Update/Patch Management Systems

Chapter · January 2011

DOI: 10.1007/1-4020-8145-6_5

3 authors, including:

Clark David Thomborson
University of Auckland
136 PUBLICATIONS **6,431** CITATIONS

SEE PROFILE

UPDATE/PATCH MANAGEMENT SYSTEMS:
a protocol taxonomy with security implications

Andrew Colarik, Clark Thomborson, and Lech Janczewski
The University of Auckland, New Zealand

Abstract: Software fixes, patches and updates are issued periodically to extend the functional life cycle of software products. In order to facilitate the prompt notification, delivery, and installation of updates, the software industry has responded with update and patch management systems. Because of the proprietary nature of these systems, improvement efforts by academic researchers are greatly restricted. One solution to increasing our understanding of the underlying components and processes is architectural recovery. One contribution to recreating an architecture is the examination of design specification literature, such as patents. If a sizeable amount of similar and hopefully diverse patents can be examined, then some general conclusions about the components and processes of existing systems may be formulated. In this paper, we present an analytic framework consisting of a five-phase protocol taxonomy based on thirty-three software-based update and patch management system patents and patent applications. Furthermore, we present a decomposition of the security design provisions contained within the patent literature, and provide some general trends derived from the data. We suggest that this research may be used to improve the security services aspect of update and patch management system products.

Key words: Architectural Recovery, Taxonomy, Patches, Updates, Patents, and Security Design Provisions.

1. INTRODUCTION

At the core of the maintenance phase of the software development life cycle are the issuance of patches (software fixes) and updates (a collection of fixes and improvements) to resolve system faults, flaws (bugs), and security holes in an attempt to extend the functional life of a software product. Due to the time and effort required to assess, locate, and acquire these updates, this

on-going effort is often delayed or over-looked by users and system administrators until some urgency or incident occurs that prompts a swift response. In recent years, software manufacturers have typically provided access to their product updates via the Internet (i.e. website, ftp, e-mail, bulletin boards and newsgroups). Interestingly, connectivity to the Internet has also created an additional burden to the issuance of updates and patches. The CERT Coordination Center maintains statistics on the number of vulnerabilities reported that can potentially be / have been exploited through malicious acts, virus infections, and self-replicating worms, among others. For the past three years, the vulnerabilities reported have continued to nearly double from the previous year. There were 1090 reported vulnerabilities reported in 2000, 2437 in 2001, and 4129 in 2002 [CE03]. The need for systematic notification, acquisition, and deployment of patches and updates has prompted the software industry to produce update and patch management systems. There are numerous producers and products of patch and update systems (see Table 1: Examples of Patch/Update System Products).

Table 1: Examples of Patch/Update System Products

Company Name	Product
BigFix Incorporated	BigFix Patch Manager
Bindview Corporation	bv-Control
Citadel Security Software	Hercules
ConfigureSoft	Security Update Manager
Ecora Corporation	PatchMeister
Gibraltar Software	Everguard
Harris Corporation	STAT Scanner
Hewlett-Packard	Security Check Patch
McAfee Security	OilChangeOnline
Microsoft Corporation	Windows Update
PatchLink Corporation	Patchlink
Ringmaster Software Corporation	Ring Master
Shavlik Technologies	HFNetChkPro
St. Bernard Software	UpdateEXPERT
Sun Microsystems	Patch Management Module

From a design and academic perspective, a primary problem emerges: How do we, as researchers, peel away the proprietary tendencies of organizations to hide the inner designs and processes of their products in order to better understand, communicate and hopefully improve the development of such systems? In the event that the original architectural

design literature is unavailable, one possible approach is to reverse engineer the architecture through the use of system code, views, and documentation [Ei98]. The authors of this paper propose a supplemental approach using the information disclosed by inventors in patents and patent application documents, and in particular when such resources are unavailable.

In section two of this paper, the authors present a discussion on the contributions of a taxonomy towards reconstructing system architectures. In sections three and four, we present our patent search criteria, and provide a protocol phase taxonomy derived from 33 update and patch management system patents and patent applications. We then re-examine the patents for security design provisions by each phase of the taxonomy in section five, and present a discussion of the security design implications and limitations of our findings in section six.

