilitation plan four and a half years ago." The Washington Post reported[12] that the Commissioner called the report "research fraud," suggesting that TRAC used a bad definition for backlog. TRAC, of course, used the same definition for the term that SSA has been using for years.

One way of trying to assess SSA's efficiency, at a very high level, is to look at the amount of dollars per "customer" that SSA spends on administering its various programs. This ratio really cannot be used as an absolute measure because we cannot realistically assign dollar values to social outcomes (people have indeed tried; I do not have to do it for the points I am making here). What we can do is look at such ratios historically and see how well SSA does year after year. We can say that SSA's investments are paying off if the cost per customer goes down some time after it makes these investments (remember, service quality has not really gone up).

We can actually estimate annual administrative costs for SSA's key service areas- OASI (Old Age and Survivors Insurance), DI (Disability Insurance), SSI and Medicare Part D. Here, let us restrict ourselves to OASDI, which comprises OASI and DI. These are paid primarily from the OASI and DI Trust Funds, respectively; sometimes, Congress allocates extra funding for these programs. Every year, SSA takes out some amounts from these funds to support these programs, and they make adjustments the following year for under-funding or over-funding from the two funds into the appropriate programs. SSA reports annually on these amounts, and historical figures can be found on their website.[13] As mentioned above, these figures do not include special funding SSA received in support of its OASDI programs, like the over $1 Billion via ARRA[14], the American Recovery and Reinvestment Act of 2009. Likewise, a historical

[10] http://www.ssa.gov/asp/SumGoalsObj.pdf
[11] http://trac.syr.edu/tracreports/ssa/253/
[12] http://www.washingtonpost.com/politics/progress-on-disability-benefit-backlog-disputed/2011/06/17/AGdS6wbH_story.html
[13] http://www.ssa.gov/OACT/STATS/admin.html
[14] http://www.ssa.gov/oig/communications/testimony_speeches/04282009testimony.htm

account of the number of OASDI beneficiaries can also be found on the SSA website[15]. We can then use these figures to estimate the annual fund allocation amount per beneficiary for these two programs. If we do that, the graph will be jagged, because of the adjustments that I have described above. Taking 4-year trailing averages (the sum of OASDI admin amounts for four consecutive years divided by the sum of beneficiaries serviced in each of these years) gives a good estimate of the actual admin cost per beneficiary for each of these two programs. The graph of this is given in Figure 2.

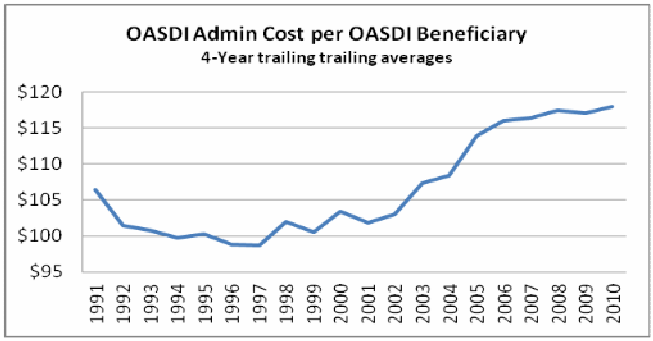

Figure 2: Estimate of OASDI Admin Cost per OASDI Beneficiary

As we can see, until 1997, the graph looks good; SSA was getting more efficient. However, from 1998 onward, efficiencies started to deteriorate. Some of this efficiency loss is due to rents, wages and security costs (especially after 9/11) going up faster than CPI. A major cause for this upward trend is that the ratio of DI beneficiaries to OASI beneficiaries has been increasing rather significantly during the past decade, and DI costs per DI beneficiary are much greater than OASI costs per OASI beneficiary. In 1999, 14.6% of all OASDI beneficiaries were DI beneficiaries; by 2010, the number was 18.9%. But are these reasons enough to explain the inefficiencies highlighted in Figure 2?

