The New Configuration Management Game for UK Military Combat Systems

Mr. Michael Looney
DERA Portsdown West,
Portsdown Hill Road,
Fareham,
PO17 6AD, UK
Tel: 44 1705 336515
Fax: 44 1705 336505
e-mail mjlooney@ dera.gov.uk

Mr. George Allan
Senior Lecturer
Department of Information Systems
University of Portsmouth
Hampshire UK
PO4 8JF
Tele:- +44 (0)1705-844034
Fax +44 (0)1705-844006
e-mail George.Allan@port.ac.uk

Abstract

This paper is an assessment of the changes, which are occuring in the configuration control, and management area of military combat systems in light of the commercial component based procurement approaches now being identified for the future. The new challenges are highlighted and a survey of the present research trends are considered to see how well they might support the new configuration management process needed.[1]

1. Introduction

1.1 *Historical background*

The end of the cold war has brought about many changes in areas associated with the development of military software intensive combat systems. They have in the main been caused by the reductions in funding and the need to procure more cost effective solutions resulting in the adoption of commercial technology within what were previously very much 'one off' bespoke systems. Each system was developed and, in most cases, supported, through its operational life either by the original contractor or the military themselves or a combination of both. They were responsible for the modification and enhancing of the system over time and the release of all new versions and upgrades, all aspects of the software were under their control. As a result, logistic support was relatively straight forward and the appropriate processes were established to enable

[1] The views expressed in this paper are those of the authors and in no way represent the policies or opinions of their organisations.

the maintenance tasks and changes to be carried out in a controlled manner. The concept of software obsolescence was never considered, any software module that had been developed was fit for purpose and was used until that specific function either needed modification due to some external change or when it was no longer required at all. Then it was replaced with the new software which met the new functional requirement, and so the system evolved in controlled stages through its operational life.

However this rather cosy existence has come to an end with the introduction of commercial software components into the military domain. The game has changed, the rules are very different, there are more players, they are much more aggressive; military systems are now in the commercial market place where change is the norm and only by keeping one gismo ahead of the competition ensures survival. 'One off buys' are no longer really viable for software components which are obsolete in about half the time the previous military system development cycles took to finish because of the problems associated with licences and vendor liabilities for products. As we all know commercial software products are sold with known faults, they have short lives before a newer, better and faster, but still faulty, version is released. These are all issues which must now be faced by the military procurement organisations.

1.2 *The growth of COTS*

In past developments, contracts to produce software systems (or to act as the prime contractor for a complete computer system) were awarded to companies on a list of MoD preferred suppliers. The MoD would, in the main, dictate the terms and conditions of the contract and hold regular project meetings to monitor and control the development in accordance with MoD requirements and procedures. Subsequent maintenance would be undertaken by the supplier as part of the modus operandi. This was, and is, an expensive financial business and costly in time on the part of both the contractor and MoD personnel.

Today the market is moving toward software houses developing niche markets - becoming specialists in a particular field such as finance and banking, medical systems, or legal systems for solicitors and company lawyers. It may be possible to guide a few suppliers in the specialist area of Defence but it will not be possible to drive the market.

As a consequence of the above, the MoD will have to run with commercially produced off-the-shelf (COTS) computer products (Looney, 1997). The use of COTS products has the advantages associated with reusable software and hardware components (Fonzo, Bradley and Garhart, 1997) which may be acceptable in order to cut costs in a number of areas but can lead to certain difficulties unless proper and rigorous procedures are in place to manage the delivery and subsequent changes to COTS products.

1.3 *Difference in Control*

Software suppliers now develop what they think they can sell. Taking note of the DoD rebuff from Microsoft, it might be a hard reality but the MoD must be seen as a minor player these days in that, while the systems procured are expansive, only few are developed. Thus control has passed from the MoD to the vendor and MoD has a reduced role as paymaster and project manager.

With an increasing amount of software available in the commercial market and changing performances in hardware standards, it is difficult to keep abreast of new products. The issue here is whether the MoD customer knows the full characteristics of what is being received from the supplier. The problem will be to know exactly what is embedded within a computer system at delivery and then to control what gets put in thereafter.

1.4 *Unstable market place issues*

If the MoD are procuring COTS products – both software packages and hardware packages – from several suppliers who act independently of each other. There are numerous questions to be answered include:

- What if a supplier ceases to trade – will another supplier's product do the same job?
- Will this alternative product be compatible with existing packages?
- What if a current supplier changes future releases of software –
- How can MoD be confident of upward compatibility?
- What is the effect on interfacing with previous releases of the systems?
- How could MoD cope with these problems in the worst case scenario?
- When procuring a new system will it interface with existing systems?

