

**Project Name:** Software SRED - General guidelines  
**Project Number:** 1

**Start Date:** 2008-01-01  
**Completion Date:** 2008-12-31

## Key Criteria Summary

Sample Software Industry

### 1 - Software SRED - General guidelines

**Benchmarks:** Internet searches: 12 sites / articles  
 Patent searches: 43 patents  
 Competitive products or processes: 3 products  
 Potential components: 7 products  
 Queries to experts: 5 responses

**Objectives:** find the proper use of a nested model: 5 Error messages  
 Reduce power consumption: 1.25 V per hour

**Uncertainty:** 1 - Technological Uncertainty-II Key evidence examples  
**Key Variables:** Concurrency, Inter-operability, Legacy Requirements, Performance, Stability

Activity	Testing Methods	Results - % of Objective	Variables Concluded	Hours	Materials \$	Subcontractor \$	Fiscal Year
1 - Eligible Activities	Analysis / simulation: 2 alternatives	(none)	Concurrency Inter-operability Legacy Requirements Performance Stability	155.50	99.00	124.00	2008

**Uncertainty:** 2 - Will the integration of the new software reduce physical power consumption  
**Key Variables:** (none)

Activity	Testing Methods	Results - % of Objective	Variables Concluded	Hours	Materials \$	Subcontractor \$	Fiscal Year
	(none)	(none)	(none)	0.00	0.00	0.00	

### 801 - Integrate SQL dbase, x-n, & internet servers (NEW FORMAT)

**Benchmarks:** Patent searches: 3 patents  
 Similar prior in-house technologies: 1 products /  
 Potential components: 3 products  
 Queries to experts: 2 responses

**Objectives:** Throughput: 20 events/s

**Uncertainty:** 1 - Database and Transaction Server Interaction  
**Key Variables:** concurrency, CPU usage, Footprint, Memory usage/thread handling, throughput

Activity	Testing Methods	Results - % of Objective	Variables Concluded	Hours	Materials \$	Subcontractor \$	Fiscal Year
1 - System modeling	Analysis / simulation: 1 alternatives Physical prototypes: 3 samples	Throughput: 30 events/s (152 %)	concurrency CPU usage Footprint Memory usage/thread handling throughput	425.00	0.00	3.00	2008

**Uncertainty:** 2 - the application of generalized linear models to data  
**Key Variables:** concurrency, CPU Usage, footprint, Memory usage/thread handling, throughput

Activity	Testing Methods	Results - % of Objective	Variables Concluded	Hours	Materials \$	Subcontractor \$	Fiscal Year
	(none)	(none)	(none)	0.00	0.00	0.00	

**Project Name:** Software SRED - General guidelines  
**Project Number:** 1

**Start Date:** 2008-01-01  
**Completion Date:** 2008-12-31

**802 - Database methodology**

**Benchmarks:** Internet searches: 21 sites / articles  
 Patent searches: 14 patents  
 Similar prior in-house technologies: 1 products /

**Objectives:** Access speed with large database: 15 s

**Uncertainty:** 1 - Relational Data Model Analysis - [Supporting Act.] **Key Variables:** performance

Activity	Testing Methods	Results - % of Objective	Variables Concluded	Hours	Materials \$	Subcontractor \$	Fiscal Year
1 - Literature Review	Analysis / simulation: 4 alternatives	(none)	performance	502.00	123.00	12.00	2008

**Uncertainty:** 2 - Comm model vs. Relational Environment **Key Variables:** performance

Activity	Testing Methods	Results - % of Objective	Variables Concluded	Hours	Materials \$	Subcontractor \$	Fiscal Year
1 - Data Communications Model Analysis	Process trials: 1 runs / samples	(none)	performance	12.00	12.00	17.00	2008

**Uncertainty:** 3 - Relational Access + Packet Access Combination **Key Variables:** performance

Activity	Testing Methods	Results - % of Objective	Variables Concluded	Hours	Materials \$	Subcontractor \$	Fiscal Year
1 - Model Comparison Tests	Process trials: 7 runs / samples	(none)	performance	16.00	0.00	328.00	2008
2 - Hybrid Model Attempt	Physical prototypes: 1 samples	Access speed with large database: 10 s (133 %)	performance	0.00	0.00	0.00	2009

**803 - Network failure problems**

**Benchmarks:** Internet searches: 12 sites / articles  
 Patent searches: 4 patents  
 Similar prior in-house technologies: 1 products /

**Objectives:** Concurrent accesses: 1750

**Uncertainty:** 1 - System Uncertainty **Key Variables:** determining cause of network failure, integration of components, maintaining response time

Activity	Testing Methods	Results - % of Objective	Variables Concluded	Hours	Materials \$	Subcontractor \$	Fiscal Year
1 - Development and Testing	Analysis / simulation: 5 alternatives Process trials: 32 runs / samples Physical prototypes: 1 samples	Concurrent accesses: 1900 (115 %)	determining cause of network failure integration of components maintaining response time	159.25	8.00	66.00	2008

**901 - Compression algorithm**

**Benchmarks:** Internet searches: 5 sites / articles  
 Competitive products or processes: 1 products  
 Similar prior in-house technologies: 1 products /

**Objectives:** Compression of 1 Mb map: 30 K

**Uncertainty:** 1 - Optimal compression method **Key Variables:** compression

Activity	Testing Methods	Results - % of Objective	Variables Concluded	Hours	Materials \$	Subcontractor \$	Fiscal Year
1 - Development and Testing	Physical prototypes: 4 samples	Compression of 1 Mb map: 30 K (100 %)	compression	81.25	334.00	134.00	2009

**Project Name:** Software SRED - General guidelines  
**Project Number:** 1

**Start Date:** 2008-01-01  
**Completion Date:** 2008-12-31

**902 - Software Data Warehouse Description Development**

**Benchmarks:** Internet searches: 33 sites / articles  
 Competitive products or processes: 6 products  
 Similar prior in-house technologies: 2 products /  
 Potential components: 100 products  
 Queries to experts: 3 responses

**Objectives:** CPU Utilization: 70 % busy  
 Response Time: 15 seconds  
 Data to compression: 15 :1 ratio

**Uncertainty:** 1 - Uncertainty #1: : Non-uniform dataset determination

**Key Variables:** CPU Utilization, Definition and construction of data blocks, Definition of construction of data blocks, Methods to characterize non-uniform data, Optimal use of compression dictionary

Activity	Testing Methods	Results - % of Objective	Variables Concluded	Hours	Materials \$	Subcontractor \$	Fiscal Year
1 - Develop generic data model	Analysis / simulation: 10 alternatives	(none)	Definition and construction of data blocks	735.00	850.00	13,966.00	2008
2 - Develop compression methods	Physical prototypes: 10 samples	(none)	CPU Utilization Definition of construction of data blocks	700.00	600.00	0.00	2008
3 - Compression algorithm with dynamic techniques	Physical prototypes: 3 samples ... prototype revisions: 12 revisions	Response Time: 22 seconds (84 %) Data to compression: 131 :1 ratio (1260 %) CPU Utilization: 66 % busy (116 %)	(none)	695.00	6,000.00	2,550.00	2008
4 - Extend data compression methods	Process trials: 102 runs / samples	(none)	CPU Utilization Definition and construction of data blocks Optimal use of compression dictionary	1,230.00	0.00	0.00	2008
5 - Correlate compression block size with initial data set	Analysis / simulation: 22 alternatives	(none)	CPU Utilization Definition and construction of data blocks Methods to characterize non-uniform data	375.00	0.00	3,750.00	2008

**Uncertainty:** 2 - generalized linear models take count data from paired comparisons what is the outcome?

**Key Variables:** linear models, paired comp

Activity	Testing Methods	Results - % of Objective	Variables Concluded	Hours	Materials \$
	(none)	(none)	(none)	0.00	0.00

**Project Name:** Software SRED - General guidelines  
**Project Number:** 1

**Start Date:** 2008-01-01  
**Completion Date:** 2008-12-31

**1001 - Scaling vs. speed vs. compression**

**Benchmarks:** Competitive products or processes: 1 products  
 Potential components: 7 products  
**Objectives:** Encoding rate: 5 s/G

**Uncertainty:** 1 - Determining optimal trade-off  
**Key Variables:** compression, computing platform, scaling, speed

Activity	Testing Methods	Results - % of Objective	Variables Concluded	Hours	Materials \$	Subcontractor \$	Fiscal Year
1 - Encoding algorithm	Process trials: 52 runs / samples	Encoding rate: 5 s/G (100 %)	scaling speed compression computing platform	0.00	0.00	0.00	2010

**1002 - Improve Vista**

**Benchmarks:** Internet searches: 5 sites / articles  
 Patent searches: 2 patents  
 Competitive products or processes: 2 products  
 Similar prior in-house technologies: 6 products /  
**Objectives:** Avg. Laptop Battery Life: 5.3 hours  
 Avg. Load Time: 25 seconds  
 Avg. Time Between Disruptions (crashes): 40 hours

**Uncertainty:** 1 - Meeting objectives given architecture complexity, 3rd party issues  
**Key Variables:** (none)

Activity	Testing Methods	Results - % of Objective	Variables Concluded	Hours	Materials \$	Subcontractor \$	Fiscal Year
1 - Collecting and analyzing user data	Analysis / simulation: 5000 alternatives Physical prototypes: 1 samples ... prototype revisions: 7 revisions	(none)	(none)	0.00	0.00	0.00	2010
2 - Improving stability & reliability	Physical prototypes: 2 samples ... prototype revisions: 37 revisions	Avg. Time Between Disruptions (crashes): 42 hours (108 %)	(none)	0.00	0.00	0.00	2010
3 - Optimizing windows management	Physical prototypes: 1 samples ... prototype revisions: 4 revisions	(none)	(none)	0.00	0.00	0.00	2010
4 - Decreasing load time	Physical prototypes: 1 samples ... prototype revisions: 3 revisions	Avg. Load Time: 27 seconds (83 %)	(none)	0.00	0.00	0.00	2010
5 - Improving battery life	Process trials: 10 runs / samples Physical prototypes: 3 samples ... prototype revisions: 8 revisions	Avg. Laptop Battery Life: 5.32 hours (102 %)	(none)	0.00	0.00	0.00	2010

**Project Name:** Software SRED - General guidelines  
**Project Number:** 1

**Start Date:** 2008-01-01  
**Completion Date:** 2008-12-31

---

**Software Industry**

The following guidelines are intended to provide examples of “experimental development” projects which would qualify for Canadian SR&ED (Scientific Research & Experimental Development) tax credits.