2. CONTRIBUTION OF A TAXONOMY

Before a discussion on the significance of a taxonomy towards reconstructing software architectures may occur, some basic understandings of what comprises an architecture, and some of the issues in acquiring and documenting an architecture needs to occur. Shaw & Garlan (1996) state that:

> *"The architecture of a software system defines that system in terms of computational components and interactions among those components. Components are such things as clients and servers, databases, filters, and layers in a hierarchical system. Interactions among components at this level of design can be simple and familiar."*

As academics attempting to conceptually improve on existing systems, poorly documented or non-existent architectures (in documented form) pose a significant problem. Even when architectures do exist, they may no longer be valid because many systems simply have evolved beyond their original documentation due to in-process design development, and maintenance of existing code to adapt to changing conditions [Ka99]. Thus, reverse engineering and decomposition of existing systems becomes essential. The process begins at the lowest level of abstraction with an examination of the product's source code and documentation. These are used to develop a set of software views with the use of domain knowledge by the researcher. Combining all these elements, in theory, should lead the researcher to developing a set of architectural elements that will be used to formulate the system's architectural representation [Ei98].

The discovery process of re-creating a software's architecture involves an assembly of disparate sources of information by interpretive means. It is a very subjective process that may result in substantial variation from one researcher's interpretation to another. Because the researcher's interpretation is based on available information and the researcher's own understanding of the subject matter, the process can be prone to error. It would, therefore, be reasoned that any additional contribution to clarifying the components or subsystems that comprise an architecture adds to the accuracy and consistency of its reconstruction. We propose that a taxonomy is a useful tool in architectural reconstruction, and creating a common reference point for researchers to improve existing products and processes.

The American Heritage Dictionary (2000) defines "taxonomy" as "the science, laws, or principles of classification"; and the "division into ordered groups or categories" [Jo00]. The significance to architectural discovery provided by a taxonomy lies within the context of the ordered groups. The contributions provided by a taxonomy towards understanding an underlying architecture are:

- Additional domain knowledge by which a researcher may interpret other aspects of accumulated design documentation and coding decomposition analysis [Ei98],
- Provide a visual representation as to how the components detailed are organized [Ka99] [Pa00],
- Explicitly/implicitly ask who, what, where, when, and how questions in order to provide abstractions to the corresponding categories of the taxonomy [So92], and
- Provide a means for defining what data are to be searched for and recorded, as well as a means for a comparison between specimens [La94].

In the following sections, we present our patent search criteria, the proposed, developed taxonomy, and apply the taxonomy phases to the security design provisions contained within the patent documents.

3. SEARCH CRITERIA

The original focus for this line of research was to add to our understanding of the fundamental design components and specifications that are predominant in the updating systems environment by providing a representative sample of global patents. Thus, our intention was to review

the patent literature that represented complete, software-driven systems for the purpose of studying the interconnecting processes. To be eligible for our review, the literature had to involve the update of operational and/or application code excluding firmware (i.e. not processors, modems, etc). The emphasis of the search was placed on "method of" instead of "apparatus for" in order to reduce the amount of hardware dependence in the specification. Where appropriate, patent applications were also included. Lastly, due to the popularity of English regarding end-user targeted markets, and commercial development centres, our emphasis was placed on patents filed in English.

The first step was to search the patent databases' abstracts for occurrences of "patch", "update", "dissemination", etc. in order to initially narrow the volume of potential systems. This initial step reduced the possible systems to approximately 2,000. After appraising the remaining abstracts, 100 patent and patent applications were selected for detailed examination by applying the search criteria. This process resulted in 24 patents and 9 patent applications from which we formulated a better understanding of the state-of-the-art in update and patch management systems.

Because our goal is to develop a technological understanding of update and patch management systems, we examined only the descriptive matter and diagrams of each patent. We did not analyze the claims of any patent for their novelty, originality, or specificity. Such matters are of vital importance in legal proceedings, but were not a factor in our development of the taxonomy.