2. The Challenges of COTS Components

As has been indicated, COTS software differs from other non development items (NDI) since commercial markets, independent contractors, and vendors control the design configuration and support (i.e. enhancements, modifications, and upgrades) of COTS products, not the MoD. COTS software is customised by the vendor in a way intended for his anticipated market, but is not modified by the MoD if it is to remain "COTS" software by definition. The advantages claimed for COTS software include cost, schedule, availability, reliability, standardisation, lower development risk, and ease of migration to future technologies. The MoD needs not only to be aware of the general technical challenges associated with COTS acquisition (identification, assessment, selection, supportability, etc.) but should also consider such challenges as integration/interoperability and system software architecture and growth potential and how to control or manage what the market place provides.

2.1 *System Software Architecture*

To take full advantage of a COTS components based approach to system development, an open systems architecture is needed. An open systems architecture requires the identification and specification of system interfaces, functionality and data schemes in a way that allows future commercial technology upgrades to be adopted. In the development of an open system, modules that will not change should be separated from those of a more dynamic nature which will evolve over the system life. To achieve an open system, COTS products that are scaleable, portable, interoperable, conformance tested, and independent of their hardware platforms should be selected. Open system interface standards are the basis for successfully selecting and integrating COTS products from many different vendors and the profiles which identify, document and baseline the interface features to be used in the system should be maintained as part of the configuration. As a result of this process of documenting and verifying system interfaces, the engineering data necessary to accomplish technology insertions and product upgrades becomes available and must be part of the process of CM.

2.2 *Integration and Interoperability*

Interoperability, or the ability of two or more systems to exchange information and effectively utilise the information exchanged, is a key issue with COTS products. To achieve this COTS software products should meet known, accepted, and where possible non-proprietary interface standards. The requirement must be to select COTS products which assessment shows to be compliant with such accepted interface standards. Using controlled, standardised interfaces should facilitate future changes and upgrades without impacting the entire system. However, the MoD will need to plan for the allocation of additional integration time and resources to resolve any COTS interface and performance problems during the system development and test. As part of the COTS assessment and selection process, the ability of a specific COTS product's interoperability with other products to be used in the system under development should be determined. The use of some form of flexible ‘middleware’ can also allow disparate products to interact without having to modify those products. For COTS products with no positive interoperability records, either another COTS product with interoperability data should be found or the product identified as a risk item. For those critical COTS products, prototyping should be considered as a means to identify interoperability issues early in the development. The MoD will need to plan for possible cost and schedule impact during the integration and test phases resulting from integration and interoperability problems associated with COTS products.

2.3 *Life Cycle Support*

During the development and also when a system becomes operational, it must be monitored to identify any support related concerns (to include consideration of product obsolescence and product performance issues). A continuing market research process must be put in place which takes into account the support information collected during the in service life of the system to provide the identification of the relevant market developments. Alternative COTS products (primarily at the lowest replaceable unit level) will need to be identified, and evaluations conducted in order to select the most effective alternative COTS products. This approach then drives the procurement, integration, test, checkout, and ultimate deployment of system upgrades and the associated support.

As actual operational data is collected, it should be used to identify the opportunities to tailor the system structure and identify areas for potential improvements in terms of newer, improved products to be integrated into the next available system upgrade. In this way, the market research, support assessment, and system upgrade processes are part of a continuous improvement process for the operational system carried out as part of the overall CM process.

2.4 *Licences/Warranties*

Program Managers must implement a warranty program to take advantage of warranties available from COTS product vendors. Most vendors of COTS hardware include a standard failure-free warranty in the price of the item. The length of the warranty varies from a few months to 3 or more years.

Warranties come in several variants, but basically there are four types:

- Standard Hardware/Software Repair/Replacement
- Extended Hardware Repair/Replacement
- Hardware Maintenance Services
- Software Maintenance Services

Standard warranties provide basic assurance that the hardware or software item purchased will be "defect free" and functionally correct for a defined period of time. Generally, that period of time begins when the item leaves the door of the vendor.

Commercial software warranties usually apply to the integrity of the media storage device (diskette, CD ROM or magnetic tape) and the operability of the software in a stand alone environment. Liability is limited to the replacement of the magnetic media only and no responsibility is accepted for any other consequence resulting from the use of the software component integration with other software and the effects it has on other products or its ability to work with any other product are not covered in any way. Testing of software upgrades must be carried out to determine system compatibility in the same way that they were performed during the assessment and selection of the original component.