**Content Summary:**

1 - Software SRED - General guidelines:..... 6

801 - Integrate SQL dbase, x-n, & internet servers (NEW FORMAT):..... 11

802 - Database methodology:..... 14

803 - Network failure problems: ..... 17

901 - Compression algorithm: ..... 20

902 - Software Data Warehouse Description Development:..... 22

1001 - Scaling vs. speed vs. compression: ..... 25

1002 - Improve Vista:..... 27

**Project Name:** Software SRED - General guidelines  
**Project Number:** 1

**Start Date:** 2008-01-01  
**Completion Date:** 2008-12-31

## **1 - Software SRED - General guidelines:**

### **Scientific or Technological Objectives:**

<b>M e a s u r e m e n t</b>	<b>C u r r e n t P e r f o r m a n c e</b>	<b>O b j e c t i v e</b>
find the proper use of a nested model (Error messages)	12	5
Reduce power consumption (V per hour)	1.75	1.25

[NOTE: THESE GUIDELINES ARE REPRODUCED FROM EXERPTS OF, "GUIDANCE ON ELIGIBILITY OF SOFTWARE PROJECTS FOR THE SR&ED TAX CREDITS," AS PUBLISHED BY THE CRA IN CO-OPERATION WITH CATA & THE SOFTWARE INDUSTRY, SEPTEMBER 2000.]

Advancement - Note that an advancement in technology can rarely be described by listing software functions and features at an "end-user" level. Advances are typically made through innovation in software architectures, designs, algorithms, techniques or constructs within the field of information technology or computer science. The advancement need not be large.

Note: Simply claiming to have developed the first or best software suite for a given purpose does not in itself prove that the taxpayer has made a technological advancement. A new and unique software suite can be built using only well known combinations of constructs, tools and methods without technological advancement. This is analogous to designing and building a unique and complex office building without making any advancements in the field of civil engineering.

Evidence of Technological Advancement could include credible third party literature or comparisons of the capabilities sought against those previously available from the taxpayer himself. As in a court of law, there are no rigid definitions of what constitutes credible evidence.

### **Technology or Knowledge Base Level:**

Benchmarking methods & sources for citations:

- Internet searches: 12 sites / articles -- We searched extensively and found no out of the box solutions to meet our criteria
- Patent searches: 43 patents -- We looked at 43 patents that were similar to our requirements but were insufficient
- Competitive products or processes: 3 products -- We looked at 3 other companies working with similar technologies
- Potential components: 7 products -- There are seven potential components we are looking at using
- Queries to experts: 5 responses -- We spoke to Derek Tarko P. eng He was helpful but advised us this wasn't available

Hint: As a means to identify the advancement(s), the taxpayer might identify the technological reason why his architecture or technique was not used before. How does it compare with earlier solutions or with the current solution of a competitor? What earlier technical constraint has been overcome?

### **Field of Science/Technology:**

Software (1.02.03)

### **Intended Results:**

- Develop new processes
- Develop new materials, devices, or products
- Improve existing processes
- Improve existing materials, devices, or products

### **Scientific or Technological Advancement:**

#### **Uncertainty #1: Technological Uncertainty-II Key evidence examples**

The objective here is to outline options for developing sets of questions which may act as catalyst to provide an effective and efficient method of identifying key evidence of eligibility.

1. Identify the limitations/constraints imposed by the technology components being utilized. What technical challenges did these constraints create?
2. Identify the degree of control the claimant has to modify the technology components. What technical challenges did these constraints create? Examples:

**Project Name:** Software SRED - General guidelines  
**Project Number:** 1

**Start Date:** 2008-01-01  
**Completion Date:** 2008-12-31

---

- Are you using any of the components in a unique, previously undocumented or unconventional fashion?
- Is the vendor able to confirm the suitability of these components for use in said fashion?
- Is the vendor capable of providing a deterministic description of the components predicted response when used in this unique fashion?

[NOTE: THE CRA FINDS THIS TYPE OF THIRD PARTY EVIDENCE VERY VALUABLE AS SUPPORTING EVIDENCE THAT THE WORK INVOLVED A "DEPARTURE FROM STANDARD PRACTICE." AS SUCH WE RECOMMEND THAT THIS EVIDENCE BE SAVED WHENEVER POSSIBLE.]

3. Identify the constraints or uncertainties or paradoxes presented when certain components/objects/technology platforms are operated in conjunction with other software entities. Do you have control over these interactions, can you or the vendors of these components predict the effects of these interactions?

4. Identify any constraints resulting from considerations of;

- Inter-operability
- Conformance to standards
- Performance (step response, throughput)
- Concurrency
- Footprint
- Scale-ability
- Stability
- 3rd party components
- legacy requirements

What technical challenges did these constraints create?

5. Identify any key characteristics of a technology platform you are using to which the manufacturer of the technology component cannot provide a fully deterministic characterization of the platform when utilized in the fashion required by your project.

6. Is the integrated performance of the software components incorporated within the project fully deterministic? I.E. can the behavior of the components be fully projected both on a stand alone basis as well as when operating within an integrated environment? Can you predict the desired outcome? If not why not?

7. What technology risks/constraints/problems appeared after the project began?

8. What was or will be hard or technically difficult to do & why?

9. What restrictions are presented by the attributes of objects/components or the API's presented by components on environmental platforms such as operating systems?

Related issues to illustrate via research steps & conclusions:

10. If you had to do it again what would you do differently?

11. What technical alternatives did you look at, what did you discard & why?

12. What are the technical design trade-offs associated with these alternatives?

13. What are/were the possible technical outcomes other than the results you are seeking?

The most significant underlying key variables are:

Inter-operability, Performance, Concurrency, Stability, Legacy Requirements

---

## **A c t i v i t y # 1 - 1 : E l i g i b l e A c t i v i t i e s**

**Work performed in Fiscal Year 2008:**

**Methods of experimentation:**

- Analysis / simulation: 2 alternatives - Design of Experiment involves designing a set of ten to twenty experiments, in which all relevant factors are varied systematically. When the results of these experiments are analyzed, they help to identify optimal conditions, the factors that most influence the results, and those that do not, as well as details such as the existence of interactions and synergies between factors.

**Project Name:** Software SRED - General guidelines  
**Project Number:** 1

**Start Date:** 2008-01-01  
**Completion Date:** 2008-12-31

•  
III a) Problem Statement & CRA opinions:

Some claimants believe that debugging is experimental work as called for by the definition of SR&ED in that it involves a series of trials and observations. Some claimants believe that all software development is experimental for the purposes of the Act by the very nature of the activity.

The CRA's and the Department of Finance's position is that some software development does not meet the SR&ED guidelines because it is not basic research, or applied research or experimental development. Software that is developed through routine design activities in projects which have only routine design challenges does not meet the SR&ED criteria of Technological Advancement and Technological Uncertainty.

Discussion: An experiment within the context of the SR&ED Program involves setting up test conditions and making observations or measurements aimed at filling gaps in our technical knowledge. The result of the experiment, whether it is successful or unsuccessful, provides an increase in knowledge of software systems relative to the Technological Advancement sought and/or the Technological Uncertainties.

The new knowledge is applicable beyond the system under test. Thus inherently, Technological Uncertainties are associated with advancements in technology knowledge. One making a claim should always be able to identify the technological advancement in his knowledge that is associated with solving a technological uncertainty, i.e. what was learned through experimentation.

In software development, immediate problems are usually solved by "trial-and-error" rather than by experiment in the sense of the Income Tax Act. Trial-and-error involves executing a series of probes that were not sequenced in a systematic pre-plan. The objective here is to resolve a functional problem (as in routine debugging) rather than to gain understandings that are expected to be more widely applicable. The lesson learned in each iteration of "trial and error" is simply "that an option didn't work" and they are not applicable in a much broader sense. For each iteration the probe is chosen that is now judged to be the most efficient in resolving the immediate problem. The process proceeds quickly from iteration to iteration.

Resolving problems through the "trial-and-error" approach is eligible support work, but it is not the basis for a Technological Advancement, as the knowledge gained does not produce a true improvement in our understanding of the technologies.

**Results:**

However, this model may result in the Product or City term having 0 degrees of freedom in the output given by the computer program. Again, an "error message." While Product and City are crossed effects and the interaction is correct, panelists are nested within city (different panelists in different cities). Thus, the correct model is: Source of Variation

City  
Panelist within City  
Product  
Product\*City  
Error df  
1  
98  
2  
2  
196

Consequently, recheck the appropriateness of your model when faced with non-sensical analysis results from your computer package.

Most statistical software packages are completely dependent upon the user regarding the selection and implementation of the correct model to analyze. Whatever model the user defines, the computer will "fit" that model. Incorrect conclusions can be drawn when inappropriate models are fit to the data. The difficulty is that the computer program cannot inform you when the model is inappropriate.

Some common errors in model specification include the use of interactions without replication, the use of interactions with incomplete or missing data, and the misuse of nested models. Fortunately, many software packages give you hints that you have misspecified the model and many errors can be avoided by properly interpreting these hints.

For instance, a model is fit and the output provides no estimate of the Mean Square Error (MSE) and no f-tests or p-values are given. The model as specified was correctly fit to the data, so no error messages are seen by the user. However, the lack of f-tests is itself an error message indicating that all the degrees of freedom (df) have been exhausted leaving nothing for the MSE. This commonly happens when interactions are fit without replication in the data. Suppose 50 consumers each saw 4 products and the following model was fit: Source of Variation

**Project Name:** Software SRED - General guidelines  
**Project Number:** 1

**Start Date:** 2008-01-01  
**Completion Date:** 2008-12-31

---

Panelist  
Product  
Panelist\*Product

**Conclusion:**

III b) Technological Conclusions

[AUTHOR'S NOTE: THE IDEAL CONCLUSIONS WOULD BRIEFLY DETAIL HOW THE RESULTS COMPARED WITH INITIAL EXPECTATIONS AND OUTLINE ANY FURTHER CONCLUSIONS WHICH COULD AFFECT FUTURE DEVELOPMENTS OF THIS NATURE.]

