

# Telidon Data Restoration: Literary Review\*

Alexandra Tenney<sup>†</sup>  
alexandra.tenney@ucalgary.ca

## 1 TELIDON: A NEW APPROACH TO VIDEOTEX SYSTEM DESIGN

H. G. Brown et al. "Telidon: A New Approach to Videotex System Design". eng. In: *IEEE transactions on consumer electronics* CE-25.3 (1979), pp. 256–268. ISSN: 0098-3063. DOI: <https://doi.org/10.1109/TCE.1979.273222>

### 1.1 Summary and Contribution

Herb Brown, William Sawchuk, and John Storey intend to describe the technical specifications of the Canadian answer to Videotex and how it differs from alternatives of the time. Videotex was a communication protocol that, during the 1970s, had been increasingly popular for interconnected network telecommunications. Multiple countries at the time had used this protocol in conjunction with microcomputers and Canada was exploring its unique version of these computing systems. Brown compares British and French systems, which both use a "mosaic approach" with "fixed format character-oriented systems" or bit-mapping. The Canadian "Picture Description Instructions" (PDI) uses "basic geometric primitives" for more advanced graphics with a similarly sized keypad. Brown details the workings of PDIs, their accessibility, and future Telidon developments and applications.

This article serves as the first public documentation on the technical operations of Telidon. It gives insights into the initial hopes and goals of the Telidon, as well as a snapshot of its creators' mindset. The article shows the context in which Telidon was born, a world where several teletex/videotex systems had already proven to be successful but, in terms of technological capability, were primitive.

### 1.2 Rationale

Both articles by Brown and his team serve as insight into the mindset of the team in charge of developing Telidon. They give historical context to design decisions like PDIs, and decisions to use preexisting technology like television. For my work, this is directly related because I need to understand the design of Telidon to be able to reverse engineer it.

### 1.3 Strengths and Weaknesses

This article explains PDIs, and how they extend to preexisting systems well. It also gives a general overview of how alternative systems of the time function and why Telidon improved upon these competing systems.

Suppose this article is to serve as documentation of the functionality of the Telidon computing system. In that case, it ignores much of the hardware and software beyond PDIs but is essential in the function of the computing system. As readers, we gain no insight into the hardware chosen, the OS used, or how the software was programmed. Regarding the digital artifacts the system outputs

and users make, we need to understand how this interacts with the system.

### 1.4 What can we learn?

The article serves as foundational knowledge for the function of the Telidon, specifically PDIs.

## 2 TELIDON- A REVIEW

H. Brown and W. Sawchuk. "TELIDON - A Review". eng. In: *IEEE communications magazine* 19.1 (1981), pp. 22–28. ISSN: 0163-6804. DOI: <https://doi.org/10.1109/MCOM.1981.1090448>

### 2.1 Summary and Contribution

This is an overview of the Telidon computing system and its technical function to assert its viability as an alternative Videotex system. The authors, worked for the Communications Research Centre (CRC) on the Telidon project.

Brown and Sawchuk compared Canadian Telidon to Minitel and Prestel to demonstrate its superiority as a videotex system. Telidon's success depended on its ability to cater to different consumer needs, making it modular and adaptable. Unlike Minitel and Prestel, which were inflexible and required specific hardware and displays, Telidon was designed to work with various hardware and devices, making it more accessible. Furthermore, Telidon's graphics were universally considered superior, and were backward and forwards compatible. Telidon, which avoids the limitations of other systems while providing similar features, is a viable option for bringing interconnected networks to the public.

Heading the creationist team, Brown is the baseline of Telidon literature, bringing an improved encoding language to the world of Videotex. For us in the future, this article gives a technical overview of the system's functions after it had been improved and how it differed from alternatives of the time.

### 2.2 Rationale

Before restoring data from a Telidon system, it is essential to understand the technical specifications and why design choices were made. This will help frame my research as I go and possibly help to decode specific data found in the Telidon binary dump.