Standard warranties may be of little use to the Government because:

- they generally do not go beyond the original purchaser
- the time period is fairly short in duration when compared with system development time let alone the in-service time frame

Software Maintenance Services provide the user with updates and upgrades to the purchased software on a renewable service basis usually covered by the licence fee. This option ensures visibility and accessibility to the latest versions including possible fixes to problems but very much at the vendors priority not that of the customer.

Each vendor's warranty should be analysed to evaluate its cost, benefit, and effectiveness. The warranty evaluation should consider:

- Warranty Annual Cost
- Predicted Failure Rate
- Repair Cost (Out of Warranty)
- Change History
- Warranty Transfer
- Warranty Administrative Cost
- Support Period Warranty
- Anticipated Change Frequency

Warranty evaluation analysis and evaluation should be conducted as part of the supportability assessment performed during market research and assessment prior to selection and the information retained as part of the CM task.

2.5 *The New Process of Configuration Control*

Using COTS products will allow for a more rapid technology refresh throughout the life of the system and the use of COTS configuration items (CI) must be supported during all of its phases with a strong Configuration Management (CM) process. Since vendors will individually control the configuration of their products rather than the MoD, COTS products will need to be configuration managed rather than configuration controlled by the MoD. CM of the COTS-based systems will occur at the LRU level and this will be very much dependent on the type of system and its size and complexity. It can be defined as the systematic proposal, justification, evaluation, co-ordination, approval (or disapproval) of a proposed change, the implementation of those approved changes, and the management of a CI after establishing a baseline for that component. CM begins with the generation of the initial baseline and continues throughout the life cycle of any CI.

It might be possible to hold baselines of rapidly changing COTS components at a lesser level of detail, therefore requiring a less complex CM process to be carried out on them. However, it is important that the necessary open system information, such as interface and standards profiles, is maintained at the appropriate level of detail to allow the replacement/upgrade task to be carried out effectively while the level of internal detail of the component is of lesser concern.

Changes will either be driven by an upgrade programme (performance enhancement or additional capability requirement) or by a technology refresh process of form, fit & function (F3) substitution to avoid obsolescence. Typically an upgrade results in a specification change and this requires a particular level of testing/retesting but if the refresh is driven by changes which do not directly impact the F3 specification then a lower level of testing/retesting might be appropriate. The extent to which a substitute COTS item is 'one for one' replaceable and the amount of testing/retesting required will depend on the selection factors which have been applied, such as product history, vendor history, compatibility both up and down, interchangeability etc. However, some items will have substantial market-generated performance data. This data may actually be more extensive than that generated through testing programs or experimental use and again may impact the level of testing required.

In the situation where several COTS items are being integrated for a military application, diagnostic software and/or built-in-tests for each of the individual components are less likely to be standard and a diagnostic shell may be needed to provide a single status indication and fault isolation capability. This will require its own baseline and become a CI which will evolve with any changes to that set of components

All of these aspects must be considered as part of the actual CM process.

3. Current Areas of CM Research

In their position paper, van der Hoek, Heimbigner and Wolf (1995) raise the question of whether Configuration Management research has a future. The new standard in CM systems - typified by products such as Adele, ADC, ClearCase, Continuus/CM, and CCC/Harvest - largely satisfies the basic commercial CM functionality requirements posed (Dart, 1991). This implies that research in the area of CM is either unnecessary or that new challenge in CM be found. The authors believe that challenges exist and conclude that CM research does have a future. From the MoD point of view there is very clearly a need to carry out research in this area. As already indicated there are several new challenges arising due to the changes which are being put in place in the procurement of software intensive systems to achieve the same capability but at reduced through life costs.

In industry and academia, research in CM is currently in two major directions: one in investigating new CM methods for improved software development and the other in developing system-wide CM. Further work is being conducted to ascertain how CM-aware IS/IT people are and how wide-spread its use. This section deals with these three areas.

3.1 *CM for Software Development*

Software CM (SCM) is defined as the tracking and control of software development and its activities (Eaton, www). There is considerable research in this area being carried out by the Software Engineering Research Laboratory (SERL) at the University of Colorado, USA. The major projects are as follows.

