Generic Models and Document Tools for Media Production and Data Management in Networked Servers

Simo Karttunen* and Heikki Nikulin*

Keywords: Production, data management, workflow, tracking, network, WWW, client/server, intranet, Java, Java Beans, media objects, distributed components, prepress, design, compound documents

ABSTRACT

Media production takes place in the networked server environment. Many servers have special features or roles and they are highly dedicated. This is the case especially in printed media production, prepress and press frontends. In the Internet and in other web media (e.g. intranet, extranet and web broadcasting), the servers used are more standard and open. They are called web servers which consist of certain software and tools, and are run on the basic server platforms, e.g. Windows NT and Unix. Most production processes handle digital and formatted data. These publishing systems, processes, products, sessions and jobs are data sources for production management, e.g. the metadata of content files, and process information, the events and states of the jobs′ elements, often located in several companies. By using OpenDoc or Java Bean components, object-oriented trackers and Java applets/applications, in an intranet messaging infrastructure, it is possible to exchange production data and to collect it to be parsed and used in PMS databases. Models of and experiments with these proposed component-based print or web publishing solutions are reported.

Introduction

In the development of media systems we are entering a new phase which could be called the period of web-centric and dynamic document systems. Information processes and publishing media have become very complex and time-critical. There are a large number and variety of image, sound, video and multimedia formats, objects and components for the information content and packages.

* **Simo Karttunen,** Professor DTech and **Heikki Nikulin** BSc,
VTT Information Technology, Finland: Phone+3589 4561, fax +3589 455 2839.
E-mail: Simo.Karttunen@vtt.fi and Heikki.Nikulin@vtt.fi

Therefore, tracking, messaging and control of the processes and their servcers, subsystems and applications are logical steps from **isles** of autonomous systems, and manually-oriented batchwise jobbing to networked, highly ayutomated and controlled **media workflow** systems. At the same time we may separate at least four integration trends in the media production:

• Linking between the production **processes** and servers for tracking and intersystem control or for data management and PMS [1, 2, 8, 15]. In the media production, formatted files or job components should flow through a network, i.e. LANs, servers and intranet and they need data management.

• Linking between the **archipelagos** of business, marketing and administration systems and the systems of the production chain. There are more open solutions in the software to solve such linkings.

• General data processing, normally client/server, systems, are integrated into **companywide** solutions starting with e-mail, groupware, software distribution, through middleware suites to repositories, datawarehouses, datamarts, online analytic processing (OLAP), data security, and global PMS services.

• Networking as a real business backbone using intranet or extranet software and **distributed component-based** frameworks and business objects, e.g. Java and OpenDoc components and the entire CORBA architecture [1, 3, 4, 8]. The CORBA approach uses components to handle the business information instead of more ad hoc linking, scripting and client/server methods (the two previous trends listed above). This is the gateway of media production to object-oriented data management, content and network-centric processing.

The fourth trend in this listing is the most recent and requires special attention. The models and tools presented in this paper are proposals for **more open**, component-based systems [3, 4], and for **content-, document- and network-centric** computing, products and media business ideas. Today, the web media still have to search for their role as profitmakers while the publishers of printed media integrate with other media and defend themselves by developing their products and processes. Former print publishing companies are today multimedia and crossmedia publishers using their proprietary information content for products and services also in the web media.

Basic concepts

We define here some basic concepts and refer to previous definitions [1, 3, 4]. Today, the production is almost entirely digital. The resources used to collect the information content and to perform the editorial packaging of the content into distributable media products are still very expert-driven. Publishing has to be restructured and redefined. At the same time, publishing is just another service or production subject to normal rules and business practices. Many generic technologies and tools would help, given the chance.

The closest partners and rivals of the media in the information sector, i.e. the software production and telecommunication systems, have similar problems in their production processes, and the component-based systems will bring more openness and code persistence [3, 4, 8]. These industries have many similar processes, business objects and even some overlapping markets. For instance the software contains many printed and web manuals, not only the program code. Documents and documentation are only one of the many similarities. Let us now define some basic concepts:

Publishing	The design and distribution of documents in media for public or targeted distribution in networks or as printed products.
Media	The main alternatives in the distribution of documents such as Internet/WWW, intranet, extranet, TV, video, print and other recorded media like CD-ROM, video, CD and DVD
Document	Formatted information in static formats or dynamic document frameworks where documents may contain dynamic elements, e.g. the components in the OpenDoc framework.
Framework	An entity or set of interoperable components in the distributed **component** environments, e.g. in the CORBA architecture. In object terms, frameworks are functional preassembled class libraries of components. OpenDoc is a typical example.
Container	A "start-document" opened in a design session to create a proprietary compound information file (a dynamic document). Many types of content can be included, they appear with their tools and are "in-place-editable". Container technologies are a wide spectrum of components which make the compound and dynamic files possible and provide for the interoperation of the components during document-centric design sessions.
Component	Mostly objects added with validated interfaces to **interoperate** with each other and with the CORBA frameworks. Java Beans and OLE components have less features than the OpenDoc.
CORBA	An architecture and set of standards for distributed components, objects broker (ORB), and its protocols which enable a high degree of reusable code, openness for new components and the required interface solutions for the whole CORBA-compliant system market.
OpenDoc	Thefirst document framework (1996-97) for the CORBA world: OpenDoc **components** are validated interoperable components and commercial products [3, 7, 8] made by numerous vendors. OpenDoc losed its position as the document framework of OMG and CORBA. JavaBeans have been proposed to attain a similar position in the CORBA world [4].
Network	Standard Internet communication protocols enable high-level networking, e.g. intranets or web broadcasting, to meet the needs of public or private networks run on the existing basic network services, such as the FTP and WWW (on TCP/IP).
Web media	A large variety of media and services which are based on the

high-level networking, today WWW, and on mass or target addressing. They may contain multimedia or print linking. The object-powered network is called **Object Web** [4].

Printed media	The well-known mass communication products (book, ad-print and newspaper) which are physically distributed, and new web-distributed, on-demand and custom-printed documents.
Intranet	A network tool set for private, secure and configurable server networks using Internet standards. It may be extended to any web servers of other companies.Then it is called the **extranet**.

This list of working definitions leads us to **discuss publishing** from different and wider angles, including the traditional print publisher, software publisher, integrator-user, web site builder, web server and browser maker, web publisher, middleware vendor, advertiser, information provider, content producer, network operator, TV and film producers, and us, the information and system users. The list points out similarities between the media, especially between printed and web media, once rather separate and now intermixed and integrated.

The change is in the **focus**: From data, separate systems, dedicated software, procedural code and computers to network- and document-centric information processing in **sessions** and to more productive man-machine interfaces. We try to use these redefined concepts in the following chapters to build new process models. We refer to new tools available for media production and workflow.

That these changes seem so obvious is because of state of art of the socalled procedural code and mass distribution software, the **shrink-wrap** packages. Though inevitably **functional and necessary** in our production, this DTP software causes serious problems by pushing new versions - and also new "me-too-products" - to the users who must handle tens of formats and support all the latest packages and their new versions. These problems have been discussed - but not enough - in the literature [1, 3, 4, 8]. The fourth trend in the above list, i.e. the distributed components, will bring a gradual and elegant solution to these problems.

Intranet and server tools

High-level networking is as young as WWW, about four years. The web browsers, i.e. the WWW clients, imply a new standard for user interfaces. As the documents are increasing and the respective **web sites** use many types of formats, databases and dynamic documents, we have a complex and time-critical system environment. Also the printed products have shorter lifecycles and contain more variable information for new markets such as digital printing, short-run colour printing, and "distribute-and-then-print".

In print publishing many process stages need servers and physical stores for the product elements, i.e. images, text, ads, pages, impositions, films, plates, inserts and copies (the four last-mentioned in fact can be stored both as data and as physical entities). We may not actually need web servers but there is no big difference between normal platforms and **servers,** compared with **web servers**. In fact, web servers are improved servers adapted to launch intranets and

Internet web sites. It is obvious how useful the 'web server benefits', i.e. mainly the Internet standards, are in relation to the capture and messaging of **metadata** - from the servers automatically - and to the use of intranets as a **workflow** infrastructure.

In a recent paper [8] we reported on a prototype Tracker Server, based on the socalled **Java application**, which is capable of capturing time data from any intranet-connected web servers. This cheap and easy-to-implement solution can be applied to build light systems for tracking in intranet and web media server environments. Compared with the more fundamental approach presented in the next chapters, the **Tracker Server** leads us to more immediate applications and integration links. When using Java applets and Java applications, like the Tracker Server does, we have the possibility to progress to Java Beans and if needed, to the validated OpenDoc components [3, 4, 7, 8].

The **Java technologies** are an open tool et to build objects and components [4]. Java was started with Java language - portable code - but it is nowadays a mobile object model and operating system. There are also other intranet and web server technologies. A very important category are the security solutions which, in fact, separate the less sensitive Internet web sites from the virtual private networking (i.e. intranets and extranets) with firewalls and a secured access. Then, various security methods, such as authentication, digital signatures, access control, virus detection, encryption and digital are the potential solutions. Contrary to common belief dedicated private networks were not very secure. In this paper we mention but do not analyse the broad issue of web security technologies.