Cut-off analysis - In the context of software development and the legislation, experiments might be aimed at resolving design or architectural alternatives or systematically probing an inadequately specified interface. The experimental approach itself should be designed. Debugging activity aimed at discovering coding errors and misunderstandings of interface requirements (call sequences, parameters, data formats, etc.) generally do not lead to Technological Advancements and embody only routine design uncertainties. Tests (in the trial-and-error mode) are designed and executed in a rapidly evolving sequence rather than being pre-planned to settle a design alternative. These activities are not "aimed at filling technological gaps" within the meaning of those words in IC97-1 and are thus ineligible in their own right.

Nevertheless, if this debugging activity is part of any software development. It can be claimed as eligible supporting activity when it is essential to the demonstration of a Technological Advancement and the resolution of Technological Uncertainty.

III C) Technical documentation retained (evidence):

The software paper also provides the following examples & guidance on sources of SR&ED software evidence. Ideally, these would be briefly referenced within the? Activities? themselves.

- Developer's Journals:

Lab books, personal day timers, electronic journal entries, or any other form of short form individual notation by technical staff can be used as supporting documentation.

- Architectural Design/ High Level Design documents:

Documents of this nature typically provide a "structural overview" of the project's objectives from a purely technological perspective including some level of analysis with respect to the alternatives considered, as well as providing a foundation or rationale with respect to the constraints within which a solution is being sought.

- Design Review Minutes

- Written Correspondence:

Correspondence of all forms (Letters, Email, Fax, memos) between team members as well as external software technology vendors can provide a wealth of evidence. In particular correspondence with vendors can often contain evidence of state of the art, or confirmed technological uncertainty. Team members' correspondence can provide evidence of a systematic experimental process, as well encapsulate technological objectives and advances sought.

- Performance Requirement Specifications

- Test results

- 3rd Party Documentation:

This documentation often contains API specifications standards for the software components in question as well detailed descriptions of the pertinent internal architecture and mechanisms embodied by the technology.

- Defect tracking records

- Test plans

- Project Schedules & Resource allocations

- Source code:

While source code provides the root evidence of technological advances achieved, it is generally not considered an effective medium for conveying eligibility. Within the context of an eligible SRED project and with significant effort on behalf of the claimant and the CCRA technical reviewer, source code can provide evidence of technological advances being sought in the presence technological uncertainty. When relying primarily on source code for evidence, the presence of a complete revision history and incremental archiving of intermediate revisions of source code is very useful to the establishment of technological advance through the use of experimental development.

**Project Name:** Software SRED - General guidelines  
**Project Number:** 1

**Start Date:** 2008-01-01  
**Completion Date:** 2008-12-31

---

- Software Development Methodology Standards:

These are documents which formally outline the methodology or process upon which any given software development project within the claimant's operation will be guided. These types of standards usually dictate a series of other documents and process milestones to be recorded. Identification of this type of document provides significant evidence of a mature scientific research or experimental development processes being utilized. These documents usually provide a road map to the majority of other key work product entities generated within the development process.

Key variables resolved: Concurrency, Inter-operability, Legacy Requirements, Performance, Stability

**Uncertainty #2: Will the integration of the new software reduce physical power consumption**

**There are no Activities associated with this uncertainty.**

**Project Name:** Integrate SQL dbase, x-n, & internet servers (NEW FORMAT)  
**Project Number:** 801

**Start Date:** 2008-01-01  
**Completion Date:** 2008-12-31

## **801 - Integrate SQL dbase, x-n, & internet servers (NEW FORMAT):**

### **Scientific or Technological Objectives:**

<b>M e a s u r e m e n t</b>	<b>C u r r e n t P e r f o r m a n c e</b>	<b>O b j e c t i v e</b>
Throughput (events/s)	1	20

[NOTE: THIS EXAMPLE IS REPRODUCED FROM, "GUIDANCE ON ELIGIBILITY OF SOFTWARE PROJECTS FOR THE SR&ED TAX CREDITS," AS PUBLISHED BY THE CRA IN CO-OPERATION WITH CATA & THE SOFTWARE INDUSTRY.]

[NOTE - THIS IS AN EXAMPLE OF HOW THE EVIDENCE NECESSARY TO SUPPORT THE CLAIM TYPICALLY ARISES NATURALLY FROM THE STANDARD DOCUMENTATION AND WORK PRODUCTS OF A GIVEN SOFTWARE DEVELOPMENT EFFORT.]

[AUTHOR'S NOTE: IDEALLY THE TAXPAYER WOULD ATTEMPT TO QUANTIFY THE OBJECTIVES THEY ARE TRYING TO ACHIEVE. A QUANTIFIABLE OBJECTIVE HAS BEEN ADDED ABOVE, TO ILLUSTRATE.]

A large container rental company is developing a custom, geographically distributed, transaction based, enterprise wide, operations, reservations, billing, and inventory yield management system. The new system will replace an ageing and simple UNIX terminal based main-frame reservation and contract recording system.

The nature of the SR&ED problem appeared in the later half of the project as a result of unexpected interactions between the transaction server component technology and the SQL database technology. Within the given architecture the two components combined to constrain the manner and mechanisms related to the level of granularity at which the SQL database could undertake record locking within a given table(s). The end result was an unexpected and severe impairment with respect to both the concurrency and throughput as it pertained to the processing of transactions.

### **Technology or Knowledge Base Level:**

Benchmarking methods & sources for citations:

- Patent searches: 3 patents -- Nothing found
- Similar prior in-house technologies: 1 products / processes -- Existing system is a simple UNIX terminal based main-frame reservation & contract recording system.
- Potential components: 3 products -- We looked at 4 potential components
- Queries to experts: 2 responses -- We spoke to 2 software engineers and there is no off the counter solution

The independent consultants made the recommendation to develop rather than purchase a system as 3rd party solutions did not feature required functionality.

[EVIDENCE - PERTAINING TO THE STATE OF THE ART IS THE REPORT EXAMINING THE SOLUTIONS AVAILABLE AND THE RECOMMENDATION TO MAKE VERSUS BUY.]

The claimant did not have the internal development expertise to necessary to design and implement the new system, and consequently subcontracted a respected Canadian software development firm to undertake the project.

[EVIDENCE - RETENTION OF SUCH A DEVELOPMENT FIRM PROVIDES EVIDENCE OF ACCESS TO QUALIFIED PERSONNEL WHICH IN TURN RELATES DIRECTLY TO THE VALIDITY OF ADVANCEMENTS SOUGHT AND UNCERTAINTIES ENCOUNTERED.]

The new system architecture was implemented utilizing object oriented software technology components in an N Tier thin client configuration. The functional requirements with respect to transactional, reporting, and yield management processes for the system resulted in the requirement to support very complex transactions. This in turn required the implementation of a very large and complex database schema.

[NOTE: IDEALLY, WE WOULD TRY TO QUANTIFY THIS DATABASE ENVIRONMENT AND VARIABLES IN QUESTION].

### **Field of Science/Technology:**

Software (1.02.03)

### **Intended Results:**

**Project Name:** Integrate SQL dbase, x-n, & internet servers (NEW FORMAT)  
**Project Number:** 801

**Start Date:** 2008-01-01  
**Completion Date:** 2008-12-31

- Improve existing processes

## Work locations:

Commercial Facility

## Scientific or Technological Advancement:

### *Uncertainty #1: Database and Transaction Server Interaction*

The development team undertook a series of testing and corrective actions but was unable to isolate the root cause for the combined interaction behaviour of the database and transaction server technology components.

The most significant underlying key variables are:  
concurrency, throughput, Footprint, CPU usage, Memory usage/thread handling

### **A c t i v i t y # 1 - 1 : S y s t e m m o d e l i n g**

#### Work performed in Fiscal Year 2008:

##### Methods of experimentation:

- Analysis / simulation: 1 alternatives - The use of PLS models eliminates drawback of both Ordinary Least Squares (OLS) Regression and Principal Component Regression (PCR). OLS requires more samples (products) than variables to be included in the model, which is typically not the case when attempting to compare consumer and descriptive data. In fact, it is not unusual to present only 8-10 products to the consumer and descriptive panels yet collect information on a couple dozen descriptive attributes. PCR eliminates this difficulty, as well as the problem of multicollinearity that is present in much of the descriptive data. However, the first principal component, formed from the descriptive data, is not necessarily related to consumer acceptance thereby weakening the model for predictive purposes. The use of PLS through Preference Cluster Mapping eliminates both of these difficulties. Further, because it still fits a model to the data it remains possible to predict consumer liking of new products from the existing model by simply running additional descriptive panels.
- 
- 
- Physical prototypes: 3 samples - Each of the 3 potential solutions was then implemented and tested. The solution which showed the most improvement was then further refined using another series of experiments.

The development team contacted the vendor of the components (which in this case was common to the Operating system, SQL database, Transaction server and Internet server software technology components) and requested assistance with the problem. The vendor investigated the problem and made several recommendations in an attempt to solve the problem, but was unable to direct the company to a solution to the problem.

None of the directives from the vendor were able to correct the system performance. In fact during the course of the investigation the vendor was unable to accurately predict the resulting system performance with respect to several of the suggestions they made.

[EVIDENCE OF THE TECHNOLOGICAL UNCERTAINTY - TEST PLANS, TEST LOGS, TEST PROGRAMS, DEFECT TRACKING RECORDS, EMAIL CORRESPONDENCE WITH VENDOR(S) RELATING TO THE PROBLEM.]

The development teams continued to utilize a series of prototypes and experimentation to empirically characterize the behaviour of the system in order to gain further insight into the problem. Subsequently 3 experimental solutions were prototyped. Each of the potential solutions was then implemented and tested. The solution which showed the most improvement was then further refined using another series of experiments.

[EVIDENCE OF SYSTEMATIC EXPERIMENTATION - THE TEST PROTOTYPE PROGRAMS, TEST PLANS, TEST RESULTS, EMAILS, & DEFECT TRACKING ENTRIES.]