4. TAXONOMY PHASES PRESENTED

During the detailed examination of the patent documents, we discovered that the patents contained a sequence of communication steps for initiating a communication session, performing some exchange of information regarding updates, performing some determination as to the requirement to deliver an update, transporting the update, and initiating an installation. Within each of these protocol phases, there emerged distinct categories as to the means or methods to facilitate each phase. It is these phases and categories that we base the following taxonomy (see Table 2: Updating / Patch Management Systems Protocol Taxonomy).

Table 2: Updating / Patch Management Systems Protocol Taxonomy (5 sub-taxonomies)

	Activity Phase	Category		Count	Patents (see references)
Protocol	Contact	Client	User Defined	19	A2, A4a, A7a, A8a, A9a, P1, P2, P3, P7, P8a, P9, P12a, P13, P14, P15, P16, P20, P22a, P24a
			System Defined	19	A4b, A5, A6, A7b, A8b, A9b, P4, P5a, P6, P8b, P10, P11a, P12b, P17, P18, P21, P22b, P23, P24b
		Server	User Defined	2	A3a, P11b
			System Defined	6	A1, A3b, A7c, A8c, P5b, P22c
		Integrated / Combined		1	P19
	Selection	Client to Server		20	A2, A4, A5, P2, P3, P4, P5, P7, P8, P9, P10, P12, P14, P15, P16, P17, P18, P20, P23, P24
		Server to Client		10	A3, A6, A7, A8, A9, P6, P11, P13, P21, P22
		Integrated / Combined		2	A1, P19
		No Exchange		1	P1
	Determination	Index / Manifest / Table		28	A1, A2, A3, A4, A5, A6, A7, A8, A9, P1, P2, P3, P5, P6, P7, P9, P11, P12, P13, P14, P15, P16, P18a, P19, P20, P21a, P22, P23
		State Change		3	P10, P18b, P21b
		Configuration Information		4	P4, P8, P17, P24
	Transport	Client	Pull	25	A1a, A2, A3a, A4, A5, A6, A7a, A8a, A9, P1, P4a, P6, P9, P10, P11, P13, P14, P15, P16, P17, P18, P20a, P21a, P22a, P24a
		Server	Push	11	P2, P3, P4b, P5, P7, P8, P12, P19, P20b, P21b, P23
			Reference Location	6	A1b, A3b, A7b, A8b, P22b, P24b
	Installation	Client	Manual	11	A1, A2a, A5, A6a, P1a, P2a, P3a, P7a, P12a, P16a, P20a
			User Enabled	19	A2b, A6b, A7a, A8a, P1b, P2b, P3b, P4a, P5a, P7b, P9, P11a, P12b, P13, P14, P15a, P16b, P19, P20b
			Automated	13	A2c, A3, A6c, A9, P5b, P6, P12c, P15b, P16c, P17, P20c, P21, P23
		Server	User Enabled	3	P8a, P11b, P24a
			Automated	12	A4, A7b, A8b, P2c, P3c, P4b, P7c, P8b, P10, P18, P22, P24b

The above activity phases can best be thought of as answering the following questions:

- Contact: When is contact initiated and by whom?
- Selection: Where is the exchanged information compared?
- Determination: What is the basis for the determination of an update?
- Transport: How is the update acquired?
- Installation: Who has control of the installation?

Within each phase, there are self-explanatory categories identifying the methods discussed in the patent literature. However, we would like to clarify several categories within the Contact, Selection and Determination phases. In the Contact and Selection phases, "Integrated / Combined" designates that the processes are a combination of communications and transfers of information between the client and the server (much more than a handshake). In the Determination phase, "Index / Manifest / Table" refers to a software data list that is used for comparison with a master list. "State Change" refers to the status-data of the software. When a given state change is detected between two systems (client/server), the master system (server) restores or updates the software of the servant (client) to the new state.

It should be noted that within each patent there were variations and multiple embodiments of the claims presented. This allowed an inventor to assert variations on the approaches/methods claimed. Thus, in Table 2 there are multiple category entries within each phase for the same patent. For instance, patent P8 describes an embodiment in which a client permits the user to initiate the communication manually (P8a: user defined) or the client contacts the server on a timely/periodic basis (P8b: system defined).