Databases are powerful resources in media production. Many servers include databases and a database may be distributed or replicated on several servers. The relational SQL database is a common systematic method to store organised data for applications and also for the PMS, production management systems, as discussed before [1, 2, 5, 6, 9, 15]. In the intranet applications we have to support access to many types of databases. Therefore we have a wide variety of tools for the multi-database needs [3, 4, 10] and for specific web databases [10].

Intranet , Java technologies and databases are the foundation of the socalled **push** technologies, more specifically **web broadcasting**. Several vendors [11] have new products which are to distribute information through channels from a well-managed host server (transmitter) to large number of private customers with a specific tuner to open and use their subscribed channels. Observe the use terms of broadcast terms, "channels and tuners" in this context. The term narrowcast has also been used to point out the possible narrow target groups.

This is a good example of a total solution for business and workflow set up as an intranet (or Internet) **service provider** concept. Our first push systems, i.e. intranet broadcasting applications are under development [13]. Publishers always look for new information needs and many target groups might be reached and served by this two-direction web technology.

Here we start a discussion and definition of **component-based** media data management, **CMDM**. The component, as in **Fig 1**, was defined as a more specific software term in the above chapter Basic Concepts. Components imply improved openness compared with the plain objects, especially interoperability. They are self-describing and autonomous [3, 4]. Media data, editorial packages and document handling software will be reformed into objects and components as reported recently [3, 4, 8], while existing DTP will also be used.

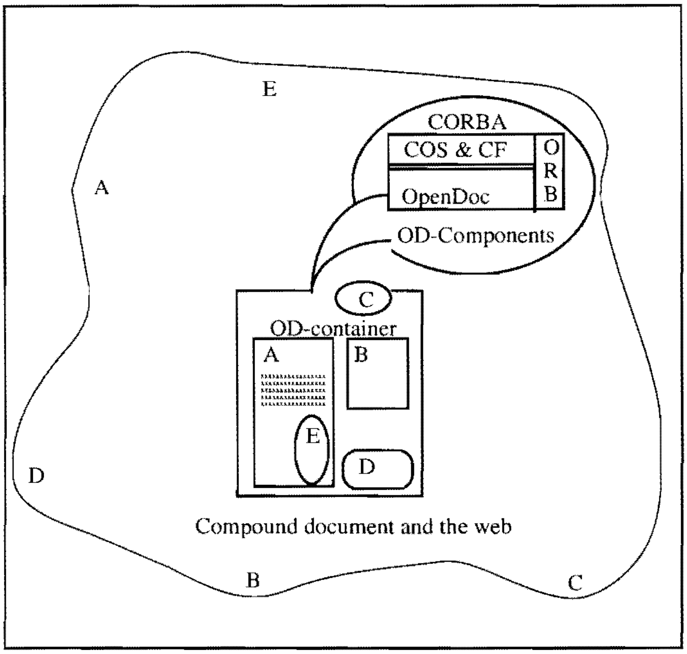


Fig 1 A component-based publishing system uses containers. A compound document, e. g. an OD-container may be opened for design and distribution, and contains parts (OD-components) which are located anywhere in the universal web environment. It uses the CORBA support, i.e. ORBs, the Common Object services, COS, the Common Facilities, CF which mean typical component benefits such as the interoperability. In this example we think components (A-E) as some visual content holders.

The component-based media data management, **CMDM**, is simply a new concept of handling media components, instead of objects, data or files, and improving the preplanning, the design, the packaging, the marketing and the distribution of media services or products. The real daily value of CMDM will reveal itself gradually when the tools for the components and interoperability and component technologies [4, 8, 16] become more common.

Instead of tracking one server, and one application per server, as we did in our experiments [8], we can track **many** web servers, documents and components.

The documents, databases and components are distributed and this distribution is within our control. The publisher and his networking partners always define where and how certain documents or databases are produced and updated, and what **distribution** policy and machinery (e.g. www web sites, intranet, portable or compound documents, prints or broadcasting) will be used to serve the clients and the **end-user**, the intranet procerss staff or the reading and listening people.

The management of media data has been a problem ever since the digitalization of media production in the late 1980s. Nobody paid much attention to the fact that the loose files of media contents are so numerous and pass so many production stages before the distribution [8, 9, 15]. Therefore is a clear need to introduce new concepts and models to define the component-based media data as well as to manage such data and jobs (by tracker components and ORBs) from their native files to interoperable self-managing components, such as compound documents and distributed services.