[NOTE: IDEALLY, THE RESEARCHER WOULD COMPARE RESULTS TO INITIAL EXPECTATIONS AND TRY TO EXPLAIN ANY VARIANCES. ]

#### Results:

- Throughput: 30 events/s (152% of objective)  
Continual consumer testing is a time and cost consuming venture in the development of new products in which several variations are tested before a "winner" is agreed upon. Descriptive testing can offer a cheaper solution but provides no information about consumer acceptance of the product. Partial Least Squares regression models provide

**Project Name:** Integrate SQL dbase, x-n, & internet servers (NEW FORMAT)  
**Project Number:** 801

**Start Date:** 2008-01-01  
**Completion Date:** 2008-12-31

---

a statistical method to relate descriptive profiles collected from a trained panel to consumer acceptance data collected from a large consumer group, thus providing an avenue to shorten the time and reduce the cost of product development.

**Conclusion:**

The final solution resulted in the utilization of a combination of a series of unorthodox connection pooling and directed record locking techniques.

[EVIDENCE - OF THE TECHNOLOGICAL ADVANCE - FINAL SOURCE CODE, TEST PROGRAMS, TEST RESULTS, EMAILS TO/FROM VENDOR.]

Key variables resolved: concurrency, CPU usage, Footprint, Memory usage/thread handling, throughput

**Technical Documents:**

- Server Interaction  
software examples core\_issues summary.xls -- 26112 bytes
- Server Interaction  
software examples core\_issues summary.xls -- 26112 bytes

***Uncertainty #2: the application of generalized linear models to data***

Just because a model predicts a decrease in performance as you assume more sensory dimensions are used to conduct a sensory task, can you use this as proof that it explains or predicts Gridgemen's Paradox. I think not. For example, consider your own experience and technique in conducting a difference test. Is performance more tied to the dimensionality of the sensory experience or to the ability to correctly identify a sensory difference resulting from some physical or chemical difference in the samples?

The most significant underlying key variables are:  
concurrency (unresolved), throughput (unresolved), footprint (unresolved), CPU Usage (unresolved), Memory usage/thread handling (unresolved)

**There are no Activities associated with this uncertainty.**

**Project Name:** Database methodology  
**Project Number:** 802

**Start Date:** 2008-01-01  
**Completion Date:** 2009-06-30

## **802 - Database methodology:**

### **Scientific or Technological Objectives:**

<b>M e a s u r e m e n t</b>	<b>C u r r e n t P e r f o r m a n c e</b>	<b>O b j e c t i v e</b>
Access speed with large database (s)	30	15

[NOTE: THIS PROJECT DESCRIPTION IS BASED ON THE CRA'S EXAMPLE OF AN ELIGIBLE PROJECT FROM THEIR SR&ED SOFTWARE DEVELOPMENT INDUSTRY GUIDELINES: INFORMATION CIRCULAR 97-1.]

The objective is to develop and implement a new data basing method in order to double the speed of the database currently achieved in Version 3.5 of our "property record management system."

### **Technology or Knowledge Base Level:**

Benchmarking methods & sources for citations:

- Internet searches: 21 sites / articles -- No results related to our query
- Patent searches: 14 patents -- Searched Google patents
- Similar prior in-house technologies: 1 products / processes -- Existing system has excessive access times (>30 seconds) with large databases (>1 gigabyte).

XYZ Co. has developed a proprietary DMS (database management system) as part of their PRMS (property record management system) product. The DMS works well with small data sets, but has excessive access times (>30 seconds) with large databases (>1 gigabyte).

[NOTE: THIS EXPLANATION OF STANDARD PRACTICE SHOULD ATTEMPT TO OUTLINE "READILY AVAILABLE INFORMATION" ON THE TOPIC CONSIDERED AND IDENTIFY THE BOUNDARIES OF "KNOWN" AND "UNKNOWN" VARIABLES. THESE IN TURN FORM THE BASIS OF THE "TECHNICAL UNCERTAINTIES". THIS INFORMATION IS USEFUL IN HELPING THE AUDITOR TO EVALUATE THE COMPANY'S "TECHNICAL QUALIFICATIONS" WITH RESPECT TO THE TECHNOLOGIES IN QUESTION.]

### **Field of Science/Technology:**

Software (1.02.03)

### **Intended Results:**

- Develop new materials, devices, or products
- Improve existing materials, devices, or products

### **Work locations:**

Analysis, Commercial Facility

### **Scientific or Technological Advancement:**

#### **Uncertainty #1: Relational Data Model Analysis - [Supporting Act.]**

What kind of negative effects might result from using a relational data model with the DMS?

The most significant underlying key variables are:  
performance

#### **A c t i v i t y # 1 - 1 : L i t e r a t u r e R e v i e w**

##### **Work performed in Fiscal Year 2008:**

##### **Methods of experimentation:**

- Analysis / simulation: 4 alternatives - Conducted a literature review of relational data models. As a result we looked at 4 alternate data models.

[NOTE: IDEALLY, CLAIMANTS WOULD PROVIDE SPECIFIC DETAILS AS TO HOW THESE MODELS DIFFERED AND SOME OF THE MOST SIGNIFICANT VARIABLES EXAMINED. IN ADDITION TO A BRIEF OVERVIEW OF

**Project Name:** Database methodology  
**Project Number:** 802

**Start Date:** 2008-01-01  
**Completion Date:** 2009-06-30

---

THE WORK PERFORMED EACH ACTIVITY SHOULD ATTEMPT TO CROSS-REFERENCE RELEVANT, TECHNICAL DOCUMENTATION INCLUDING: DOCUMENT NAME, DATE, # OF PAGES AND LOCATION.]

**Results:**

[NOTE: IF THERE WERE ANY TEST RESULTS FROM THIS ACTIVITY THAN THESE SHOULD BE STATED HERE]

Extremely large data sets are usually quite complex, frequently containing scores of variables, many of which can only be described by non-linear relationships. Numerous variables may also interact with each other. These issues all combine to make many statistical procedures, such as Analysis of Variance or regression analysis, difficult to use. Care must also be taken such that data with many variables is not "over analyzed." Not matter how large the data set is originally, if it is cut into enough segments, significant differences will be found between groups simply by chance.

**Conclusion:**

Discovered that relational data models could be inefficient when used in the DMS in some circumstances.

[NOTE: THE IDEAL CONCLUSION WOULD ALSO BRIEFLY DETAIL HOW THESE RESULTS COMPARED WITH INITIAL EXPECTATIONS AND OUTLINE ANY FURTHER CONCLUSIONS WHICH COULD AFFECT FUTURE DEVELOPMENTS OF THIS NATURE.]

Key variables resolved: performance

---

***Uncertainty #2: Comm. model vs. Relational Environment***

How will using a data model designed for data communications in a relational environment affect performance?

The most significant underlying key variables are:  
performance

---

**Activity # 2 - 1 : Data Communications Model Analysis**

**Work performed in Fiscal Year 2008:**

**Methods of experimentation:**

- Process trials: 1 runs / samples - We experimented to determine if an existing data communications model could be adapted to achieve processing efficiencies, at the expense of additional storage space.

**Results:**

[NOTE: SIMILARLY, IF THERE WERE ANY TEST RESULTS FROM THIS ACTIVITY THAN THESE SHOULD BE STATED HERE]

While large data sets introduce additional complications to their analysis, researchers should not disregard the basic statistical concepts that have served so well when analyzing smaller data sets. Data collection methods should reflect overall objectives and initial analysis should be composed of EDA and data visualization techniques. Once a complete understanding of the data has been gained more complicated methods, such as cluster analysis or data base sampling, can be attempted.

**Conclusion:**

Determined that a data communications model could achieve processing efficiencies.

[NOTE: IDEALLY WE WOULD OUTLINE ADDITIONAL DETAILS SUCH AS "PROS AND CONS" DISCOVERED WITH RESPECT TO THIS METHOD - PARTICULARLY THOSE THAT WERE OTHERWISE UNEXPECTED.]

This conclusion however uncovered new uncertainty with respect to the optimal method to combine relational and packet access methods.

[NOTE: THIS UNCERTAINTY AND RELATED ACTIVITIES ARE THEN SEPARATELY DESCRIBED.]

Key variables resolved: performance

---

***Uncertainty #3: Relational Access + Packet Access Combination***

How can we optimally combine relational and packet access against the same database to yield a minimum # of inefficiencies when processing tables in the DMS?

The most significant underlying key variables are:  
performance

---

**Activity # 3 - 1 : Model Comparison Tests**

**Work performed in Fiscal Year 2008:**

**Project Name:** Database methodology  
**Project Number:** 802

**Start Date:** 2008-01-01  
**Completion Date:** 2009-06-30

---

**Methods of experimentation:**

- Process trials: 7 runs / samples - Conducted 7 comprehensive benchmark tests to compare performance between the two models.

[NOTE: IDEALLY CLAIMANTS WOULD PROVIDE SPECIFIC DETAILS AS TO HOW THESE DATABASES DIFFERED AND WHY THIS WAS BELIEVED TO BE TECHNICALLY SIGNIFICANT. WE SHOULD ALSO ATTEMPT TO SUMMARIZE SOME OF THE MOST SIGNIFICANT VARIABLES EXAMINED.]

Most statistical software packages are completely dependent upon the user regarding the selection and implementation of the correct model to analyze. Whatever model the user defines, the computer will "fit" that model. Incorrect conclusions can be drawn when inappropriate models are fit to the data. The difficulty is that the computer program cannot inform you when the model is inappropriate

**Conclusion:**

While some of the tables could be processed more efficiently if they were in packet form, others were best managed through relational techniques.

[NOTE: THE IDEAL CONCLUSION WOULD COMPARE RESULTS WITH INITIAL EXPECTATIONS AND TRY TO PROVIDE TECHNICAL EXPLANATIONS FOR THESE DIFFERENCES.]

Key variables resolved: performance

---

**A c t i v i t y # 3 - 2 : H y b r i d M o d e l A t t e m p t**

---

**Work performed in Fiscal Year 2009:**

**Methods of experimentation:**

- Physical prototypes: 1 samples - Experimentally employed a hybrid approach involving both relational and packet data management techniques. Created a prototype Data Model DMS with the intention of making it faster than existing one.