5. SECURITY PROVISIONS

The term "secure" carries with it a multitude of subjective implications and exceptions. Before a software product can be identified as secure, the security objectives, i.e. "a statement of intent to counter identified threats and/or satisfy identified organization security policies and assumptions" [Co99], must be considered with regards to standard security fundamentals. Because these security objectives are either confidential, poorly documented or non-existent, we needed a way to examine each phase of the protocol taxonomy so that any security provisions contained within the design documentation (i.e. the patent) could be identified and classified. Our purpose was to facilitate the emergence of any potential design trends [Pa00].

Therefore, utilizing the The Open Group Architectural Framework (TOGAF) Security Services guidelines as a basis for considering any design considerations contained within the patents, we re-examined the documentation by each phase of the protocol. These guidelines are based on the Technical Architecture Framework for Information Management (TAFIM), developed by the US Department of Defence, and are used in the development of an IT architecture. The guidelines are outlined in Table 3.

Table 3: TOGAF Security Services Guidelines [Op03]

Service	Criteria
Identification and authentication	Identification, accountability and audit of users and their actions
	Use of authentication and account data
	Protection of authentication data
	Active user status information
	Password authentication mechanisms
System entry control	Security-aware warning to unauthorized users
	Authentication of users
	Information about login attempts
	User initiated locking of a session
Audit	Authorized control and protection of the audit trail
	Recording of security-relevant events
	Audit trail control, management and inspection
Access control	Access control attributes for subjects and objects
	Enforcement of rules of access control attributes
	Enforcement of access controls
	Control of object creation and deletion, including reuse of objects
Non-repudiation	Proof that a user carried out an action, or sent or received some information, at a particular time
Security management	Secure system set-up and initialization
	Control of security policy parameters
	Management of user registration data and system resources
	Restrictions on the use of administrative functions
Trusted recovery	Recovery facilities in ways that do not compromise security protection
Encryption	Ways of encoding data such that it can only be read by an appropriate key or other secret information
Trusted communication	A secure way for communicating parties to authenticate themselves without the risk of masquerading
	A secure way of generating and verifying check values for data integrity
	Data encipherment and decipherment
	A way to produce an irreversible hash of data for support of digital signature and non-repudiation functions
	Generation, derivation, distribution, storage, retrieval and deletion of cryptographic keys

Each patent was re-examined with regards to the TOGAF security criteria (see Table 3). An allocation to an appropriate phase was assigned if the

patent provided some account of a security service or mechanism that matched at least one of the TOGAF Security Services' qualifications. An example would be that at the installation phase, a given inventor states that a digital certificate would be included with the update file to ensure the file's integrity and source of origin. The patent would be allocated to the non-repudiation and trusted communication provisions of the installation phase. Table 4 summarizes the results.

What we have attempted to provide is a classification of the assertions made in the body of patents and applications, with regards to any security design provisions that the inventor has proposed to include in their particular invention. What we have not provided in this research is any evaluation of the feasibility, reliability or efficiency of any of the specified mechanisms or systems. Such evaluations would be a fruitful, albeit difficult, subject for future research.

Our analysis of the patent literature is summarized in Table 4. We observe that inventive activity has focused mostly on the provision of security in the installation phase (75 counts in Table 4), with relatively little attention being paid to the selection and determination phases (24 and 17 counts). Frequent use was made of audit and trusted communication, but very little mention was made of security management and access control.

Overall, the patents and applications take a reasoned approach to providing security in update and patch management systems. Even so, the empty cells in Table 4 reveal security provisions that are not discussed in the patent literature we reviewed, but which we believe must be addressed at some point in the development of a truly secure patch management system.