### 2.3 Strengths and Weaknesses

The authors concretely can describe why their system is superior but need to acknowledge the timeline in which the three videotex systems emerged. Many of Telidon's successes stem from the work other systems pioneered, and Telidon's late start allowed improvements to this work. The article also fails to describe concretely how Telidon hardware interacts with the software, which in the case of my work, is pertinent.

\*Course Code: CPSC 503

<sup>†</sup>Supervisor: Dr. John Aycok

## 2.4 What can we learn?

Brown built the Telidon, and I am interested in the actual hardware and micro-computers that Telidon was built to function with.

Ultimately, this work serves as good technical documentation of the design of Telidon. Hopefully, within my work, I can give readers a comprehensive understanding of the workings of Telidon so that, if given a chance, they may reverse engineer their Telidon data.

## 3 SENDING PAGES

D. Godfrey and E. Chang. "Storing Pages". In: *The Telidon Book*. Press Percepic Ltd., 1983. Chap. 7, pp. 179–202

### 3.1 Summary and Contributions

Ernest Chang explains how the Telidon computing system could be configured to send pages- the individual frames of graphical data, from external servers to a user's home. This technical explanation allows other researchers, including Chang himself, who was not on the team at CRC who created Telidon, to give alternative options and insights into the current sending approaches.

Chang outlines two methods of sending Picture Description Instructions (PDIs) to Telidons across Canada: direct interactive and broadcast quasi-interactive. Direct interactive includes Datapac Transmission and Coaxial Cables, with Datapac ensuring raw ASCII interpretation, and Coaxial Cables being a new and untested solution for interacting with Telidon servers. Broadcast quasi-interactive is an alternative, with Telidon servers broadcasting 200-300 pages cyclically. Two modes of broadcast are available: Vertical Banking Interval (VBI) and Full Field, with VBI being preferred, but Full Field allowing access to more Telidon images.

Chang outlines the capabilities and technical specifications of Telidon's ability to send its pages, unlike any other publication.

### 3.2 Rationale

It serves as a good piece of technical documentation and is valuable in preserving Telidon software. I included this work in my literary review because it represents a group of computer enthusiasts when Telidon was gathering interest, who were not involved in making Telidon but were actively working on using the technology.

### 3.3 Strengths and Weaknesses

Chang's work has strength in its clarity and simplistic style. He explains technical topics slowly so the reader can understand how Telidon may have worked.

It can often be difficult to distinguish which parts of his Telidon specifications are recommendations or ideas of how the technology could be used and which were employed at the time.

### 3.4 What can we learn?

Because of this weakness, I would love to ask Chang, or anyone who had significant involvement with Telidon during its peak if the computing system actually used Coaxial cables for sending pages, or if this was simply a suggestion that Chang made. Could users "switch to Telidon mode and get a full interactive connection" with the Telidon servers?

Chang produces an easy-to-read technical specification on specific hardware and software components of the Telidon. I plan to similarly produce a technical specification on the digital artifacts, specifically the binary produced by Telidon and how that may be decoded.

## 4 STORING PAGES

D. Godfrey et al. "Storing Pages". In: *The Telidon Book*. Press Percepic Ltd., 1983. Chap. 8, pp. 179–202

### 4.1 Summary and Contribution

Peter Hitchcock and John Vizslai examine how Telidon functions with a supplementary database and how this database can provide content to users. In this chapter, Hitchcock initially runs through general database theory, explaining databases' basic assumptions and goals. He then outlines the three primary databases: relational, network, and hierarchical. The Telidon system uses hierarchical databases because it aligns with the way that software handles "pages" within the Telidon system. Telidon data must be interpreted as raw ASCII, constraining the use of control characters encoded in unprintable ASCII. The solution was to take a "packet approach," predefining frame sizes where the Telidon data makes up the data section of these packets. Vizslai concedes that Telidon working with a database was a difficult technical challenge and expects many more hurdles as it progresses. This chapter serves as one of the only technical specifications on the Telidon databases I have come across.

