

The Small Footprint Design Method (SFDM): Augmenting Virtualization

Introduction

The Small Footprint Design Method (SFDM) increases the potential gains from virtualization by increasing software processing efficiency in the design phase. It is a software design approach sold as a service rather than a hardware or operating-system technology.

Virtualization software, such as VMWare, makes it possible to better utilize processing and memory resources by distributing software more fluidly across a set of hardware. Where SFDM shrinks the footprint of application software the use virtualization will aggregate that transactional efficiency and directly decrease hardware requirements proportionately. Without virtualization SFDM would require enough efficiency gain into remove an entire server unit, however with virtualization sub-unit efficiency gains can result in full server unit reductions.

In preliminary testing the Small Footprint Design Method (SFDM), an innovative design method for increasing application software processing efficiency, produces rapid payback and high ROI. A great deal of business value exists in the potential increase of software efficiency. Software efficiency determines underlying hardware costs and vis-a-vis administration costs.

SFDM & Application Software Design

Application software development typically begins with a design process. Because of the complexity of programming millions of lines of code, large application design deals first with organization: a way of breaking down and modularizing programming tasks for project management. This approach is practiced widely to varying degrees of detail.

As practiced, the design process tends to ignore system efficiency. A key reason for not considering efficiency is the absence of a systematic way of addressing this aspect of development. Mostly, efficiency questions are tacked only when inefficiency is observed in hindsight. As a result, a great deal of inefficiency remains unaddressed in software because it does not present enough of a problem to users. That inefficiency is effectively an untapped reservoir of hardware requirements that could be used better or turned into cost savings.

SFDM is a design approach that anticipates inefficiency and finds opportunities for efficiency. SFDM directly addresses the lack of a disciplined means of building efficiency into software at the design stage and without rework and missed opportunities. By definition it will reduce operational costs if practiced well. Indirectly, it will also reduce energy consumption because lower operational requirements for hardware for the same software output means less usage of electricity, another cost savings and a reduction of carbon footprints additionally.

SFDM is short for "Small Footprint Design Method"; an innovative process that Process Intelligence is seeking patent protection for. SFDM provides a way for shrinking software footprints without reducing or changing performance or functionality. This design method was developed independently within Process Intelligence; however it would be better commercialized in association with a larger software company. This proposal aims to prove SFDM to potential partners and build such a partnership for commercialization of this innovation for the 2010 timeframe.

We believe that the audience that would be most interested in this work are organizations with large scale custom application development responsibilities. These might include:

- professional service groups providing outsourced application development and consulting (IBM Global Services, HP/EDS, Cap Gemini and others)
- large social networking, search and consumer oriented websites (Google, Amazon, Facebook, Wikipedia)
- emerging digital health care system providers
- large automated systems management organizations, for example electric grid utilities
- large data-intensive companies (banks, financial companies, insurers, etc.)

With respect to the risks of adopting this innovative method, SFDM fits easily into widely accepted project management practices including CMMI and Agile Development. SFDM assumes that design exercises and object modelling are practiced to some degree within a project.

SFDM impacts projects little:

- very low cost relative to project since the analytical portions of SFDM can typically be done by one consultant working over several weeks and translated into routine design changes
- can be performed largely in parallel to avoid schedule impact; in fact it may quicken schedule if improvements preclude need for performance engineering at the end of the project
- is generally practiced by a consulting performance architect for use by project architects

SFDM is practiced at the latter half of a design phase and does the following:

- expands design transactionally
- optimizes boundary conditions transactionally
- guides ordinary optimizations within predicted “hotspots”
- enables ordinary low cost lighter design options for “coldspots”

SFDM extends the Object Modelling approach expressed typically with UML (Unified Macro Language) notation in software application designs. Specifically it extends state machine modelling into a disciplined basis for efficiency optimizations. With this grant and with respect to Object Modelling and UML Process Intelligence proposes to ultimately create and submit an SFDM-based UML Profile Specification to a standards group while developing professional services business opportunities based on that innovation. As a basis for that submission, Process Intelligence hopes to expand the number and size of test cases of SFDM initially in a test setting and then in commercial settings. Because of the critical nature of application development to businesses Process Intelligence expects that a deep set of supporting test data will be needed to gain the confidence of commercial application development sponsors in order for SFDM to be purchased as a service on a large scale.

How SFDM Can Fit Into Accepted Development Standards

Large application development is carried out with great care among its practitioners. SFDM will not receive acceptance purely because it has merit as an idea. To gain entry into an actual large scale project will inevitably require intermediate steps to develop a partnership with interested firms. For example, support from the NSF will help Process Intelligence expose the use of SFDM to large firms through industry consortia, such as The Object Management Group and The OpenGroup. Process Intelligence intends to present SFDM to these consortia to get those

intermediate partnerships in place. Process Intelligence will also extend proof-of-concept examples from what they are today to a more sophisticated form that addresses any unexpected concerns voiced by these groups.