NUCM (Network-Unified Configuration Management) is a peer-to-peer repository supporting distributed CM where the programmatic interface facilitates rapid development of a CM system. (van der Hoek, Carzaniga, Heimbigner and Wolf).
SRM (Software Release Management) is a process, which makes software available to clients and users. (van der Hoek 1998)

DVS - pronounced “devious” (Distributed Versioning System) for distributed artifacts and collections (in particular software modules). This system adopts a linear versioning schema and a co-operation policy based on check-out/check-in with exclusive locks as necessary used to maintain files, directories and other collections of artifacts.

Estublier & Casallas argue that software quality can no longer be understood without taking account of the Process Model (actors, activities, methods, tools and procedures used in development).

A main difficulty in providing formal methods is the duality between the Repository space and the Work space. Repository space - stores the software objects and their versions. Work space - users create/access and modify these objects. OMS=Object Manager Systems.

There is a problem coping with version evolution, most commercial CM packages have poor data models - they propose file management instead of software management. Versioning in SEE (Software Engineering Environment) requires a data model to cope with versions, variants and revisions.

Aide De Camp is a system based on the Change Set Model where a change set is a complex object containing the changes (file fragments).

The Adele Data Model is object-oriented where objects represent files, activities, functions, strings, and dates. Adele relies on a pre-defined product model that defines *original* objects, *destination* objects and the *relationship* between them where relationships are independent entities.

'R'
'O' ------------------------> 'D'
relationship

3.2 ***System-Wide CM***

It is claimed (Zimmermann, www) that systematic CM eliminates duplication of effort and keeps effective control over planned and in-progress implementation activities. If CM is implemented system wide it should provide a reliable and unambiguous source of information on the operational IS/IT systems. It ensures the availability of valid baselines for security audit purposes in addition to concise technical data for maintenance personnel.

Because traditional configuration management systems have focused on the development activity of source code control, few mechanisms are currently in place to support software deployment activities. Software deployment is the complex process that covers activities performed after software has been developed. These activities include configuring, releasing, installing, updating, reconfiguring, and even de-installing a software system. (Hall; Heimbigner; van der Hoek, and Wolf 1997) The deployment process involves the careful co-ordination and interaction of multiple producers (including developers) and multiple consumers (including users) that may be geographically and organisationally dispersed.

The Software Engineering Research Laboratory (SERL) at the University of Colorado, USA has active research programmes in this area. The Software Dock is a distributed versioning system for collections of artefacts (Hall, Heimbigner, van der Hoek, and Wolf, 1997). They rightly observe that there is a need to develop a new generation of configuration management technologies that account for post-development activities. To be effective, these new technologies must:

- Operate in a variety of environments, ranging from a single machine to a distributed, decentralised, wide-area setting that leverages the connectivity offered by networks such as the Internet.
- Provide a way to describe site and software system configuration information.
- Provide a way to manage, access and reason about site configuration information, which includes the hardware and software environment at a site.
- Provide a way to manage, access and reason about a software system configuration, which includes dependencies and constraints inherent in the system, both for initial installation and as a way to maintain the working state of a system.
- Make it possible to monitor the environment surrounding a deployed system, watch for changes in that environment and automatically adapt the software to those changes.
- Allow consumers, in loose co-operation with software producers, to maintain their installed system with little or no need for human intervention).

The Software Dock is a system of loosely-coupled, co-operating, distributed components that are bound together by a wide-area messaging and event system. The components include field docks for maintaining site-specific configuration information by consumers, release docks for managing the configuration and release of software systems by producers and a variety of agents for automating the deployment process. Both the information about release and the information about field sites are represented as hierarchies of data that, when combined, form a federated software deployment registry with the conceptually global name space. Events generated by operations on the hierarchies propagate throughout the federated registry and are received by interested agents. The agent technology enables concomitant actions to be automatically performed in response those events

The Software Dock architecture consists of five primary components,

- A field dock is a server residing at a consumer site that maintains a registry of local configuration information. It provides interfaces for agents to access the information and to subscribe to events that are generated when information in the registry changes.
- A release dock is a server residing at a supplier site that provides a registry of information about software releases. It provides interfaces for agents to access the information and to subscribe to events.
- A federated deployment registry is a conceptual aggregation of the field dock and release dock registries that provide a global name space.
- An agent is a executable program that performs specific tasks of the deployment process at release and field sites on behalf of producers and consumers.

- A wide-area messaging/event system (WAM/E) is an Internet-scale notification mechanism that enables the flow of messages and events between agents located at release and field docks.