### 4.2 Rationale

One of the more considerable hurdles I will have to overcome in reverse engineering the Telidon binary data is how the Telidon was meant to reconstruct pages, specifically through pointers. Reading and reviewing the little documentation on these pointers and how it was constructed was essential to my work and, thus, this literary review.

### 4.3 Strengths and Weaknesses

Hitchcock and Vizslai use many diagrams and visualizations that enhance their technical explanations, especially for something as visual as database design.

Like the previous chapter in this book, it often needs to be clarified which technical design specifications are implemented in working Telidon's.

### 4.4 What can we learn?

I want to employ the same visual format via diagrams and pictures that this chapter did to enhance the readability of my work.

## 5 DESIGN CONSIDERATIONS OF NAPLPS, THE DATA SYNTAX FOR VIDEOTEX AND TELETEX IN NORTH AMERICA

W.H. Ninke. "Design Considerations of NAPLPS, The Data Syntax for VIDEOTEX and TELETEX in North America". eng. In: *Proceedings of the IEEE* 73.4 (1985), pp. 740–753. ISSN: 0018-9219. DOI: <https://doi.org/10.1109/PROC.1985.13197>

## 5.1 Summary and Contribution

North American Presentation Level Protocol Syntax (NAPLPS) was designed as a joint effort between the CRC and American telecommunications company AT&T. William Ninke explores the design of NAPLPS and why particular designs were incorporated that later versions of Telidon employed.

Ninke seeks to defend the new standard by systematically explaining the protocol's goals and a non-exhaustive description of how the protocol functions. Within the goals section, Ninke dictates the way the goals of the protocol influenced and ultimately made decisions that were included in the protocol. In the functions section, Ninke systematically outlines each major function of NAPLPS, described with language that echoes their goals from the previous section, reinforcing his argument on why decisions were made for the protocol.

While NAPLPS already had a 158-page standards document; Ninke explained the broad strokes of the inner workings of NAPLPS and gave insight into the decision-making of a standards committee. Since the official standards document is the property of AT&T, the company which championed the development of NAPLPS, Ninke gives an accessible way to understand the standard.

## 5.2 Rationale

NAPLPS is necessary for significantly more relevant documentation and similar computing systems to Telidon. To understand the structure of the binary dump it is crucial to understand the protocol the system was built to interact with.

## 5.3 Strengths and Weaknesses

As a non-exhaustive documentation source for NAPLPS, this article was well done. The graphics enhance understanding, and each protocol aspect is well explained. I specifically enjoyed the graphics in the protocol explanation, as they enhanced explanations.

This paper misses the social factors and the interplay between nations that likely contributed to choices during the development of NAPLPS. It is much easier to believe and write that each decision was made by pure rationality. However, social factors like power imbalances and compromises likely played a role in why certain parts of NAPLPS were included.

## 5.4 What can we learn?

At the end of the article, Ninke expresses concerns about how they intend to adapt the protocol to fit some unrealized needs of Videotex. Many of these concerns can be traced directly back to the eventual demise of Telidon and likely other videotex systems. Given the fate of all videotex systems, would anything different in the protocol may have changed that. Were videotex systems doomed from the start, destined to be usurped by the World Wide Web, or did their new, standardized protocol seal its fate? If I can fully reverse the Telidon binary dump, I intend to mirror the graphics paired with an explanation of protocol functions in my paper.

## 6 PRESERVATION OF DIGITAL OBJECTS AT THE ARCHAEOLOGY DATA SERVICE

Janet Delve, David Anderson, and Jenny Mitcham. "Preservation of Digital Objects at the Archaeology Data Service". In: *Preserving Complex Digital Objects*. Facet Publishing, 2014. Chap. 5, pp. 49–62

### 6.1 Summary and Contribution

Jenny Mitcham explores the Archeology Data Service (ADS), an organization founded "to preserve digital data [...] and making it available for scholarly re-use". Through several extensive surveys, ADS confirmed that across academia, there was a desire to re-use and preserve large data sets within the public domain. ADS asks the question: What ensures data is archivable in a way?