[NOTE: AN IDEAL DESCRIPTION WOULD OUTLINE SOME OF THE MAJOR TECHNICAL ALTERNATIVES CONTEMPLATED AND RELATED ASSUMPTIONS MADE.]

Initial testing indicated that the new DMS was 75% faster than the existing DMS through use of the newly developed hybrid data access techniques.

**Results:**

- Access speed with large database: 10 s (133% of objective)  
Consequently, recheck the appropriateness of your model when faced with non-sensical analysis results from your computer package.

**Conclusion:**

Processing time for query and update capabilities were improved to <10 seconds for >1 GB databases.

[NOTE: IDEALLY CLAIMANTS WOULD PROVIDE SPECIFIC CONCLUSIONS TO EXPLAIN RESULTS HAVE RELATIONAL AND PACKET DATA MANAGEMENT COMBINATION VS. INITIAL EXPECTATIONS. I.E. WHY DID SOME SETS WORK BETTER THAN OTHERS?]

Key variables resolved: performance

**Project Name:** Network failure problems  
**Project Number:** 803

**Start Date:** 2008-01-01  
**Completion Date:** 2008-12-31

---

## **803 - Network failure problems:**

### **Scientific or Technological Objectives:**

<b>M e a s u r e m e n t</b>	<b>C u r r e n t P e r f o r m a n c e</b>	<b>O b j e c t i v e</b>
Concurrent accesses ()	760	1750

[THIS PROJECT IS BASED ON THE CRA'S EXAMPLE #3 FROM, "CROSS-SECTOR SHOP FLOOR GUIDANCE DOCUMENT" (JULY 29, 2002)]

The objective was to determine why the CallHome high-speed network does not meet the original design criteria, and to take correction such that the network will facilitate 500 high-speed access ports, with 1750 concurrent transactions, at a maximum 25% reduction in response time.

CallHome's technological advancement sought was the solution to the network failure. The network had been designed according to the current theory, and had failed to provide the theoretical performance. Once normal network troubleshooting proved ineffective, and did not solve the problems, CallHome realized more design work was required to determine the underlying problem with the technology. Solving this problem represented a technological advancement.

### **Technology or Knowledge Base Level:**

Benchmarking methods & sources for citations:

- Internet searches: 12 sites / articles -- Nothing matched our query
- Patent searches: 4 patents -- Searched Google patents, we looked at 4 similar products that didn't meet our specs
- Similar prior in-house technologies: 1 products / processes -- Theoretical capacity was 1750 concurrent transactions, but system was crashing in practice.

A communications company, CallHome, designed and built a high-speed Internet access network to offer its clients. Three different vendors' equipment were involved in the implementation of the network, including the local telephone company equipment. The initial design was capable of 500 high-speed ports. Maximum theoretical network capacity was established as 1750 concurrent transactions, with a maximum 25% reduction in response time. After reviewing the overall network design, all the equipment vendors agreed that their equipment could operate in the target network architecture, and these numbers could be comfortably achieved.

After six months of operation, CallHome had sold 225 high-speed accesses, and the network management system was reporting utilization numbers of 700-800 concurrent accesses, with an 18% reduction in response rate. Although this reduction in response time raised concerns about network capacity, there were no customer complaints, and the vendors continued to stand by their position. CallHome then had a major marketing campaign that resulted in another 60 high-speed customers.

After their service was activated, the network began crashing for no apparent reason. The network management software could not pinpoint the problem, and the equipment vendors could offer no reasons for the failures. Customers began canceling their service.

### **Field of Science/Technology:**

Software (1.02.03)

### **Intended Results:**

- Improve existing materials, devices, or products

### **Work locations:**

Analysis, Commercial Facility

### **Scientific or Technological Advancement:**

**U n c e r t a i n t y # 1 : S y s t e m U n c e r t a i n t y**

[NOTE: WHICH VARIABLES ARE UNPREDICTABLE WITH RESPECT TO DETERMINING THE "OPTIMAL COMBINATION OF COMPONENTS"? THESE ISSUES ARE CREATED BY DEPARTURES FROM STANDARD PRACTICES.]

**Project Name:** Network failure problems  
**Project Number:** 803

**Start Date:** 2008-01-01  
**Completion Date:** 2008-12-31

-What method should be used to determine the cause of the network failure, given the network management software, the vendors, and all other network indicators are normal?  
-The vendors would not provide detailed information on their firmware source code. Their equipment complies with network and protocol interface standards. Can we, and how do we, develop a test bed that will provide critical technical information necessary to identify the technical problems?  
-Once the inconsistencies in networking address index caching were identified, it was technically uncertain how we could develop an interface that will compensate for the different index caching techniques, without compromising on response times.

The extent of system uncertainty of possible resolutions to problems is unknown. For instance, a possible solution may solve one component but may also cause the other components to fail. Therefore, it is possible that a solution cannot be developed to address the needs of all the components. For example, a software solution of Vendor A may not correctly interface with a specific piece of firmware from Vendor B and C.

The most significant underlying key variables are:  
determining cause of network failure, integration of components, maintaining response time

## **A c t i v i t y # 1 - 1 : D e v e l o p m e n t a n d T e s t i n g**

### **Work performed in Fiscal Year 2008:**

#### **Methods of experimentation:**

- Analysis / simulation: 5 alternatives - 4 architectural designs experimentally evaluated. All designs failed, which resulted in a completely new approach that involved developing new techniques for disparate data normalization.
- Process trials: 32 runs / samples - 20 tests on network management system, 7 tools developed to look at equipment interaction, 3 units from each of 2 vendors tested.
- Physical prototypes: 1 samples - We built a prototype interface unit and evaluated it in our test bed. After extensive testing, we realized that a slight change in the new normalization process order would improve performance by 45%.

[AUTHOR'S NOTE: THE DESCRIPTIONS BELOW WERE PROVIDED IN THE CRA'S EXAMPLE. THE DATA ABOVE (# TRIALS/ALTERNATIVES) IS PROVIDED TO ILLUSTRATE SOME OF THE ADDITIONAL DETAILS THAT WOULD IDEALLY BE INCLUDED.]

[NOTE: TRY TO CLARIFY VARIABLES IN QUESTION AND ILLUSTRATE ANY UNEXPECTED INTER-RELATIONS.]

Engineers and other technical staff studied and analyzed the problem. Our preliminary conclusions led us to speculate that the problem happens when the user community demands 760 concurrent transactions.

Conducted a number [HOW MANY?] of different experiments on the network system management raw data. From the analysis we determined that the network management system was not properly reporting on the interface conditions between vendor A and vendor B equipment.

Developed and built a number [HOW MANY?] of different network management tools to look at how the equipment from vendor A and B were interacting with each other. Analysis of implementing the tool revealed that the two vendors were using slightly different network address caching methods.

Three different units from each vendor were tested, all with the same results of a crashed network in our test bed simulations.

Both vendors refused us access to their firmware source code, and reported there was no problem with their equipment. They stated that their equipment conformed to OSI network and protocol interface standards, so that access to the source code was not necessary.

We began to research different techniques for integrating the different vendor equipment without the performance reduction we were experiencing in our current network.

Four different architectural designs were experimentally evaluated based on available knowledge coupled with our own experience. All designs failed, which resulted in a completely new approach that involved developing new techniques for disparate data normalization.

We built a prototype interface unit and evaluated it in our test bed. After extensive testing, we realized that a slight change in the new normalization process order would improve performance by 45%.

**Project Name:** Network failure problems  
**Project Number:** 803

**Start Date:** 2008-01-01  
**Completion Date:** 2008-12-31

---

Simulation load testing indicated that overall network performance with 500 ports should allow 1900 concurrent transactions with a reduction of 18% in performance. This provides almost a 9% improvement over the original network design.

The hardware solution was then developed and implemented over the entire network to determine if the system uncertainty had been resolved. The implementation was successful.

**Results:**

- Concurrent accesses: 1900 (115% of objective)

A final model can then be chosen that includes only the statistically important descriptive attributes. This model is much simpler than a model including most if not all of the original variables and when used for prediction the simpler model typically shows a decrease in prediction error. Further, a simpler model makes the identification and creation of an "ideal" product much more direct as fewer variables need to be examined for their effect on consumer acceptance. Development costs are also lowered as descriptive panels only need to collect information on a handful of attributes for use in predicting consumer acceptance of these new products.

**Conclusion:**

CallHome learned that the different vendor's equipment used different techniques for indexing and routing of network addresses. This was achieved after extensive experimentation in the network test-bed and the development of new technically advanced network management analysis tools.

CallHome developed a new set of integration firmware that not only resolved the original firmware disparities, it introduced new techniques for networking index caching that theoretically pushed the CallHome network capability to 1900 concurrent access requests, with an 18% reduction in response time. Also, the company attempted to develop system integration techniques, which unfortunately failed because they did not improve reliability.

[NOTE: IDEALLY, WE NEED COMPARISONS TO INITIAL EXPECTATIONS & TO CONCLUDE ON THE UNCERTAINTIES STATED: I.E. WHAT IF ANY ARCHITECTURES WOULD ACCOMPLISH THE COMPRESSION OBJECTIVES AND WHY? IT IS THE HYPOTHESES OR CONCLUSIONS TO EXPLAIN THESE RESULTS, RATHER THAN THE RESULTS THEMSELVES, WHICH THE CCRA WISHES TO SEE EVIDENCE OF.]