Much of the above research is addressing the future problems of distributing software to the customer IS/IT platforms direct from the developer/supplier.

Another centre of research is The CM Research Foundation at Leeds University Business School (Kidd, undated, www) which is attempting to form “research clusters”. They are partners in a project (AdCoMS, undated, www) for an advanced configuration system. This is a European research project supported by a consortium of European manufacturers in the automotive and aerospace sector brought together to develop requirements and specifications for future IT tools and although work has not significantly progressed, it is being monitored with interest.

Research by Tohjo and Yamamura (1997) draws on a comparison between a conventional paper-based approach and what they call the Construction Manager System (CMS). The paper considers “resource and service provisioning” as main CM processes. This work introduces the “construction ticket” to define work-to-be-done. It uses diagrams to illustrate “flow-through” but is unconvincing in its argument that this improves traditional CM.

4. CM Awareness

From participation in consultancy and research in other computing areas, it is apparent that many companies do not know what CM is and are unaware of the benefits (Allan, 1997a; Allan, 1997b). A research project was conceived in September 1997 to ascertain the current awareness of CM in UK industries, in particular within IS/IT departments supporting core business.

An empirical investigation took place to estimate how many UK companies use CM and to what extent. Research approaches were considered for the best way to plan and execute this investigation. The large number of companies involved and the diversity of their geographic locations favoured the survey as the research method. The research questions would address CM awareness and usage. If this could be established then the use of CM within company could be further investigated to consider CM maturity in UK companies. Thus the research method most appropriate was a passive survey using a questionnaire to gather initial data followed by selective structured interviews for companies already aware of and using CM. Therefore the research was planned in two stages:-

- Stage 1 ascertained the awareness and usage of CM in UK companies.
- Stage 2 is revisiting selected companies currently using CM and to conduct an in-depth maturity study in an attempt to ascertain any trends within industry.

4.1 *Conclusions from UK survey*

CM awareness in the UK is in its infancy. The research found that 60% of respondents had no awareness of CM.

It is probable that some of the non-respondents did not understand CM and/or had not heard of it. This will compound the empirical estimate of 60% unawareness.

Larger IT departments tend to be CM aware amongst all their IT/IS staff - this might indicate a CM culture in businesses where IT has a high profile.

In larger companies with high awareness and usage of CM, the task of CM manager is often shared. This might suggest a direct proportionality between CM maturity and CM dedication: the greater the CM maturity within company the greater the requirement for dedicated personnel.

In general, UK companies with a history of CM maturity tend to use it for software more than hardware.

5. Conclusions

With future builds and modifications of military systems being dependent on commercial components there is likely to be an increased need for awareness of the market place and its impact on the availability and technical complexity and these must be addressed in any processes being adopted. To maintain future competitive advantage, it is likely that software changes and updates will be much more frequent and that it may well not be possible to physically check all aspects of the software packages prior to installation without a prohibitive amount of time during which systems are out of operation or due to the limited numbers of skilled software and hardware engineers. There will be a need for a process of configuration management from the developer's initial prototypes, through full development, delivery and installation to operational running which must retain the necessary information associated with the components being used and their relationships with all other parts of the system. It will be necessary to have mechanisms to know what software is being received and what is embedded within the software package, but it will no longer be enough just to know what is in a system. While the MoD's role in controlling the actual changes in the software components may be reduced it should have an increased part to play in ensuring an evolutionary path is available to allow the adoption of newer, more capable technology as it becomes available. As has been pointed out there must be a high level of market awareness associated with the process to cope with the upgrading of the components either through obsolescence or changed requirements. The minimisation of the impact of installation, the knock on or ripple effect through the operational system should be a primary objective and that will only be achieved with a very thorough CM process in place.

The researchers such as those from SERL (Hall, Heimbigner, van der Hoek, and Wolf, 1997) offer an architecture for CM in a wide-area network. Much of this work is based on the PhD work and thesis of Hall (1998) and the intelligent agent method of the software dock could be a workable and efficient method for the distribution of software components in the future. This could be seen as a way forward and should be considered in any future MoD research work. Also Dart (1996) outlines ten steps necessary when implementing CM and gives an evaluation method for selecting an appropriate CM tool once the process is in place. She emphasises that modern CM can provide benefits in QA processes and change management.

The CCTA in their Information Technology Infrastructure Library (ITIL) emphasise that CM is an important aspect of any development and that successful implementation depends on having sufficient skilled and properly trained staff . The views being expressed by the DoD groups, looking at the problems associated with the introduction of COTS products, indicate that they feel that, in future, a system may well never reach a ‘stable’ state and that there will be continual development for the whole of the in-service life which will entail a major new approach to the CM activity.