Let us first address the effects of the rise in the DI to OASI ratio by doing the following thought experiment. Suppose that at the end of 1999, SSA had decided to stop all IT modernization and just continued as it was for the next 11 years. Let us further suppose that the average OASI cost per beneficiary for these 11 years was equal to the average OASI cost per OASI beneficiary of the previous decade ($72.82, a rather low-key assumption, as though SSA has not learned to do any better than just average), and similarly that the average DI cost per DI beneficiary for these 11 years was equal to the average DI cost per DI beneficiary of the previous decade ($281.56). Extrapolating to actual number of OASDI beneficiaries that SSA serviced during the past decade, the agency would have delivered OASDI services to these same beneficiaries for close to $3 Billion less than what it had actually spent.

The argument above showed that while the rise in DI to OASI ratio contributed significantly to the performance loss highlighted in Figure 2, there still remain a near $3 Billion loss to explain. The added costs due to the fact that federal employee wages increased faster than CPI and that rents and security

[15] http://www.ssa.gov/OACT/STATS/OASDIbenies.html

costs increased significantly will contribute perhaps another $1 Billion to the deficiency (amortizing these costs among SSA's other programs, SSI and Medicare, suggests a lower sum). In other words, the typical prime suspects do not, by themselves, explain the inefficiencies of Figure 2.

The real reason for the loss of efficiency is that with its current strategy for IT modernization, SSA has gotten to the point where the more it invests in IT improvements the less efficient it becomes. In the past eleven years, SSA spent on IT a total of $4.1 Billion above the baseline (the average IT spending during the 1990's), but it has gotten a lot less than that in return.
We see this degradation of capability in so many concrete ways every single day. We see it in the length of time SSA requires for software development; for example, SSA is currently estimating at least two years and over 170 people-years to modify several I-Claims forms by adding a few fields. We see it in SSA's failover capabilities and plans; the Agency claims that its current disaster recovery time is five days and it plans to bring that down to one day sometime next year (let's check on this next year). Modern mission critical environments have failovers that enable recovery in seconds. We see this in the continually growing costs for just keeping SSA's current systems running. Figure 3, created by SSA's Office of Strategic Investments with input from its Office of Systems and Office of Budget and Finance Management, shows expected costs of "keeping the lights on" for the coming decade if SSA continues doing IT the way it is doing it now.

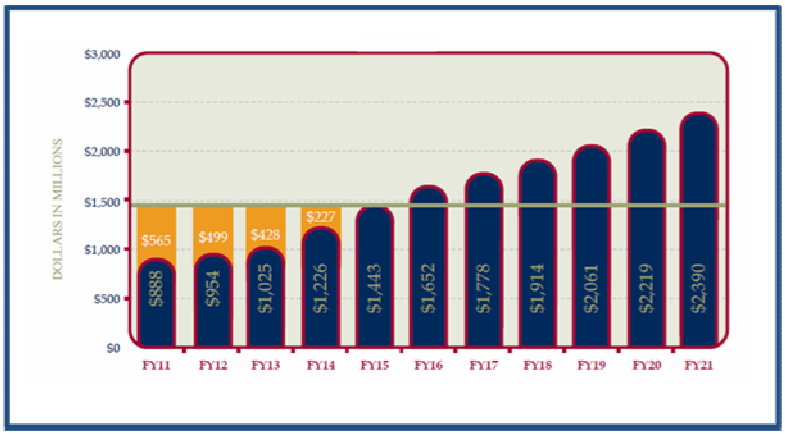

Figure 3: Expected IT Costs of "Keeping the Lights On"

The blue rounded-top columns represent expected IT costs just to keep SSA IT systems running. These costs are based on planned IT activity in support of SSA's current and expected environment (for example, added costs in maintaining its intended new computing center. The actual costs of building the computing center are not included because SSA gets special funding for this). The green line cutting across the chart delineates SSA's FY11 total IT budget, $1.453 million. The yellow segments of the columns in years FY11 to FY14 represent the amount of money that will be available for IT innovation if SSA keeps its IT spending constant throughout the decade. As the chart shows, beginning FY15, not only

does SSA not have any more money for IT innovation, it has to continually increase its IT budget just to keep things running.