- System Entry and Control Services, in the Determination phase (0 counts). The patent literature reveals no provision for authenticating users or objects. This may become a problem if the data to be compared is encapsulated as an object.
- Access Control, in the Transport phase (0 counts): no provision for privilege management. This function could be used as a push distribution point for malicious code in the same sense that viruses exploit E-mail address books to distribute themselves.
- Non-Repudiation, in the Contact phase (0 counts): no proof of identity, even though transactions may be audited. In the Determination phase (0 counts): a lack of non-repudiation may be a limiting factor in secure deployments where it is important to establish proof of origin, original content, delivery, and/or original content received.

Table 4: Number of Patents with Various Security Provisions, by Phase

33 Patents Total		Activity					Total Counts
		Contact	Selection	Determination	Transport	Installation	
Security Provisions	Identification and authentication	12 (E1)	5 (E2)	2 (A4, P8)	6 (E3)	8 (E4)	33
	System entry control services	7 (E5)	1 (P8)		1 (P21)	2 (P8, P21)	11
	Audit	6 (E6)	4 (E7)	4 (E8)	13 (E9)	18 (E10)	45
	Access control	1 (P8)	2 (P8, P13)	2 (P8, P13)		3 (E11)	8
	Non-repudiation		1 (P8)		2 (P13, P17)	9 (E12)	12
	Security management	2 (P8, P17)	1 (P8)			4 (E13)	7
	Trusted recovery			1 (P21)	2 (P17, P21)	17 (E14)	20
	Encryption	6 (E15)	4 (E16)	3 (E17)	10 (E18)	4 (E19)	27
	Trusted communication	6 (E20)	6 (E21)	5 (E22)	12 (E23)	10 (E24)	39
Total Counts		40	24	17	46	75	202

Entries with more than 2 patents or applications

E1: A3, A7, A8, A9, P4, P5, P8, P10, P18, P21, P22, P24

E2: P5, P8, P17, P21, P24

E3: P2, P3, P5, P7, P8, P21

E4: A7, A8, P8, P15, P16, P21, P22, P23

E5: A7, A8, P4, P5, P8, P21, P22

E6: A7, A8, P8, P10, P18, P22

E7: A7, A8, P4, P22

E8: A7, A8, P4, P22

E9: A2, A7, A8, P5, P6, P9, P13, P14, P15, P16, P20, P21, P22

E10: A2, A7, A8, P2, P3, P4, P5, P6, P7, P8, P9, P13, P14, P15, P16, P17, P20, P22

E11: P9, P13, P15

E12: P8, P9, P13, P14, P15, P16, P17, P20, P23

E13: P2, P3, P7, P16

E14: A2, A7, A8, P2, P3, P6, P7, P8, P9, P11, P14, P15, P16, P17, P19, P20, P22

E15: A7, A8, P8, P10, P18, P22

E16: A2, P8, P10, P18

E17: P10, P18, P21

E18: A2, P5, P9, P10, P14, P15, P16, P18, P20, P21

E19: A9, P10, P16, P21

E20: A2, A7, A8, P8, P10, P18

E21: A2, P8, P10, P13, P17, P18

E22: A1, P10, P17, P18, P21

E23: A1, P9, P10, P13, P14, P15, P16, P17, P18, P20, P21, P23

E24: A7, A8, P10, P14, P15, P16, P17, P18, P22, P23

- Security Management, in the Determination (0 counts) and Transportation phases (0 counts): we interpret this as an insufficiency in requirements specification, rather than as a defect in design. These forms of security are easily provided by underlying network protocols and operating systems.
- Trusted Recovery in the Contact (0 counts) and Selection phases (0 counts): no secure method is proposed to authenticate, generate and verify integrity check values. We interpret this as another insufficiency in requirements specification, rather than in design.

Our summary matrix (Table 4) can also be used to generate research questions. For instance, as noted above, we found no provision for non-repudiation in the contact phase. Two questions that come to mind are "Would there be any benefit to the server if the client initiated contact in a non-refutable manner?" and "Would there be any benefit to the client if the server initiated contact in a non-refutable manner?" We believe that the answers are "yes", and that enquiry along these lines would lead to improvements in the design, requirements specification, and other documentation for update and patch management systems.

A detailed examination or study of the subjectivity involved in conducting this research may be of value for establishing parameters for future academic research that attempts to draw conclusions about a technological field by examining the patent literature.