Mitcham explains ADS' data collection policy based on usefulness and historical significance, as well as documentation of the particular artifact. The author cites two major projects: "The Big Data Project" and VENUS. From The Big Data project, the lesson learned was that complexity, not size, presents archiving challenges. ADS applied this lesson to large data sets within VENUS.

Mitcham determines that "raw data and documentation, therefore, go hand in hand" because it ensures future re-usability and meaning for the data. This principle outlines the ADS archival strategy for complex datasets. Finally, ADS finds that there is "no one-size-fits-all preservation strategy," but "starting a dialog with the digital archive" as soon as possible creates the best conditions for success.

### 6.2 Rationale

The ADS' considerable work conserving and restoring digital works means that the organization has experienced challenges and developed solutions for preserving digital media. The organization also has valuable opinions on developing software and media to benefit preservationists.

### 6.3 Strengths and Weaknesses

This article's recommendations on ensuring technology is archivable are unique and needed in the field.

The ADS primarily focuses on 3D models and visualization data as their medium of digital artifacts, leaving out a plethora of digital media significant to culture and history. Focusing on a narrow data field means many other digital artifacts are left out.

### 6.4 What can we learn?

I would ask the author if they ever foresaw ADS expanding the types of data it was primarily interested in preserving.

Regarding the program I plan to develop, I have redetermined the importance of documentation further than simply for re-use today. Instead, it ensures both the archival survivability of my program and Telidon's.

## 7 DIGITAL PRESERVATION AND CURATION: THE DANGER OF OVERLOOKING SOFTWARE

Janet Delve, David Anderson, and Neil Chu Hong. "Digital Preservation and Curation: The Danger of Overlooking Software". In:

*Preserving Complex Digital Objects*. Facet Publishing, 2014. Chap. 9, pp. 111–124

## 7.1 Summary and Contribution

Neil Chu Hong brings to the surface the chronic habit within the digital preservation space of ignoring software and focusing efforts simply on data preservation. Hong explains, "a key challenge in digital preservation is being able to articulate, and ideally prove, the need for preservation," and that the same is true for software, leaving gaps in what is preserved and how it is preserved.

Hong summarises the framework by assuring that "there is no simple and universally applicable formula for determining if your software needs to be preserved, and how to go about preserving." Instead, the software can be categorized into its four primary preservation purposes. From there, those purposes can be associated with five preservation approaches: technical preservation, emulation, migration, cultivation, and hibernation.

Hong can refocus the field of digital preservation and outline a methodical approach toward when and how that software should be included in the preservation.

## 7.2 Rationale

As Hong points out, much of the current literature focuses on preserving data and artifacts of software, which aligns with much of the Telidon preservationist work previously done. While this means we have ensured some works of Telidon will continue to exist for future generations, it means when new Telidon filesystems or binaries have been found, like in my case, they are left inaccessible. This contribution centers the importance of my own work, and while it is not software preservation in a literal sense, additional software elongates the life and accessibility of Telidon works.

## 7.3 Strengths and Weaknesses

The framework Hong provides through the purposes of preservation and approaches to preservation allows software developers and organizations to diagnose how to preserve their software efficiently.

However, the author contradicts himself with this framework, as there are indeed use cases of software that need to adequately fit into these descriptions of preservation purposes. The article would benefit from a section that indicates steps or questions a researcher can ask when they find their software does not fit into one of these use cases.

## 7.4 What can we learn?

While a few developers are working on developing software to read through Telidon binaries, I intend this software to be available for re-use. I can employ Hong's framework to document and preserve my software.