[CRA - RATIONALE FOR ELIGIBILITY: DID THE PROJECT EMBODY SYSTEM UNCERTAINTY? SEVERAL VENDORS WERE INVOLVED TO SUPPLY THE CLAIMANT WITH DIFFERENT DEVICES TO ASSIST IN FINDING A SOLUTION TO THEIR NETWORK FAILURES. FROM THE PROJECT DESCRIPTION, THE TECHNICAL REVIEWER THOUGHT SOME OF THE ATTEMPTED SOLUTIONS COULD HAVE BEEN SUPPLIER TRIALS, WHICH MAY NOT BE SR&ED. HOWEVER, THE CLAIMANT WAS ABLE TO SHOW THAT EACH TIME A DEVICE WAS TRIED THE CLAIMANT HAD TO MODIFY THEIR SYSTEM TO ACCOMMODATE THE DIFFERENT DEVICE. THE CLAIMANT THEN EVALUATED AND ATTEMPTED TO UNDERSTAND WHY THE DEVICE FAILED AND PROVIDED FEEDBACK TO THE VENDORS TO ENHANCE OR RECTIFY THE DEVICE FORCING A CHANGE TO THE UNDERLYING TECHNOLOGY. HEREIN LIES SYSTEM UNCERTAINTY BECAUSE IT WAS UNKNOWN WHAT EFFECT DEVICE A WOULD IMPOSE ON DEVICE B AND HENCE THE BEHAVIOUR OF THE OVERALL SYSTEM. ALSO, IT WAS UNKNOWN WHAT ALTERATIONS TO THE CLAIMANT'S SYSTEMS WERE REQUIRED. QUITE OFTEN SYSTEM UNCERTAINTY IS PRESENT WHEN WORKING WITH VENDORS TO SOLVE PROBLEMS BECAUSE OF THE UNEXPLORED NATURE OF DEVICES SUPPLIED FROM DIFFERENT SOURCES. ALSO ELIGIBLE WERE THE SUBCONTRACTED COSTS TO INSTALL AND REMOVE THE DEVICES FOR EACH TRIAL AS THIS WAS ALSO ENGINEERING WORK IN DIRECT SUPPORT OF THE SR&ED PROJECT.]

Key variables resolved: determining cause of network failure, integration of components, maintaining response time

**Project Name:** Compression algorithm  
**Project Number:** 901

**Start Date:** 2009-01-01  
**Completion Date:** 2009-04-30

## **901 - Compression algorithm:**

### **Scientific or Technological Objectives:**

<b>M e a s u r e m e n t</b>	<b>C u r r e n t P e r f o r m a n c e</b>	<b>O b j e c t i v e</b>
Compression of 1 Mb map (K)	90	30

[THIS EXAMPLE IS REPRODUCED FROM "GUIDANCE ON ELIGIBILITY OF SOFTWARE PROJECTS FOR THE SR&ED TAX CREDITS," AS PUBLISHED BY THE CRA IN CO-OPERATION WITH CATA & THE SOFTWARE INDUSTRY.]

The objective was to develop a new compression tool for GIS information with the capability of compressing a 1 meg map down to 30K. This has to be accomplished with less than 2% data loss.

Our product is very similar to that of ABC-IT Inc, however due to limited memory and battery life only limited size maps could be loaded, and a limited number of notes could be attached. Our competitor has released their software with a 50% further decrease in their compressed maps, i.e. they can compress a 1 meg. map to less than 40K. Our current best compression is to get a 1 meg image down to 90K. In order to maintain market share we must at least meet their performance, and develop a new compression technique. Our eventual goal is to be down to 30K.

### **Technology or Knowledge Base Level:**

Benchmarking methods & sources for citations:

- Internet searches: 5 sites / articles -- Found 5 website that had 13 articles, nothing matched our criteria
- Competitive products or processes: 1 products -- Competitor can compress 1Mb map to <40K.
- Similar prior in-house technologies: 1 products / processes -- Our current product can compress 1Mb to 90K.

In September of last year our competitor ABC-IT Inc. released a new tool suite for the compression and modification of electronic maps and overlays. The tools are designed for use on small platforms, (PDA's, Palm Pilots, Palmtops) They allow the user to make notes, and modify the electronic maps as they are doing field work. They can then upload the changes to their desktop PC, back at the office, where a full GIS package resides.

### **Field of Science/Technology:**

Software (1.02.03)

### **Intended Results:**

- Develop new materials, devices, or products

### **Work locations:**

Commercial Facility

### **Scientific or Technological Advancement:**

#### **U n c e r t a i n t y # 1 : O p t i m a l c o m p r e s s i o n m e t h o d**

Optimal compression method(s). Specifically, what, if any, architectures would accomplish this compression objective?

The most significant underlying key variables are:  
compression

#### **A c t i v i t y # 1 - 1 : D e v e l o p m e n t a n d T e s t i n g**

**Work performed in Fiscal Year 2009:**

##### **Methods of experimentation:**

- Physical prototypes: 4 samples - The main issues were obtaining sufficient compression and allowing separation of the map and overlays.

[AUTHOR'S NOTE: THE DESCRIPTIONS BELOW WERE PROVIDED IN THE CRA'S EXAMPLE. THE DATA ABOVE (# TRIALS/ALTERNATIVES) IS PROVIDED TO ILLUSTRATE SOME OF THE ADDITIONAL DETAILS THAT WOULD IDEALLY BE INCLUDED.]

**Project Name:** Compression algorithm  
**Project Number:** 901

**Start Date:** 2009-01-01  
**Completion Date:** 2009-04-30

---

Through development and experimentation with several approaches [AUTHOR'S NOTE: IDEALLY THE DESCRIPTION WOULD DETAIL HOW MANY PROTOTYPE VARIATIONS DID WE ATTEMPT? I.E. 5, 50, 500? WERE THEY ALL SIMILAR OR COMPLETELY DIFFERENT? IF DIFFERENT, HOW SO AND WHY?], we managed to develop a compression tool using a data communication standard (X2 standard for hardware compression), and a method of analyzing the maps and overlays, synchronizing them into a single image and then using a modified version of MPEG 3 compression.

The modified software compression allows for easier separation of the map from the overlay once the data is transferred from the hand held unit to the desktop PC.

[NOTE: THIS DESCRIPTION IS STILL FAIRLY WEAK IN THAT THE "ACTIVITIES" & "CONCLUSIONS" ARE CURRENTLY BASED TOO HEAVILY ON A "GOALS - RESULTS" ORIENTATION RATHER THAN ILLUSTRATING WHY IT WAS SO HARD TO GET TO THE FINAL SOLUTION AND THEN IDENTIFYING RELEVANT "TECHNICAL CONCLUSIONS."]

**Results:**

- Compression of 1 Mb map: 30 K (100% of objective)

The use of PLS models eliminates drawback of both Ordinary Least Squares (OLS) Regression and Principal Component Regression (PCR). OLS requires more samples (products) than variables to be included in the model, which is typically not the case when attempting to compare consumer and descriptive data. In fact, it is not unusual to present only 8-10 products to the consumer and descriptive panels yet collect information on a couple dozen descriptive attributes. PCR eliminates this difficulty, as well as the problem of multicollinearity that is present in much of the descriptive data. However, the first principal component, formed from the descriptive data, is not necessarily related to consumer acceptance thereby weakening the model for predictive purposes. The use of PLS through Preference Cluster Mapping eliminates both of these difficulties. Further, because it still fits a model to the data it remains possible to predict consumer liking of new products from the existing model by simply running additional descriptive panels.

**Conclusion:**

ELIGIBLE AS WRITTEN:

According to the CRA, "Generally, this Advancement WOULD QUALIFY, BUT it would NOT qualify in either of the following two situations."

[INELIGIBLE IF:]

1. While doing the preliminary technical feasibility work we discovered a company in the US that has a tool ideally suited for our needs. We are currently working out a licensing agreement for resale. With a couple of parameter changes, their tool will give us our target compression.

[RATIONALE: A ROUTINE SOLUTION WAS FOUND AND IMPLEMENTED WITHOUT SYSTEM UNCERTAINTY WITH RESPECT TO THE OPTIMAL METHOD(S) OF INTEGRATION.]

OR,

2. In the early part of the technical feasibility study portion of the project, we learned that one of the senior software engineers had resigned from ABC-IT Inc. We hired him and he is redeveloping their algorithm for our application. We have decided that matching the ABC-IT Inc. performance will be adequate.

[RATIONALE: THOUGH THE DEVELOPMENT MAY HAVE BEEN ELIGIBLE FOR THE COMPANY, HIRING THE NEW EMPLOYEE IMMEDIATELY EXPANDED ITS "STANDARD PRACTICE KNOWLEDGE BASE" TO INCLUDE THIS EMPLOYEE'S KNOWLEDGE. SINCE THE SOLUTION COULD BE DEVELOPED BY HIM WITHOUT ANY FURTHER "TECHNOLOGICAL UNCERTAINTY" IT DOES NOT REPRESENT AN ELIGIBLE EXPERIMENTAL DEVELOPMENT ACTIVITY.]

[NOTE: IDEALLY, WE NEED COMPARISONS TO INITIAL EXPECTATIONS & TO CONCLUDE ON THE UNCERTAINTIES STATED: I.E. WHAT IF ANY ARCHITECTURES WOULD ACCOMPLISH THE COMPRESSION OBJECTIVES AND WHY? IT IS THE HYPOTHESES OR CONCLUSIONS TO EXPLAIN THESE RESULTS, RATHER THAN THE RESULTS THEMSELVES, WHICH THE CRA WISHES TO SEE EVIDENCE OF.]

Key variables resolved: compression

## **902 - Software Data Warehouse Description Development:**

### **Scientific or Technological Objectives:**

<b>M e a s u r e m e n t</b>	<b>C u r r e n t P e r f o r m a n c e</b>	<b>O b j e c t i v e</b>
CPU Utilization (% busy)	95	70
Response Time (seconds)	60	15
Data to compression (:1 ratio)	5	15

### **Technology or Knowledge Base Level:**

Benchmarking methods & sources for citations:

- Internet searches: 33 sites / articles -- 33 sites & 14 resulting articles reviewed
- Competitive products or processes: 6 products -- no methods to characterize non-uniform, dynamic data
- Similar prior in-house technologies: 2 products / processes -- -- benchmarks for CPU utilization, Res
- Potential components: 100 products -- over 100 potential components reviewed (open source & proprietary)
- Queries to experts: 3 responses -- Queries to experts: 3 responses -- no methods to characterize non-uniform, dynamic data of this envi

### **Field of Science/Technology:**

Computer hardware and architecture (2.02.08)

### **Intended Results:**

- Improve existing processes

### **Work locations:**

Analysis, Commercial Facility

### **Scientific or Technological Advancement:**

#### **Uncertainty #1: Uncertainty #1: : Non-uniform dataset determination**

We are uncertain as to how and whether it is possible to develop a method to identify and exploit the unique properties of non-uniform data sets. We are also uncertain whether we can use compressed data blocks vs. entire tables to traverse the database and how much of a performance improvement this will result in.