6. SUMMARY

In this paper, we have argued that taxonomies can be a valued contribution in the understanding and the reconstruction of system architectures, and that they may be effective in organizing system(s) design documentation. We developed an analytic framework for describing and characterizing update and patch management systems. The framework was developed from a consideration of the systems disclosed in the bodies of thirty-three (33) patents and patent applications. Our analytic framework has the following elements: a generalized update process, a decomposition of the update process into five (5) phases, several alternative methods for accomplishing each phase of the protocol, and a consideration of the security services that may be provided by each phase. We have established that when a taxonomy is combined with industry design specifications (our 5 phase protocol & TOGAF), useful trends may be inferred, and additional research questions may be developed for pursuing the improvement of system architectures [Co03].

REFERENCES

[CE03] CERT Coordination Center Statistics, http://www.cert.org/stats/cert_stats.html, 2003.

[Co99] Common Criteria Management Committee, "Common Criteria for Information Technology Security Evaluation, Part I: Introduction and general model, Version 2.1", August 1999.

[Co03] Colarik, Andrew, "A Secure Patch Management Authority", PhD Thesis, University of Auckland, November 2003.

[Ei98] Eixelsberger et al., "Recovery of Architectural Structure: A Case Study", *Proceedings of Second International ESPRIT ARES Workshop, LNCS 1429*, pp. 89-96, 1998.

[Jo00] Johnson, Samuel, "American Heritage Dictionary of the English Language, Fourth Edition", Houghton Mifflin Company, 2000.

[Ka99] Kazman, Rick, and Carriere, S. Jeromy, "Playing Detective: Reconstructing Software Architecture from Available Evidence", *Automated Software Engineering*, 6, pp.107-138, 1999.

[La94] Landwehr et al., "A Taxonomy of Computer Program Security Flaws", *ACM Computing Surveys*, 26(3), September 1994.

[Op03] Open Group, "The Open Group Architectural Framework Version 7", http://www.opengroup.org/togaf/p3/trm/tx/tx_secur.htm, 2003.

[Pa00] Payne, Christian, "The Role of the Development Process in Operating System Security", *Proceedings of the Third Information Security Workshop, LNCS 1975*, 2000.

[Sa96] Shaw, Mary, and Garlan, David, *Software Architecture: Perspectives on an Emerging Discipline*, Prentice Hall, 1996.

[So92] Sowa, J.F., and Zachman, J.A., "Extending and formalizing the framework for information systems architecture", *IBM Systems Journal*, 31(3), 1992.

Patent Applications

[A1] US 2002/0004402 A1, Suzuki, Naoya, Assignee: None, "Update notification system, update monitoring apparatus, mobile communication terminal, information processing apparatus, contents acquisition instructing method, contents acquiring method, and program storing medium", January 10, 2002.

[A2] US 2002/0016956 A1, Fawcett, Phillip, Assignee: Microsoft Corporation, "Method and system for identifying and obtaining computer software from a remote computer", February 7, 2002.

[A3] US 2002/0016959 A1, Barton et al., Assignee: Network Associates Technology, Inc., "Updating computer files", February 7, 2002.

[A4] US 2002/0100035 A1, Kenyon et al., Assignee: None, "Asynchronous software update", July 25, 2002.

[A5] US 2002/0112230 A1, Scott, C., Assignee: None, "Software update management system with update chronology generator", August 15, 2002.

[A6] US 2002/0184619 A1, Meyerson, M., "Intelligent update agent", December 5, 2002.

[A7] US 2003/0046675 A1, Cheng et al., Assignee: None, "Automatic updating of diverse software products on multiple client computer systems", March 6, 2003.

[A8] US 2003/0046676 A1, Cheng et al., Assignee: None, "Automatic updating of diverse software products on multiple client computer systems", March 6, 2003.

[A9] US 2003/0070087 A1, Gryaznov, D., Assignee: None, "System and method for automatic updating of multiple anti-virus programs", April 10, 2003.