## 8 VIDEOTEX ART RESTORATION: TECHNICAL AND CONCEPTUAL CHALLENGES

John Durno. "Videotex Art Restoration: Technical and Conceptual Challenges". In: *16th International Conference on Digital Preservation* (2019)

## 8.1 Summary and Contributions

In his poster, John Durno investigates the challenges of restoring Videotex Art. According to the Digital Preservation Coalition, Videotex has become "partially extinct" and "critically endangered." Durno claims the artworks he is trying to restore are "important early examples of our emergent digital culture", and thus hold value to attempt to restore for those interested in Canadian culture, art, and the digital revolution. Durno notes Telidon 699's lack of documentation and compatibility issues, making emulation difficult. NAPLPS, which has standardized documentation, poses fewer problems. Telidon 699 was mainly for development, necessitating more hardware and documentation for Telidon artwork.

Durno documents the challenges of restoration work precisely encountered in his work with the Canadian Telidon. However, the challenges apply to those who attempt to modernize legacy media. Durno's challenges are most relevant to other Videotex/Teletex Systems, like Prestel and Minitel, but can also be applied to similar media of the time.

## 8.2 Rationale

My work fits into Durno's initial work, so it is essential to understand the technical challenges researchers have faced in restoring other Telidon works. While I will not have to account for the artist's intentions, I will need more documentation on the protocols and a lack of physical hardware.

## 8.3 Strengths and Weaknesses

The goals and approach of the article are clear, and the language is easy to read. This allows those interested in Videotex art but who need more technical skills to get excited and support restoration efforts.

Durno presents the challenges to Videotex art restoration but not solutions, making it hard to support the restoration project.

## 8.4 What can we learn?

How much Videotex art needs to be preserved, and how? For the question of *how much* does this work need to continue until all remaining Telidon filesystems have been restored and archived? For the question of *how* does that preservation need to maintain its original form, running on a Telidon? Or can it be updated to modern formats and maintain its original form?

Durno has encouraged me to keep track of the major challenges I face, and I continue my research and write about them in my final report.

## 9 WHERE ONLINE SERVICES GO WHEN THEY DIE

Benj Edwards. *Where Online Services Go When They Die*. June 2014. URL: <https://www.theatlantic.com/technology/archive/2014/07/where-online-services-go-when-they-die/374099/>

## 9.1 Summary and Contributions

Benj Edwards documents Jim Carpenter's attempt and subsequent success to reverse engineer Prodigy. Once Prodigy abandoned its

non-profitable project, a digital era was left "trapped behind a technological minefield of obsolete storage formats, protocols, programming languages, and computer systems." Missing the graphics and motivated to collect screenshots of familiar screens, Carpenter drove through his old filesystem, hoping for images.

Reverse engineering Prodigy's graphical displays will require more time and samples than just Carpenter's old installation. However, by searching through his old filesystem, Carpenter found a series of cached graphics from his last Prodigy connection, allowing him to extract some images. Although Carpenter has not yet fully restored Prodigy or extracted all images, his work demonstrated that the graphics were still accessible and inspired him to continue.

When an "online service disappears, a piece of our civilization's cultural fabric goes with it." Carpenters' work and Edwards' recount demonstrate the ability to recover these online services through reverse engineering. For technologies based on NAPLPS, like Telidon, Carpenter's work can serve as an example of how reverse engineering and minor programs can revitalize these technologies and restore a crucial cultural moment in our history.

## 9.2 Rationale

Carpenter's strategy in reverse engineering the Prodigy files is the same as mine within Telidon and can be used as a reference for my work. Carpenter claimed that even during Prodigy's peak, he never really liked the technology, "but [...] still used Prodigy every single day. It was the graphics." Both Prodigy and Telidon, in their later years, used the same graphics, as they were produced using NAPLPS, a protocol that was jointly created with the Telidon creators.

## 9.3 Strengths and Weaknesses

Edwards strongly argues for the need to preserve our digital heritage despite the companies that develop it. This argument dictates to academics interested in legacy computing systems and digital culture why his work is essential and why others should continue to invest in this field. He also can prove, through the success of Carpenter, that it is possible to do this without the support of the business that builds these products.