The Object Management Group is an international computer industry consortium which develops standards, notably publishes a large number of UML-based standards.¹

Unified Macro Language (UML) is a standardized general-purpose application software modelling language widely used to logically map user requirements to programming design artifacts and ultimately to software components.² UML is a widely-practiced means of expressing object models, a staple of application software architecture. A variant of UML, SysML enables performance analysis within system models however SFDM is aimed at software design, not systems, and thus needs to integrate itself into large application software design practices which are commonly based on UML.

Commercial Potential

In a typical application software design and development project estimated to have total costs in excess of \$2 Million and greater than 4 production application servers, an SFDM consultation would cost approximately \$50,000.

Application development is a huge professional services market with worldwide annual sales of \$7Billion dollars³ conservatively. The larger IT enterprise software industry has a great deal of application development which would benefit from SFDM and that market exceeds \$200 Billion in sales annually.⁴ As a standard practice any large application development project will typically include a design phase that can consume from 10% to 30% or more of a project budget.⁵ Within the design phase, accepted methodologies are practiced and comprise the essence of that work.

Software design services aimed at producing low-energy software would be sold to this market as a value-add service. The use of this method in a design phase would translate to:

- business expansion for the sponsor of this proposal (Process Intelligence, Inc.)
- competitive sales advantage by reselling the service (Professional Services partners),
- cost savings for their clients
- energy savings for the nation.

Bottom-up analysis suggests that the data-collection, heuristics, recommendations and final review needed to accomplish the goals built into SFDM would take approx 8 person-weeks over the course of a project and would require senior architect level resources typically billing near \$180/hr. This would total closer to \$57,600 per project and would be closer to 5% of the project budget.

It is fair to expect that at a cost of \$50,000 in SFDM services a large scale application would fully

1 [Http://www.omg.org](http://www.omg.org)

2 [Http://www.omg.org/cgi-bin/doc?ad/97-08-11](http://www.omg.org/cgi-bin/doc?ad/97-08-11)

3 IBM and Gartner Group, 2007, “*IBM Number One in Worldwide Application Development Software Market Share*”, http://presszoom.com/print_story_133078.html

4 Gartner Group, 2009, “*Gartner Says Worldwide Enterprise Software Market to Experience Flat Growth in 2009*”, <http://www.gartner.com/it/page.jsp?id=923312>

5 Robert L. Scheier, 2005, “*Visual Modeling’s New Look*” <http://adtmag.com/Articles/2005/03/01/Visual-Modelings-new-look.aspx?Page=3>

recover that cost by lowering hardware capital expenses for near-immediate payback and allowing recurring energy and admin staffing costs to contribute fully to ROI from the outset.

Top-down analysis assuming:

- \$1.2 Million for applications⁶, thus market = 7000 applications / year
- 10% of applications are clear candidates, 25% are strong & less-obvious
- \$50,000 / project, cost of consulting SFDM services

suggests a market of \$35 Million dollars worldwide for a service such as SFDM expanding to \$100 Million dollars with full industry awareness. Some prospective application development organizations may find comparable ways to approach the same problem without SFDM, however there would be a concrete advantage to licensing an established technique, such as we expect SFDM to become.

Using virtualization as a measuring tool, according to IDC⁷ the virtualization market is about \$2 Billion dollars in size. The effect of SFDM from a revenue standpoint should compare with virtualization. That is, the money spent on virtualization in a data-center, perhaps \$50,000/yr, compares to the potential use of services at 50,000 / per application assuming that the same data center receives a major update of one of its applications per year.

The incentive for a company to buy application development services from a company that has adopted SFDM would be cost reduction. The hardware reductions possible through the use of SFDM combined with virtualization would come in several well-accepted and well-proven forms:

- Electricity expenses
- Hardware purchases, including servers, storage, network devices
- Server administrative costs

The value proposition to the end client, in contracting with an application development firm that uses SFDM or in using SFDM directly, is:

- in a mirrored production environment with stress test and functional test systems a reduction of 2 servers (one in main production the other in DR) multiplies to potentially four servers.
- A reduction of a single server will save approximately \$15,000 before an application is actually deployed, thus 4 servers provides immediate payback on a \$50,000 consultation
- Recurring maintenance, electric and admin costs per server approximate to \$7500/year. Multiplied out for 4 servers it would return the original outlay as pure ROI every 2 years on a recurring basis.
- The use of SFDM locates ways of decreasing excessive development effort in cases where systems are overdesigned for performance. For example, rarely-used function can be separated from the elaborate development patterns used for high-performance functionality. Using SFDM in this manner can reduce development costs outright.
- The use of SFDM can reduce costly performance engineering exercises which can idle large teams for weeks at the end of a project at great expense

⁶ C. Larman, 2004, Agile and Iterative Development: A Managers Guide, Addison Wesley

⁷ IDC, 2006, http://virtualizationdaily.com/archives/54_virtualization-market-booming-according-to-idc.html