Typically, organizational business requirements should drive technology. With information enterprises, since information and technology are so intertwined, we may even accept that business and operational requirements be symbiotically and concurrently driven. But in SSA's current environment, technology drives its business. SSA's Policy executives have told me that there are some policy changes they would recommend to implement in order to significantly simplify overly complex processes, but they are constrained from doing so because SSA's Systems folks tell them that to do so would require such enormous technology resources and time that the costs would be prohibitive.

Perhaps SSA can achieve its goals (listed in their ASP; see reference 5, above) by continuing along its current trajectory, but at ever-increasing costs and ever-decreasing efficiencies and longer times to deliver results. SSA would certainly not be able to achieve the vision of Section 2 in this manner. Continuing doing business as usual, under such circumstances, is not an acceptable strategy at any time, but it is more egregious during a period of enormous budget constraints and increasing workloads and customer expectations for better (Internet age) services.

5. A Critique of SSA's Current Approach to Enterprise Architecture

SSA's approach to modernization is described in its document "SSA Enterprise Architecture Transition Strategy for 2011 through 2016"; last version I have seen was dated November 17, 2010. It establishes an "IT Lifecycle Framework that provides the over-arching guidelines for defining SSA's baseline and target architectures." Then, following standard Enterprise Architecture (EA) paradigms, it "clearly links SSA's investments to the target architecture and describes SSA's plan to achieve its target 'to-be' EA within a specified timeframe, given the baseline."

The document states, "Baseline and target EAs are currently defined in SSA's EA Repository at a sufficient level of detail and completion to serve as a basis for the Transition Strategy." SSA's current target EA is indeed a very detailed collection of desired upgrades, but without a truly over-arching architecture design (except as an extension of the current design). There is no analysis to demonstrate that, supposing SSA succeeds with every step on its plan, the resulting environment will be more efficient and yield desired outcomes. In fact, as I mentioned in the previous section, an internal SSA analysis showed that costs for "keeping the lights on" would continue to increase, and at a rather dramatic pace. Yet the Agency is looking to this EA to provide IT modernization guidelines. When we consider the costs and benefits of SSA's architecture as delineated in Section 4 in light of the IT landscape described in Section 3, it becomes clear that SSA's current to-be architecture is inappropriate. The current architecture does not reflect modern IT capabilities, and certainly not the capabilities we anticipate of IT in the coming decade.

There are numerous reasons why SSA has not investigated truly modern alternative architectures. The first is that it is comfortable with what it has and scared of changing. There is good reason to be scared of big changes; historically, most have either failed or turned out to cost a lot more than originally anticipated. We will address the failure and cost overrun issues in the next two sections. Sure, there is a risk. But we have to compare this risk to the alternative- staying the course. The alternative is predictable, as can be gleaned from Figures 2 and 3: continued degradation of efficiency and increased cost pressures, which I claim is unacceptable. I do not find it acceptable for SSA to continually run to Congress and tell them that if they do not get funding increases then services will degrade. I am of the belief that we can actually deliver better services at significantly lower costs. We strive to do this every day in the private sector; we have to learn to strive to do this in government.

A second reason is that SSA's vendors, and indeed the industry at large, educate customers regarding EA in a way that does not threaten their profits. This should not be surprising; companies exist to make profits. The standard EA approach is to describe in detail the as-is architecture and then to apply various techniques of manipulating this starting point to ultimately find alternatives that are more efficient. These techniques typically find redundancies and figure out how to eliminate them. They may also find different way to granularize code modules so that re-use becomes more efficient. And they may find subsystems in the overall architecture that can be replaced with more efficient ones. But they will never find a truly new architecture, because they derive their to-be architectures from the as-is ones.

A Gartner report[16] from January, 2011, entitled "Enterprise Architecture Program Pitfalls: Don't Start with the Current State" warns against such an approach. It concludes, "Creating an inventory of the current-state EA is a low-business-value activity. The business value of EA is based on the insights into how the organization must change. Organizations that are on the path of creating a current-state inventory must re-evaluate the EA program scope, objectives, and resourcing, and change any or all of those as necessary." I agree, and this vision and strategy follows their recommendation.