The most significant underlying key variables are:

Methods to characterize non-uniform data, Optimal use of compression dictionary, Definition and construction of data blocks, CPU utilization

#### **A c t i v i t y # 1 - 1 : D e v e l o p g e n e r i c d a t a m o d e l**

##### **Work performed in Fiscal Year 2008:**

##### **Methods of experimentation:**

- Analysis / simulation: 10 alternatives - - identified most common frequency values & evaluated use of column value frequencies to create prototype compression dictionary - using a relational dbase environment

One of the drawbacks to Preference Cluster Mapping has been the lack of statistical testing to determine the most parsimonious (simplest) model. Rather, chosen models have tended to include most, if not all of the descriptive attributes and inference has been restricted to the graphical inspection of loading and score plots. From these plots "drivers" of liking, both positive and negative, are identified based on the observed proximity of these descriptive attributes to consumer liking. This often leads to a large number of descriptive attributes that appear to be driving consumer liking and thus makes the identification of an "ideal" product much more complicated both to identify and to create. Nevertheless, the use of Partial Least Square modeling has allowed such "ideal" models to be identified by focusing on the positive and negative drivers and adjusting the descriptive levels of these attributes until such an "ideal" product is identified.

**Project Name:** Software Data Warehouse Description Development  
**Project Number:** 902

**Start Date:** 2008-06-01  
**Completion Date:** 2009-03-31

---

**Results:**

No results have been recorded for this Activity.

At the end of this first phase we found that a reasonably accurate data set model could be created. This was further tested and the data set model accuracy was verified and validated against several concrete smaller-sized relational databases available to us in the data warehouse.

**Conclusion:**

Model proved feasible - developed table-wide list of most frequent values for compression dictionary

**Technical Documents:**

- CRA Software Guidelines

---

**A c t i v i t y # 1 - 2 : D e v e l o p c o m p r e s s i o n m e t h o d s**

---

**Work performed in Fiscal Year 2008:**

**Methods of experimentation:**

- Physical prototypes: 10 samples - Developed test scripts to compared CPU utilization, integrity and data throughput for operations including: parallel load, delete/update operations, full table scan & access by row.  
However, this model may result in the Product or City term having 0 degrees of freedom in the output given by the computer program. Again, an "error message." While Product and City are crossed effects and the interaction is correct, panelists are nested within city (different panelists in different cities). Thus, the correct model is: Source of Variation  
City  
Consequently, recheck the appropriateness of your model when faced with non-sensical analysis results from your computer package.

**Results:**

No results have been recorded for this Activity.

**Conclusion:**

We determined it is best to restrict query/refresh options to compressed blocks vs. entire tables  
Key variables resolved: CPU utilization, Definition and construction of data blocks

---

**A c t i v i t y # 1 - 3 : C o m p r e s s i o n a l g o r i t h m w i t h d y n a m i c t e c h n i q u e s**

---

**Work performed in Fiscal Year 2008:**

**Methods of experimentation:**

- Physical prototypes: 3 samples - Examined use of buffer cache to organize & control compression dictionaries when calls made to uncompress multiple blocks
- ... prototype revisions: 12 revisions - 2 of the 3 sample compression algorithms selected for further experimentation to include 12 different dynamic compression techniques for dataset changes. Each of these had the data integrity verified

**Results:**

CPU Utilization: 66 % busy (116% of objective)  
Response Time: 22 seconds (84% of objective)  
Data to Compression: 131 : 1 ratio (1260% of objective)  
In August 2008, a final prototype was selected for widespread commercial implementation ending this aspect of the experimental development.

**Conclusion:**

This development lead to the discovery that we could use the column value frequency of initial tables rows to create an effective block-based compression dictionary.

**Technical Documents:**

- x

---

**A c t i v i t y # 1 - 4 : E x t e n d d a t a c o m p r e s s i o n m e t h o d s**

---

**Work performed in Fiscal Year 2008:**

**Methods of experimentation:**

- Process trials: 102 runs / samples - Used external consultant - exploration into use of the implemented compression prototype for data backup and recovery operations

**Project Name:** Software Data Warehouse Description Development  
**Project Number:** 902

**Start Date:** 2008-06-01  
**Completion Date:** 2009-03-31

---

**Results:**

No results have been recorded for this Activity.  
As the result of this work it was found out and further documented that the prototype provided measurable performance improvements [QUANTIFY] when applied to very large databases in excess of 2.5 million rows (1.3 GB) such as those typically encountered in data warehouses.

**Conclusion:**

Success attributed primarily to compression dictionary vs. data blocks

Key variables resolved: CPU utilization, Definition and construction of data blocks, Optimal use of compression dictionary

**Activity #1-5: Correlate compression block size with initial data set**

---

**Work performed in Fiscal Year 2008:**

**Methods of experimentation:**

- Analysis / simulation: 22 alternatives - the implemented prototype was used to determine whether or not an optimal data table compression-block size could be determined by both the initial data set analysis and the dynamic

**Results:**

No results have been recorded for this Activity.

**Conclusion:**

Could NOT correlate compression block size w initial data set & dynamic analysis

Key variables resolved: CPU utilization, Definition and construction of data blocks, Methods to characterize non-uniform data

***Uncertainty #2: generalized linear models take count data from paired comparisons what is the outcome?***

---

The most significant underlying key variables are:  
linear models (unresolved), paired comparisons (unresolved)

**There are no Activities associated with this uncertainty.**

**Project Name:** Scaling vs. speed vs. compression  
**Project Number:** 1001

**Start Date:** 2010-01-01  
**Completion Date:** 2010-12-31

## **1001 - Scaling vs. speed vs. compression:**

### **Scientific or Technological Objectives:**

<b>M e a s u r e m e n t</b>	<b>C u r r e n t P e r f o r m a n c e</b>	<b>O b j e c t i v e</b>
Encoding rate (s/G)	20	5

[THIS EXAMPLE IS REPRODUCED FROM "GUIDANCE ON ELIGIBILITY OF SOFTWARE PROJECTS FOR THE SR&ED TAX CREDITS," AS PUBLISHED BY THE CRA IN CO-OPERATION WITH CATA & THE SOFTWARE INDUSTRY.]

[AUTHOR'S NOTE: IDEALLY THE TAXPAYER WOULD ATTEMPT TO QUANTIFY THE OBJECTIVES THEY ARE TRYING TO ACHIEVE. A QUANTIFIABLE OBJECTIVE HAS BEEN ADDED ABOVE, TO ILLUSTRATE.]

We seek to show through analysis that the key to both graceful scaling to higher speed platforms and speed maximization for a specified compression on a Pentium performing "Framis" coding is the optimal use of the Pentium look ahead.

We also want to analytically determine the compression expected as a function of the number of bytes processed in parallel and to thereby determine the speed/compression trade-off.

### **Technology or Knowledge Base Level:**

Benchmarking methods & sources for citations:

- Competitive products or processes: 1 products -- Competitor capable of encoding a gigabyte in seven seconds on a 400 Mhz Pentium III processor.
- Potential components: 7 products -- There are seven potential components we are looking at using

In September of last year our competitor, MedsInc, announced its Framis encoding software capable of encoding a gigabyte in seven seconds on a 400 Mhz Pentium III processor.

[NOTE: AN IDEAL DESCRIPTION WOULD ALSO OUTLINE WHAT METHODS WERE "PUBLIC" VS. "PROPRIETARY INFORMATION" SINCE THE RESEARCHER IS EXPECTED TO TAKE NOTE OF "READILY AVAILABLE INFORMATION" BUT IS NOT EXPECTED TO KNOW "INFORMATION PROPRIETARY TO A COMPETITOR OR KNOWN IN ONLY SPECIALIST OR ACADEMIC CIRCLES."]

### **Field of Science/Technology:**

Software (1.02.03)

### **Intended Results:**

- Develop new processes

### **Work locations:**

Commercial Facility

### **Scientific or Technological Advancement:**

#### **U n c e r t a i n t y # 1 : D e t e r m i n i n g o p t i m a l t r a d e - o f f**

The challenge is analyzing and determining the optimal trade-off between scaling, speed and compression on several computing platforms.

[NOTE: IDEALLY WE WOULD CLARIFY WHICH VARIABLES ARE UNPREDICTABLE WITH RESPECT TO DETERMINING THE "OPTIMAL COMBINATION OF COMPONENTS"? THESE ISSUES ARE CREATED BY DEPARTURES FROM STANDARD PRACTICES.]

The most significant underlying key variables are:  
scaling, speed, compression, computing platform

#### **A c t i v i t y # 1 - 1 : E n c o d i n g a l g o r i t h m**

**Work performed in Fiscal Year 2010:**

**Project Name:** Scaling vs. speed vs. compression  
**Project Number:** 1001

**Start Date:** 2010-01-01  
**Completion Date:** 2010-12-31

---

**Methods of experimentation:**

- Process trials: 52 runs / samples - 8 tests on each of 4 platforms to determine trade-off between scaling, speed & compression, followed by testing of new algorithm by 5 samples on each of the 4 platforms.

[AUTHOR'S NOTE: THE DESCRIPTIONS BELOW WERE PROVIDED IN THE CRA'S EXAMPLE. THE DATA ABOVE (# TRIALS/ALTERNATIVES) IS PROVIDED TO ILLUSTRATE SOME OF THE ADDITIONAL DETAILS THAT WOULD IDEALLY BE INCLUDED.]

During the project the company used a combination of routine methods to analyze the trade-off between scaling, speed and compression on several computing platforms.

[NOTE: THE OPTIMAL DESCRIPTION WOULD DETAIL HOW MANY PROTOTYPE VARIATIONS ATTEMPTED (5?, 50?, 500?) AND SIGNIFICANCE OF ANY DIFFERENCES]

Next we performed testing to determine whether or not the new encoding algorithm consistently met its target of being able to code a gigabyte in five seconds. This involved the testing of performance against several gigabyte samples and the writing of a report that described performance as a function of the properties of the specific data samples. [HOW MANY TESTS PERFORMED?]