6. Recommendations

There is a need for continued research in how best to carry out the future configuration management for the MoD and this should be part of the overall process investigation related to such area as the cost prediction because unless the support process is in known then the through-life costings can not be assessed.

A robust CM process should be developed by MoD to meet their needs in the new commercial component based approach to systems development which can be included in all contracts for the procurement and maintenance of operational computer systems.

Some of the activities necessary to ensure the overall process has the right capabilities include the following:

- Embedded CMDB (CM database) governed by a security kernel.
- Qualified, experienced technical competent staff to populate the CMDB during a project.
- Mechanisms to maintain MoD independence in the market place while making the best use of COTS products.
- Contractors should submit a CM plan including recommended CM tools for use in support of technology upgrades as part of their tender.

7. References

Allan, G.W., 1997a, “What is Configuration Management?”, Logistics Spectrum, Vol. 31 No. 1: pp. 15-18, ISSN 0024-5852.

Allan, G.W., 1997b, “Configuration Management and its Impact on Businesses that use Computer Platforms”, International Journal of Project Management, Vol. 15 No 5: pp. 321 - 330, Pergamon

BS 5515:1984, British Code of Practice for Documentation of Computer Based Systems

BS 6488, 1984, British Standard Code of Practice for Configuration Management of Computer-Based Systems

Dart, S., 1996, *Achieving the Best Possible Configuration Management Solution,* http://stscols.hill.af.mil/Cross Talk/1996/sep/achievin.html

Dart, S., undated, *Not All Tools Are Created Equal,* http://www.adtmag.com.pub/oct96/fe1002.htm

Dart, S., 1991, “Concepts in Configuration Management Systems”, *Software Engineering Institute*, Carnegie-Mellon University, Pittsburgh

Eaton, D., 1999a, *Configuration Management Tools Summary*, http://www.iac.honeywell.com/Pub/Tech/CM/CMTools.html

Eaton, D., 1999b, *Configuration Management Frequently Asked Questions*, http://www/iac.honeywell.com/Pub/Tech/CM/CmFAQ.html

Fonzo, A.G., Bradley, M.K., and Garhart, J.S., 1997, “Use of Commercial Products for Configuration Management of Military Network,” *IEEE MILCOM*, Vol. 3, pp. 1351-1355

Hall, R.S., 1998, “Software Deployment in a Wide-Area Setting Using A Distributed Agent-based Architecture”, PhD *Thesis*, University of Colorado, USA

Hall, R.S., Heimbigner, D., and Wolf, A.L., undated, “Software Dock”, http://www.cs.colorado.eud/users/serl/cm/dock.html

Hall, R.S., Heimbigner, D., van de Hoek, A., and Wolf, A.L., 1997, *The Software Dock: A Distributed, Agent-based Software Deployment System*, University of Colorado

Hall, R.S., Heimbigner, D., van der Hoek, A., and Wolf, A.L., 1997, “An Architecture for Post Development Configuration Management in a Wide-Area Network” *Proceedings of the 1997 International Conference. on Distributed Computing Systems*, Baltimore

Kidd, C., undated, AdCoMS, http://adcoms.finmeccanica.it/theproj.html

Looney, M., 1997, *The Wider Implications of COTS Products*, HMSO

SERL, undated, *Current Research Projects*, http://www.cs.colorado.edu/users/andre/research.html

van der Hoek, A., Heimbigner, D., and Wolf, A.L., 1996, “ A Generic Peer-to-Peer Repository for Distributed Configuration Management”, *The Proceedings of the 18th International Conference on Software Engineering*, Berlin

van der Hoek, A., Heimbigner, D., and Wolf, A.L., undated, “Investigating the Applicability of Architecture Description in Configuration Management and Software Deployment”, http://www.cs.colorado.edu/serl/arch/Papers.html#CU-CS-862-98

van der Hoek, A., undated, *Configuration Management Yellow Pages*, http://www.cs.colorado.edu/users/andre/configuration_management.html

Wallnau, K.C., 1993, “Issues and Techniques of (CASE) Integration with Configuration Management

Zikmund, W.G., (1997), Business Research Methods, USA, Dryden Press

Zimmermann, M., undated, “Just a fashion or a profession”, *Configuration Management*, http://www.usb-muo.com/whitepap.htm