Third, the IT industry is pushing the notion of application and data integration as key ingredients in to-be architectures. Integration enables the bridging of silos but, of course, at the cost of added layers of complexity. The hope is that technology will evolve fast enough so that the extra costs of the added complexity will be less than the efficiencies gained by the integration. Unfortunately, this is not always the case, especially with ancient, arcane systems. Still, such integration forms the core of SSA's modernization efforts. In our approach, integration is mostly a transitional element, used only as we are moving from the old to the new. The final architecture is designed to be self-integrating (in the SOA sense), so that external integration wrappers are not needed.

Fourth, SSA likes to view its enterprise as very large and complex. This justifies its requests for larger and larger budgets and also emboldens it to claim that it is efficient. SSA brags about new highs in daily transactions. It is not in the Agency's DNA to try to simplify its processes and to reconsider its enterprise as relatively simple. The reality is that, when it comes to transactional IT, compared to modern large enterprises, SSA is moderate.

[16] http://www.gartner.com/DisplayDocument?id=1526514

Fifth, SSA's current approach is predicated on the notion that it should leverage all of its previous IT investments. Since the Agency has already spent so much on archaic code development, it should not throw it away. The fallacy of this argument is evident to any investor. It is the same as arguing to hold on to a stock just because one has held on to it for a long time.

Sixth, and most critically, is that the notion of entrepreneurship in government is missing. There is plenty of innovation in government. But innovation is not entrepreneurship. Innovation yields clever solutions to problems; entrepreneurship yields creative destruction- the process of inventing completely new ways of thinking about businesses and then figuring out how to execute these inventions and retiring the old. Government bureaucracies, in general, discourage such thinking. Bureaucracies seek to increase their budgets; see Budget Maximizing Model[17] and Bureau Shaping Model[18]. Entrepreneurship forms the basis of my suggested approach to modernization at SSA.

6. A Strategy for Modernization

Our approach to modernization at SSA is entrepreneurial; we design and build a modern system from the ground up, and we transition to it, gradually retiring the old. Our starting point is not SSA's as-is architecture. Our starting point is what SSA has to do as an enterprise, what is required by statute and regulations. Our starting point is not constrained by what SSA has. We will design an idealized, modern architecture to deliver the services that SSA is required to deliver. We will start with a high level over-arching view of the architecture before getting down to some level of detail. This will give us the over-arching guidelines that our Enterprise Architecture demands. Only then will we consider SSA's as-is architecture, and plan a transition strategy. We will demand from the start that our to-be architecture is sustainable within a reasonable budgetary constraint, and we will design a transition strategy that is also doable within this budgetary constraint.

We will not spec out details for the overall final system; that would be a recipe for disaster. We will have a hierarchical design- top level over-arching, perhaps several cascading lower levels, and we will allow for flexibility at the leaf levels. Just like in a startup, we will isolate some piece of the system that is, by itself, a valuable and marketable subsystem, and build it from the ground up. We will use modern integration tools to temporarily co-exist with SSA's as-is architecture. Slowly, we will grow the new system and retire the old one. And just like with a startup, we will introduce creative destruction into a government bureaucracy.

It is important to understand the startup analogy in the context of a government bureaucracy. The idea is that, just as in the private sector, we start with a relatively small piece, make it successful, and then bootstrap, thereby growing the enterprise. As in the private sector, we must allow for flexibility in the

[17] http://en.wikipedia.org/wiki/Budget-maximizing_model

[18] http://en.wikipedia.org/wiki/Bureau-shaping

bootstrapping process, because we cannot anticipate all the conditions at the various stages of the process to have a growth sequence rigidly fixed. But there are several big differences between our approach here and a startup in the private sector.