[NOTE: TRY TO CLARIFY VARIABLES IN QUESTION AND ILLUSTRATE ANY UNEXPECTED INTER-RELATIONS.]

**Results:**

- Encoding rate: 5 s/G (100% of objective)

If you total up the degrees of freedom you see that there are no df left for the error term! Remove the interaction from the model and the problem is solved. (In some cases, if panelist is properly specified as a random effect the test for product differences will still be performed in the above model. Thus, if no tests are performed also check whether panelist is a fixed or random effect.

**Conclusion:**

[NOTE - THE IDEAL SOLUTION WOULD COMPARE RESULTS TO INITIAL EXPECTATIONS AND TRY TO PROVIDE ADDITIONAL TECHNICAL EXPLANATIONS WITH RESPECT TO THE STATE UNCERTAINTIES: OPTIMAL TRADE-OFF BETWEEN SCALING, SPEED AND COMPRESSION ON SEVERAL COMPUTING PLATFORMS.]

**ELIGIBLE ACTIVITIES CUT-OFF:**

In the CRA's view the company showed that the new Framis had to be interfaced to an existing data input system and that a new graphical user interface had to be developed to determine "whether or not the new Framis would scale to higher speed platforms more gracefully" and therefore the costs of the supporting activities, although routine in themselves, are eligible.

The project ends when the Framis software is sufficiently debugged and sufficiently featured that it codes a gigabyte in less than five seconds on a 400 MHz Pentium III and it can be shown that it scales to at least a couple platforms with clocks in excess of 400 MHz in a manner which is more "graceful" than was possible with the competitor's algorithm.

[NOTE: THIS POINT CAN BE EXTENDED IF THE COMPANY CAN EXPAND THE SCOPE OF THE TECHNICAL OBJECTIVE].

Key variables resolved: compression, computing platform, scaling, speed

**Project Name:** Improve Vista  
**Project Number:** 1002

**Start Date:** 2010-01-01  
**Completion Date:** 2010-08-19

## **1002 - Improve Vista:**

### **Scientific or Technological Objectives:**

<b>M e a s u r e m e n t</b>	<b>C u r r e n t P e r f o r m a n c e</b>	<b>O b j e c t i v e</b>
Avg. Laptop Battery Life (hours)	4.62	5.3
Avg. Load Time (seconds)	37	25
Avg. Time Between Disruptions (crashes) (hours)	17	40

Our objective is to improve our existing Vista operating system in order to overcome some issues discovered during field testing.

### **Technology or Knowledge Base Level:**

Benchmarking methods & sources for citations:

- Internet searches: 5 sites / articles -- Found 5 website that had 13 articles, nothing matched our criteria
- Patent searches: 2 patents -- Searched Google patents
- Competitive products or processes: 2 products -- Mac OSX & Linux - proprietary and/or use a different system architecture
- Similar prior in-house technologies: 6 products / processes -- 6 previous versions of Windows, over 30 years experience developing operating systems

### **Field of Science/Technology:**

Software (1.02.03)

### **Intended Results:**

- Improve existing materials, devices, or products

### **Work locations:**

Analysis

### **Scientific or Technological Advancement:**

#### **Uncertainty #1: Meeting objectives given architecture complexity, 3rd party issues**

- Windows code is very complicated - there are more than 50 dependency layers as well as circular dependencies. (A person working on Windows for 5 years would likely not know more than 2 of them).

- Driver problems - Lots of hardware components (old and new) had no drivers for Vista or, even if they had, their quality was very poor. These problem were the result of a redesign in Windows, which was necessary to improve the system's overall reliability and security. The uncertainty is how best to make the necessary improvements to the system, without creating further driver issues.

- Application compatibility - Lots of applications did not work on Vista because many software developers created applications that function only if the user has full administrative privileges. With the introduction of UAC (User Access Control) and other system changes, lots of old applications have problems. The uncertainty is how best to make the necessary improvements to the system, without creating further application compatibility issues.

#### **A c t i v i t y # 1 - 1 : C o l l e c t i n g a n d a n a l y z i n g u s e r d a t a**

##### **Work performed in Fiscal Year 2010:**

##### **Methods of experimentation:**

- Analysis / simulation: 5000 alternatives - more than 5000 results from Vista telemetry (metering tools) were analyzed, grouped, and used to develop cause/effect tables, which were then used in subsequent Activities to improve our OS.
- Physical prototypes: 1 samples - we also used the above results to design enhanced telemetry for our next Windows release, to improve the amount and type of information we can collect from our users
- ... prototype revisions: 7 revisions - there were 7 major revisions of the enhanced telemetry tool

**Project Name:** Improve Vista  
**Project Number:** 1002

**Start Date:** 2010-01-01  
**Completion Date:** 2010-08-19

**Results:**

Most statistical software packages are completely dependent upon the user regarding the selection and implementation of the correct model to analyze. Whatever model the user defines, the computer will "fit" that model. Incorrect conclusions can be drawn when inappropriate models are fit to the data. The difficulty is that the computer program cannot inform you when the model is inappropriate

**Conclusion:**

From analysis of the telemetry data we were able to a) determine typical causes of crashes; and b) identify workflow snags and spot common work-arounds (not crashes, but areas where we could make Windows easier to use).

---

## **A c t i v i t y # 1 - 2 : I m p r o v i n g s t a b i l i t y & r e l i a b i l i t y**

**Work performed in Fiscal Year 2010:**

**Methods of experimentation:**

- Physical prototypes: 2 samples - 1 – used results from Activity 1 to improve stability; 2 – some application compatibility issues are caused by poor OS version checking on the part of the application developer. We developed improved "shim" technology which allows Vista's compatibility manager to figure out what's causing an application compatibility issue, and recommend the fix.
- ... prototype revisions: 37 revisions

**Results:**

- Avg. Time Between Disruptions (crashes): 42 hours (108% of objective)  
The use of PLS models eliminates drawback of both Ordinary Least Squares (OLS) Regression and Principal Component Regression (PCR). OLS requires more samples (products) than variables to be included in the model, which is typically not the case when attempting to compare consumer and descriptive data. In fact, it is not unusual to present only 8-10 products to the consumer and descriptive panels yet collect information on a couple dozen descriptive attributes. PCR eliminates this difficulty, as well as the problem of multicollinearity that is present in much of the descriptive data. However, the first principal component, formed from the descriptive data, is not necessarily related to consumer acceptance thereby weakening the model for predictive purposes. The use of PLS through Preference Cluster Mapping eliminates both of these difficulties. Further, because it still fits a model to the data it remains possible to predict consumer liking of new products from the existing model by simply running additional descriptive panels.

**Conclusion:**

The main difficulty with improving stability is that for practical reasons, in-house testing can only be conducted under so many different circumstances. By using and improving our telemetry tool we have essentially greatly increased our testing field.

---

## **A c t i v i t y # 1 - 3 : O p t i m i z i n g w i n d o w s m a n a g e m e n t**

**Work performed in Fiscal Year 2010:**

**Methods of experimentation:**

- Physical prototypes: 1 samples - Re-designed code to optimize double-buffering of windows, preventing memory usage from increasing substantially as more windows are opened.
- ... prototype revisions: 4 revisions - 4 major revisions – main issue was preventing flickering and other visual degradation, when more than 10 windows were open.

**Results:**

A final model can now be chosen that includes only the statistically important descriptive attributes. This model is much simpler than a model including most if not all of the original variables and when used for prediction the simpler model typically shows a decrease in prediction error. Further, a simpler model makes the identification and creation of an "ideal" product much more direct as fewer variables need to be examined for their effect on consumer acceptance. Development costs are also lowered as descriptive panels only need to collect information on a handful of attributes for use in predicting consumer acceptance of these new products.

**Conclusion:**

Our main advancement in this activity was reducing memory usage while maintaining graphics quality.

---

## **A c t i v i t y # 1 - 4 : D e c r e a s i n g l o a d t i m e**

**Work performed in Fiscal Year 2010:**

**Methods of experimentation:**

- Physical prototypes: 1 samples - Re-designed code to load device drivers in parallel instead of sequentially.
- ... prototype revisions: 3 revisions - 3 major revisions – main issue was optimizing load balancing

**Project Name:** Improve Vista  
**Project Number:** 1002

**Start Date:** 2010-01-01  
**Completion Date:** 2010-08-19

---

If you total up the degrees of freedom you see that there are no df left for the error term! Remove the interaction from the model and the problem is solved. (In some cases, if panelist is properly specified as a random effect the test for product differences will still be performed in the above model. Thus, if no tests are performed also check whether panelist is a fixed or random effect

**Results:**

- Avg. Load Time: 27 seconds (83% of objective)  
Avg. load times were tested on 10 test machines configured to span the typical user hardware and software configurations.

**Conclusion:**

By optimizing the load balancing during device driver loading we were able to reduce Vista load time to be almost the same as XP.

---

**A c t i v i t y # 1 - 5 : I m p r o v i n g b a t t e r y l i f e**

---

**Work performed in Fiscal Year 2010:**

**Methods of experimentation:**

- Process trials: 10 runs / samples - Developed correlation between system timer settings and battery life. Determined that a 15.6ms timer can improve battery life by as much as 10% vs. using 1ms timer.
- Physical prototypes: 3 samples - 1- re-designed wireless management to allow it to drop below 100% power draw while managing the connection; 2- optimized OS kernel so that CPU can sometimes run at a lower frequency and stay idle longer; 3- re-designed code to allow changes to system timer settings.
- ... prototype revisions: 8 revisions - 8 major revisions.

**Results:**

- Avg. Laptop Battery Life: 5.32 hours (102% of objective)  
Avg. battery life was tested on 10 test machines configured to span the typical user hardware and software configurations.  
If you total up the degrees of freedom you see that there are no df left for the error term! Remove the interaction from the model and the problem is solved. (In some cases, if panelist is properly specified as a random effect the test for product differences will still be performed in the above model. Thus, if no tests are performed also check whether panelist is a fixed or random effect.)

**Conclusion:**

By integrating the 3 major improvements described above, we were able to increase battery life by 11-15% (depending on hardware and configurations).