While I understand the importance of finding the STAGE.DAT file within the Prodigy installation, it is unclear how Carpenter extracted images from it, especially since the article emphasizes how Prodigy did not use "vanilla" NAPLPS. For my research, I would have liked the description of reverse engineering to have been more verbose. I also would have liked to hear more about the trials of reversing the initial filesystem and clues that led Carpenter to believe he was on the correct path.

## 9.4 What can we learn?

Given a chance to talk to both Carpenter and Edwards, I would like to see their progress in restoring Prodigy since writing this article. I would also like to ask Carpenter about his reverse engineering methodology.

Methods of reverse engineering were employed on a NAPLPS-based system. The Prodigy Restoration Project, which Carpenter's work eventually spiraled into, is the closest related work to my

Telidon Data Recovery. From this article, I can learn how to celebrate little wins as I uncover files within the Telidon binary dump, representing being on the right track. I also look into Carpenter's Python scripts and use them to inspire my scripts.

## 10 HABITAT FOR HUMANITY: HOW A CLASSIC MMO GOT A SECOND LIFE

Ash Parrish. *Habitat for Humanity: How a Classic MMO got a Second Life*. Apr. 2022. URL: <https://www.theverge.com/23025168/habitat-for-humanity-mmo-game>

### 10.1 Summary and Contributions

Ash Parrish examines what challenges and solutions were involved in Alex Handy and his organization the Museum of Art and Digital Entertainment (MADE) restoration of the Commodore 64 video game: Habitat. Habitat is touted as the first Massively Multiplayer Online (MMO) game built in a graphical Multi-User Dungeon (MUD) style. Newer games resemble Habitat, signifying that the game created a genre and altered the culture around video games to reflect online communities. Parrish demonstrates how through luck, hard work, and "guts," teams can restore these digital relics.

Handy joins a handful of other ad hoc hobbyists in documenting the restoration and preservation work of cultural artifacts at the beginning of the digital era. Like Edwards and Carpenter, Handy is an enthusiast rather than an academic and is a shining example of success in preserving digital heritage.

### 10.2 Rationale

The challenges Habitat restorationists face are similar to those I face in my own research. Habitat was built to be compatible with proprietary hardware and OS'. All of the technology was legacy, and little documentation was kept. These challenges echo those of Telidon, and thus I can learn from the Habitat restoration project's successes and failures.

### 10.3 Strengths and Weaknesses

The author understands the challenges that stem from digital restoration, proprietary software, missing/legacy hardware, and lack of documentation. It also helps understand the amount of ad hoc work hobbyists are willing to put in to bring these digital relics back to life. Finally, this article's strength lies in the number of people it credits for this restoration and how it helps the reader understand the teamwork and collaboration that goes into a restoration.

Having source code is a tremendous advantage to restoring old software. Reverse engineering becomes more straightforward when one has to understand and pattern-match code rather than determining which unprintable characters are relevant. Habitat additionally had over 70 developers working to restore the project, staffing that this project does not have. Ultimately, while Habitat's successes exemplify what is possible with old technology, it is not a cookie-cutter example for all restoration projects.

### 10.4 What can we learn?

I would like to ask the author what has been done differently in terms of documentation in the Habitat restoration project that

ensures this same problem will not be encountered in the future and to ensure the game has been preserved for the long term.

We can learn from this work that a large community wishes to have old, pre-internet digital works restored into the public domain. Those works created culture and community, and they can be experienced once again.

## 11 REVERSE ENGINEERING OF LEGACY SYSTEMS: A PATH TOWARD SUCCESS

Alex Quilici. "Reverse Engineering of Legacy Systems: a Path Toward Success". eng. In: *1995 17th International Conference on Software Engineering*. ICSE '95. ACM, 1995, pp. 333–336. ISBN: 9780897917087. DOI: <https://doi.org/10.1145/225014.225046>

### 11.1 Summary and Contributions

Alex Quilici aims to determine if reverse engineering of legacy systems can help engineers understand legacy software despite hardware and programming language obsolescence. Reverse engineering, which produces specifications to guide developers toward an authentic replica of old systems in modernized formats, has a rigid definition of success. Quilici modifies this definition to make it "both attainable and worthwhile" and answers two questions about automatic reverse engineering programs: "can they extract complete designs from legacy software?" and "do they scale?" While the answer to the questions is: no, reverse engineering can still extract partial solutions and determine program functions through cooperative extraction, leading Quilici to conclude that legacy systems are "eventually doomed to success, not failure."