First, we cannot freely bootstrap. In the private sector, once an initial business takes hold, the startup can augment it with any follow up that seems appropriate; here, we are constrained to building only what SSA is required to do by law. Second, worker and management incentives are very different in the two domains. Third, government bureaucracies are burdened by procurement and hiring rules that are more limiting than those in the private sector. And fourth, we do not start as a lean enterprise; we still have to service the public as usual. We have to carry both the startup and its competitor simultaneously, slowly growing the new and retiring the old.

I anticipate objections to this strategy. The first is that it is too risky. My response here is that the alternative, continuing on the current trajectory, is not sustainable and hence even riskier.
The second objection I anticipate is that the task is too great. Obviously, this cannot be the case; after all, SSA is servicing its customers today with a system that humans built, so we know that the task is doable. What we have to demonstrate is that it is doable within some budgetary and service-level constraints. The latter constraint is that SSA must continue to service the public at the level of quality that they have grown to expect. As for budgetary constraints, we will assume that SSA's total LAE is constant (in real dollars) for the foreseeable future. If it turns out that SSA gets more funding, we can accelerate the process; to deal with the possibility of reduced funding, we must make the process sufficiently malleable so that we can adjust accordingly while still heading towards our ultimate vision.

Let us now start building our proposed solution. We identify what we have to do, not what we are currently doing. On the highest level, we have to know our customers and we have to make determinations and computations. This suggests that we need a Customer Relationship Management (CRM) system and some rules-based engine for determinations and decisions regarding computations. The latter is reminiscent of modern tax preparation software products, which determine what forms to use, which of various optional computations to use, and then compute payment or refund amounts. Because the most popular such engine today is TurboTax, we will call our engine for determinations and computations our TurboEngine.

CRM systems have evolved over the past three decades and certain paradigms have emerged as standard for interfaces and interactions. We propose to follow these paradigms rather than invent new ones, as these have proven to be acceptable to a wide range of customers. However, CRM systems typically provide sales force automation and marketing support, and most of their standard modules are not useful to SSA (though some handle case processing and may be good starting points). On the other hand, their workflow management systems, rights access control mechanisms, and parts of their contact management and interaction history tracking capabilities can be immediately leveraged for our needs.

Similarly, the interfaces to our TurboEngine should be familiar, in the sense that they are similar to those of standard tax preparation tools. The engine will provide end-users with guidance on completing

various tasks, highlight when things are missing or suspicious (indicating probable or possible error and asking for more input or further review), and allow for simple scenario analyses. The same TurboEngine will be used by field representatives, telephone service representatives, and customers who interact with us via the Internet, though the user-interfaces for the latter will probably be different from the one for SSA workers.

Data resides in a single virtual repository. Physically, it may consist of numerous constructs, including classical relational databases, modern data-stores such as Hadoop/HBase, or any of today's standard systems for storing and managing large sets of data with varying requirements. However, because of the virtual singleness of the data, everybody sees the same thing (when they have viewing rights, that is). Our data structure is designed from the ground up to optimally meet our special needs. For example, we often have to recompute from past data. A name today may not be the same as it was five years ago. Even a birth date today may not be the same as it was five years ago. And, importantly, a rule today may not be the same as it was five years ago. Data and rules for us will always comprise time interval components. This will help alleviate so much of the manual labor that is currently done during post-entitlement work.

We use standard technologies, and never rely on bleeding edge technologies. We do not innovate in technology; what seems like clever technological innovation with some cost saving today invariably becomes a costly albatross a few years down the road (think of SSA's MADAM). We use technology to help innovate business processes. Even though we are an information-based enterprise, we are not a technology company.

7. Transition Strategy

We plan the transition to the new architecture as a series of evolutionary steps. Each step of the bootstrap process takes the form of a transition project that advances the state of the architecture while preserving overall operational ability. The primary technological challenge is to find a way to gradually transition from current data stores to new ones that support the domain-centric view of service components. To mitigate risk, the solution must provide a fallback path that preserves information in current formats, should that be necessary at any point in the transition.

We retain SSA's current data stores during the transition period. Applications continue to operate as today until the new system allows us to retire them. Our CRM system rolls out in stages, office by office or region by region. The CRM system integrates its operations with new service components as they enter production.