The components are defined as data and tools combined in an autonomous entity to provide specific system services. We try to show how media production may be timed and coordinated by the **CMDM** and by related production management and messaging concepts, of which some generic examples are ready or under development among the CORBA common facilities and common system services [3, 4, 16].

In **Fig 1** we have some obvious building blocks of a component- and network-centric publishing environment. The compound document, e.g. implies that we can start a preplanning session at a workstation, and begin to design our service by using any other components available through the existing and updated stores of content information which we will use to form the document. Containers may call and insert components and other containers. This dynamically updated master document can announce separate freezed versions, e.g. for copy and sale.

Existing building blocks - also stored as components - imply that we are not starting from scratch but, e.g. from well-planned document and page templates or from earlier model products. During the design sessions we do not need to jump from one application to another or transfer the jobs from one workstation to another - though we may do it. We just call and open interfaces of other components and place them in the visual container for in-place editing.

To produce versions with an increasing complexity and capacity or with variable information blocks is a more immediate process. Various types of information can be included and **bound very late** in the new deliveries. Products or services may be branched flexibly by using variable-data objects, multimedia formats, hyperlinks, portable documents, or by media junctions, e.g. for different purposes, or for custom-built and targeted services.

An example outside the publishing business is given by Orfali et alii [3] in Chapter 19 of [3] which briefly reviews the development and product family of Newi, a framework and tool set for late-bound business objects [16]. We value the flexibility of the business object concept as it separates the **views** (i.e. how things look like) from the business object **models** and the interface issues.

We list eight principles of media planning and design which are relevant in the **preplanning sessions** and **design stages** of document-centric production. Some of them are new, while others have been discussed before.

1. Preplanning should start with structural **descriptions**, by using content structure tools, such as SGML/DTD, or the new XML, which leave the items, details and the physical design open but capture the structures in the content. The structure may be less important in some cases.

2. This structure must be used as **early** as possible not only to start the design stages but to support cost estimates, schedules, virtual products and the reservation of the resources and processes for the later versions, production and distribution of the document. Most later stages need this information.

3. Preplanning has a much higher **position** when the products are more dependent, related, time-critical and reuse-oriented. The dependence implies foreign and commercial components with restraints of price and availability. The more proprietary components should have a defined unit cost and internal price. There are no distinct boundaries between the preplanning and design.

4. The design stages must be able to produce **visual** and distributed miniatures, prototypes or virtual models of the final service. Some existing OpenDoc components already fulfil this requirement, as discussed later and shown in **Appendices 1 and 2.** These samples, proofs or virtual products may be useful in marketing, sales, customer service, catalogs, user interfaces and brochures.

5. The distributed products may be **dynamic** (e.g. a web site with hyper-linked containers, using components) or **static,** i.e. freezed, formatted versions of the document, e.g. its printed copies. They must be useful and commercial. Their users may be given special rights to use and even own. Products may also be mirrored from the server into the channels, opened and read by the clients using tuners as in the case of push products [11].

6. The client documents or unauthorized actions should not jepardize the **integrity of the master documents** (e.g. in a database), but they may have to be distributed or transmitted in a specific manner. To update and change the contents, there must be version control. To store you have to decide for how long, and in which format, to save and to archive or to delete the objects.

7. The **networking** used for document production must allow for segments of proprietary, virtual local area networks with secured servers, access control, authentic(ated) clients and information sources, and encrypted contents. Many socalled firewalls have a dual role: they are used to protect the content and resources while adaptively allow and manage the traffic from and to the servers of the media publishing centre.

8. The tracking of production is by definition component-based. We propose the use of JavaBeans or OpenDoc components. In practice this depends on the used object model (OLE/DCOM, OD/DSOM or Java Beans) needed for the interoperability of the components. How the **Tracker Java Beans** capture the timing data from the components, servers and applications is a different matter, and a later CMDM topic [17].

Lower-level **coexistence** and **interoperability** between the Java Beans and OpenDoc components and even with the OLE/ActiveX will be needed as pointed out [1, 3, 4, 8]. A lot of existing data and methods are ready for use. **Microsoft** gave a recent announcement (April 1997) to port its "industry standard" application suite, by breaking each application into a suite of components, and to join the OpenDoc component family in a manner similar to what Apple had announced before (for OLE/ActiveX).

This list is not complete. We have to update our needs when the basic principles change and new tools allow unheared-of solutions. The flexibility, scaleability and dynamic behaviour of the components and ORBs are the main benefits when the component suites for the media production workflow are designed by present and future publishing system vendors.