### 11.2 Rationale

This paper helps frame my research in two ways: the actual work in reverse engineering I plan to do and how my work will influence and advance future efforts within Telidon restoration and recovery.

### 11.3 Strengths and Weaknesses

The author provided practical answers to their question, offering guidance applicable to researchers and industry experts.

Quilici defines reverse engineering as a completely automatic process. I cannot entirely agree with this definition, as much reverse engineering can be done as a manual "tracing" effort through source code. In my case, at the beginning stages of my research, I am not even tracing through code but rather a binary, compiled version of code or data. While this process may eventually evolve to be automatic through a program, the initial part still falls within my definition of reverse engineering.

### 11.4 What can we learn?

Quilici asserts that it is possible to "extract partial specifications of [...] existing systems" as a part of an automatic reverse engineering program. This assertion bodes well for my work, as a program that can partially reverse Telidon files from a binary would mean more recovered files. Quilici's second assertion, that in practice, these programs scale well, is a positive promise for the future of my work as a tool for other researchers.

## 12 MODEL-DRIVEN SOFTWARE MIGRATION: A METHODOLOGY

Christian. Wagner. *Model-Driven Software Migration: A Methodology Reengineering, Recovery and Modernization of Legacy Systems*. eng. 1st ed. 2014. Wiesbaden: Springer Fachmedien Wiesbaden, 2014. ISBN: 3-658-05270-8. DOI: <https://doi.org/10.1007/978-3-658-05270-6>

### 12.1 Summary and Contribution

Christian Wagner hopes to broadly answer the question that plagues technical debt for many software engineers: how can organizations effectively handle re-engineering legacy computing systems? He dictates software re-engineering as a three-step process, extracting (reverse engineering), transformation, and forward engineering. I am focused on his explanation and assertions of reverse engineering for the literary review since it most closely matches my research. He defines reverse engineering as "the extraction and the representation of information at a higher level of abstraction" and, as the title of the book eludes, asserts it is model-driven. This reverse engineering will likely include source code, but with legacy systems, it will include all artifacts created and used by the software.

The engineer or program doing the reverse engineering tasks must derive a model of the legacy software from driving the re-engineering process. Reverse engineering, a technique mainly in the sub-discipline of information security, is used to exploit programs and understand malware. Wagner, however, re-applies the methods to software engineering and modernizing legacy computing systems.

### 12.2 Rationale

Reverse engineering as a technique is the basis of my research because "a software system can only be changed when it is understood." If I wish to reconstruct Telidon files, it is helpful to understand others who have employed reverse engineering as a step in updating legacy systems.

### 12.3 Strengths and Weaknesses

Wagner nicely integrates reverse engineering techniques with other software engineering processes to move the process forward from simply knowledge gathering. Instead, the re-engineering step within the process creates actual action.

Wagner's axioms state legacy systems must drive profit to be worth saving, so Telidon wouldn't apply. But Telidon is a legacy system that's valuable to digital history and art recovery, even if profit-driven criteria are not met.

### 12.4 What can we learn?

Wagner focused primarily on source code as the foundation for reverse engineering, and I would like to know the author's thoughts on legacy systems without source code. How can we effectively automate a reverse engineering program for these binaries?

Wagner's book helped me place my research into a different field of literature: software engineering and re-engineering. These were fields that I needed to familiarize myself with. However, my work with a legacy Telidon system fits nicely and can serve as an example to others who wish to update legacy systems.