Modern data integration tools will be used during the transition to build modular virtual databases to support our service components. These tools allow us to define dynamic mappings and transformations between virtual database views and underlying data stores. Changes to the underlying data made by older applications are instantly available to new service components in the format they wish to see it.

Changes made by service components are instantly available to older applications. Fallback is facilitated since data in existing stores always stays current. At the end of the transition period, we retire the old data stores and our use of data integration tools. We also need to create new databases to support the new service components and load them with information from current data stores and user input.

It is important to highlight that we do not intend to migrate SSA's current databases to our new databases. What we will do is populate the new databases only with data that is necessary for SSA to run its business. This is a simpler task than what SSA has been trying to do in migrating from MADAM to DB2. Current SSA databases contain a lot of redundancies and sometimes hard-to-reconcile data.

We can take advantage of successful efforts SSA has already made at virtualizing access to data. When several applications share a data access layer, it might be feasible to change that layer to base its operation on service operations rather than underlying databases.

Our planning needs to establish an order of transition projects, but we must remain flexible in the bootstrapping process. We begin by looking at SSA's business operation, understanding its individual parts and the relationships between them. We discover business domains – aspects of SSA's business that are relatively self-contained. We plan our transition by prioritizing our domains for implementation. A TurboEngine for Title II applications that can be accessed by end users over the Internet seems like a good candidate starting point. Based on work of my former team in the Office of Vision and Strategy at SSA, I am confident that this engine can fully automate a large percentage of online claims applications, and so this, by itself, will provide significant added value to the Agency. Domains that support the CRM system within SSA's larger enterprise also seem likely candidates for early implementation.

For each transition project, we build a set of service components that implement the qualities of its domain. We take care to structure their operations to reflect business requirements and not how business currently works. As part of this work, we define a virtual data model to reflect relationships within the domain. We use our data integration tools to create mappings between the virtual data model and SSA's current data stores. We may use a rules engine in components responsible for evaluating large sets of interrelated business rules.

Operational transition still needs to be explored. The new SSA will not need the vast majority of what is currently in its Processing Service Centers. CR training will change because the day-to-day operations of CR will change. Proof-of-concept stages will need to be planned and rolled out. Interactions between Policy and Systems will change. Better analytics and reporting will mean changes in performance monitoring. Software development paradigms will change. So far, while we have identified such operational issues, we are not ready to make recommendations regarding them.

At the conclusion of each transition project, the agency has completed another evolutionary step toward realizing its architectural vision. We must be flexible to allow for shifts from the ideal vision to meet constraints that arise as we develop and execute. As long as the final architecture is derived by

heading towards our ideal, even if we get to some place that is not exactly what we envision early on, we will still have a highly efficient, modern enterprise.

8. Conclusion

In a speech at the White House Forum on Modernizing Government (January 14, 2010), President Obama said, “We’ve got to get the best bang for every single dollar that the government has in its possession. And when Washington lags a generation behind in how we do business, that has real and serious impact on peoples’ lives.” SSA has a responsibility to the American people to recognize the dangers of continuing with its current way of doing business and face up to the hard challenge of creating a 21st Century agency. This paper not only highlights SSA’s problems, but also shines a light on a path to success. We, as a nation, have the talented people who are willing to work for the public good to get the job done. SSA’s leadership needs to overcome its fear of change, step up to the plate, and carefully but surely move forward.

Appendix

The following Social Security service capabilities do not exist today, nor will they be enabled if SSA continues with its current IT initiatives. They are all enabled by the proposed SSA-2020 plan, and can be delivered at costs significantly lower than today's.

- Individuals can securely transact almost any business with SSA via automated services. There are few exceptions.

- SSA automated services are available when the customer chooses, 24x7. There is no need for batch processing or backup windows. Software and hardware deployments and upgrades can occur at any time without affecting customer service. Since data and applications are kept current at multiple locations, software, hardware and telecommunications failures and anomalies are detected and managed without interruption to customer service or to SSA staff.