Patents

[P1] US 5,577,244, Killebrew, Alice, and Mann, Charles, Assignee: International Business Machines Corporation, "Methods of applying software modifications", November 19, 1996.
[P2] US 5,586,304, Stupek, Jr. et al., Assignee: Compaq Computer Corporation, "Automatic computer upgrading", December 17, 1996.
[P3] US 5,588,143, Stupek, Jr. et al., Assignee: Compaq Computer Corporation, "Automatic computer upgrading", December 24, 1996.
[P4] US 5,619,716, Nonaka et al., Assignee: Hitachi, Ltd., "Information processing system having a configuration management system for managing the software of the information processing system", April 8, 1997.
[P5] US 5,694,546, Reisman, Richard, Assignee: None, "System for automatic unattended electronic information transport between a server and a client by a vendor provided transport software with a manifest list", December 2, 1997.
[P6] US 5,732,275, Kullick, Steven, and Titus, Diane, Assignee: Apple Computer, Inc., "Method and apparatus for managing and automatically updating software programs", March 24, 1998.
[P7] US 5,809,287, Stupek, Jr. et al., Assignee: Compaq Computer Corporation, "Automatic computer upgrading", September 15, 1998.
[P8] US 5,835,911, Nakagawa, Toru, and Yuji, Takada, Assignee: Fujitsu Limited, "Software distribution and maintenance system and method", November 10, 1998.
[P9] US 5,845,077, Fawcett, Phillip, Assignee: Microsoft Corporation, "Method and system for identifying and obtaining computer software from a remote computer", December 1, 1998.
[P10] US 5,919,247, van Hoff et al., Assignee: Marimba, Inc., "Method for the distribution of code and data updates", July 6, 1999.
[P11] US 5,933,647, Aronberg et al., Assignee: Cognet Corporation, "System and method for software distribution and desktop management in a computer network environment", August 3, 1999.
[P12] US 5,974,454, Apfel et al., Assignee: Microsoft Corporation, "Method and system for installing and updating program module components", October 26, 1999.
[P13] US 5,999,740, Rowley, David John, Assignee: International Computers Limited, "Updating mechanism for software", December 7, 1999.
[P14] US 6,049,671 Slivka, Benjamin, and Webber, Jeffrey, Assignee: Microsoft Corporation, "Method for identifying and obtaining computer software from a network computer", April 11, 2000.
[P15] US 6,073,214, Fawcett, Phillip, Assignee: Microsoft Corporation, "Method and system for identifying and obtaining computer software from a remote computer", June 6, 2000.
[P16] US 6,256,668 B1, Slivka, Benjamin, and Webber, Jeffrey, Assignee: Microsoft Corporation, "Method for identifying and obtaining computer software from a network computer using a tag", July 3, 2001.
[P17] US 6,263,497 B1, Maeda, Tetsuji, and Mori, Toshiya, Assignee: Matsushita Electronic Industrial Co., "Remote maintenance method and remote maintenance apparatus", July 17, 2001.
[P18] US 6,272,536 B1, van Hoff et al., Assignee: Marimba, Inc., "System and method for the distribution of code and data", August 7, 2001.
[P19] US 6,308,061 B1, Criss, Mark, and Cowan, Paul, Assignee: Texlon Corporation, "Wireless software upgrades with version control", October 23, 2001.

[P20] US 6,327,617 B1, Fawcett, Phillip, Assignee: Microsoft Corporation, “Method and system for identifying and obtaining computer software from a remote computer”, December 4, 2001.

[P21] US 6,341,373 B1, Shaw, Robert, Assignee: Liberate Technologies, “Secure data downloading, recovery and upgrading”, January 22, 2002.

[P22] US 6,457,076 B1, Cheng et al., Network Associates Technology, Inc., “System and method for modifying software residing on a client computer that has access to a network”, September 24, 2002.

[P23] US 6,493,871 B1, McGuire et al., Assignee: Microsoft Corporation, “Method and system for downloading updates for software installation”, December 10, 2002.

[P24] WO 0190892, McCaleb, Jed, and Rive, Russell, Assignee: Everdream Inc., “Intelligent patch checker”, November 29, 2001.