- SSA automated services are available through the machine interface of the user's choice. User interface management is completely divorced from business logic so that it becomes much simpler to accommodate a heterogeneous and changing universe of individual citizen and business partner interfaces. Development of user interfaces in multiple languages is much more readily supported.

- Individuals can easily obtain relevant information about SSA programs using automated services. Program information is communicated to individuals when they want it, in a consistent fashion, and uses automated methods to determine the most suitable yet complete answer to almost all program inquiries.

- Individuals have direct access in one place to their activity history at SSA. This includes pre-claim activity that an individual might choose to save both for future reference and to streamline future claims filings.

- Almost all interactions with SSA can be fully electronic. This includes all inquiries and claims transactions originating either with the individual or with SSA, and includes any notice or other formal communication created by SSA.

- Benefit estimates are made using the same code as claims. Given the same data, an estimate and a benefit payment are guaranteed to be the same. There are no surprises. The accuracy of an estimate is affected only by the information provided by the beneficiary.

- SSA's benefit estimate and claim applications are both intuitive and informative. Potentially complex decisions such as month of election are clearly understood using a combination of what-if scenarios and clearly written context driven help.

- Interactions with SSA are data driven. Only the data still needed to complete an SSA determination are requested.

- A potential beneficiary can make and optionally save for future use numerous benefit estimate scenarios, as needed. The desired scenario can be turned into an application with a mouse click. Information known to SSA or saved for future use by a potential beneficiary are all reused. Most benefit estimates convert to benefit applications with almost no additional input required.

- Almost all transactions are processed to completion online. At the end of the transaction, the individual has received official notice of SSA's decision and rationale, and if benefits are due, knows exactly how much will be paid and when, and what factors were considered in determining the benefit payment amounts. Program actions are event driven and batch processing for the most part no longer exists.

- A complete history is maintained of all transactions, including all evidence and business rules used to process the transaction at that time. All information is retained to the extent permitted and not deleted or replaced. Both customers and SSA staff can always see the source and effect of any change, what specific factors SSA considered, and be able to view a record as it appeared before and after any transaction, even where information has changed multiple times for the same payment event. There is no need for separate audit, history or archive files. Integrity management occurs as individual transactions are processed.

- Uniform transaction description and storage and complete transaction records permit automation of many notices. Notices are intelligible.

- SSA staff has a single facility to view all interactions for and information about an individual customer. This facility provides staff with a complete view of all completed, pending and diaried future activity for the individual. Multiple control lists and development worksheets no longer exist.

- SSA management has a complete view of work processed, pending and projected for their scope of their responsibility.

- Business intelligence is greatly enhanced with near real-time access to almost all available claims data. Only summary data is separately maintained from the authoritative data source. Summary data can always be traced back to the state of the individual specific records at the specific point in time used to produce the summary data.

- True modular managed development leads to applications developed faster at less cost and with few defects. Application development is based on a combination of modular services, active rules repositories and engines, and generated user interfaces. Small or moderate changes can often be rapidly released and without extensive testing of an entire system. Moderate and large changes can often be implemented by incorporating new rules and without extensive prescriptive coding.

- Application developers are primarily policy and business process experts. Systems experts concentrate more on the information technology infrastructure. The effect of policy and operational changes can be modeled prior to implementation. Most testing and validation is automated.

- SSA business services are available to internal and external partners for incorporation into their products. Centers of innovation and excellence within SSA can easily augment and transform major SSA business services. Private third parties can provide consolidated services to their clients, such as corporate retirement applications also including social security benefits. Policy issues permitting, SSA benefit application and maintenance activities can be consolidated or shared with other governmental entities.

- SSA's many data exchange activities become standards based and event driven. Data exchanges are simple to create and maintain. More inbound data exchange activities are initiated to obtain the additional data necessary to reduce program costs and error.

- SSA is information technology agnostic. SSA is not beholden to any particular vendor or hosting method. Products and processes are modularized so that interdependence between components is minimized. SSA can readily take advantage of best of breed or least cost products and platforms, as they evolve, without incurring significant conversion costs.