Object Web and Publishing

In recent literature [3, 4, 16], the objects and their interoperable derivatives, the components, become the smallest building blocks of the system design. The components, as well as their larger sets, such as application frameworks and business objects [16], can be understood as entities and packages, or deliveries of work, data and distribution to the users. If you compare this aspect to the above definition of publishing, you find a close similarity. We may state that component-based media data management, **CMDM**, is a set of component tools to take the publishing business to the future of modern business objects and frameworks.

Fig 2 shows some **hypothetical components** for this redefined Object Web publishing in which there is no distinct boundary between the print and the various web media or between the various media formats. IIOP and ORBs are used as the open architecture to integrate potential "new publishing suites or applications" to other CORBA-compliant frameworks or business objects.

In **Fig 2** we have deliberately omitted the lines to connect the boxes. The ORBs and frameworks are distributed, and they run on a set of **web servers** which are a part of the company VLAN, i.e. on the same hardware platforms as the existing applications and operating systems. Observe that the CORBA ORBs and the common services and facilities do not centralize anything else but the interface machinery, i.e. tools for the component interplay [3, 4, 16]. These will be sold as a central part of the compound document and business object suites.

Hence the web servers form the **virtual networks** of the Object Web with their public Internet and private extranet/intranet segments. The Internet connectivity is a superset above the companies' proprietary and virtual TCP/IP-based LANs.

The **component tools**, more closely presented in references [3 and 4] which describe the CORBA 2.0 ORB, e.g. the Interface Repository APIs, IDL stubs, Dynamic Invocation Interface (DII), Dynamic Skeleton Interface (DSI), Basic Object Adapter (BOA), Implementation Repository, Common Services and Facilities (component sets for generic services supporting the ORB) and the ORB Interface, will be the interoperation grounds of the media components and documents. The IIOP and the ESIOP may be used in large environments to combine CORBA ORBs with proprietary ORBs [4].

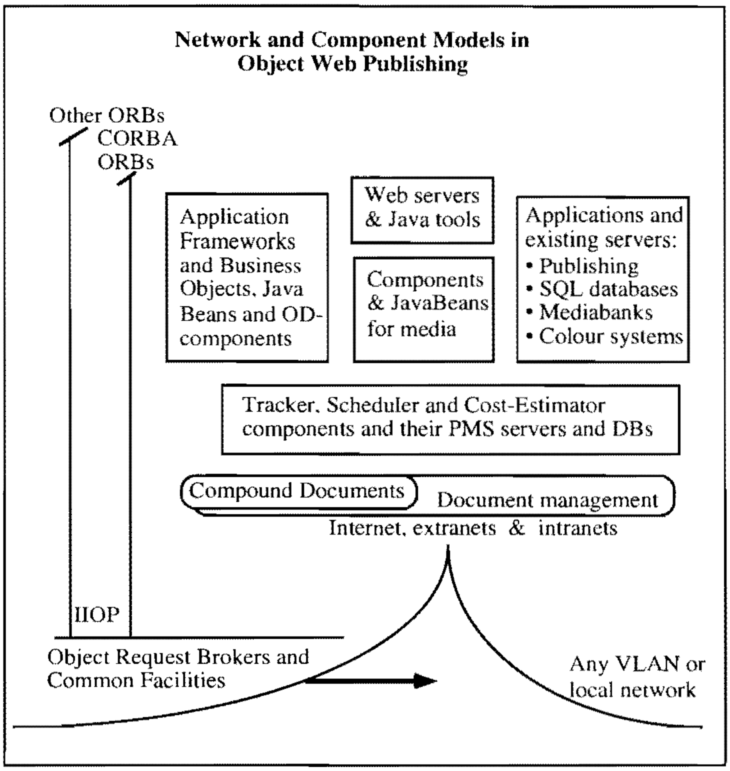

Network and Component Models in Object Web Publishing
Other ORBs
CORBA ORBs
Application Frameworks and Business Objects, Java Beans and OD-components
Web servers & Java tools
Components & JavaBeans for media
Applications and existing servers:
• Publishing
• SQL databases
• Mediabanks
• Colour systems
Tracker, Scheduler and Cost-Estimator components and their PMS servers and DBs
Compound Documents
Document management
Internet, extranets & intranets
IIOP
Object Request Brokers and Common Facilities
Any VLAN or local network

Fig 2 Object Web is a general trend which brings the object tools in the web media (i.e. Internet, extranets and intranets) and has a direct impact on the document management and publishing concepts of the coming years. IIOP, ORBs and the component models bring the data management on a higher level of integration and abstraction. Software, DTP, and data, e.g. in the existing SQL DBs, may be reformed to components which mean code- and vendor-neutral component-based systems, suites, frameworks and business objects. Distributed components support virtual production.

The components as entities fall into a few categories depending on the point of view: a) as integration entities e.g. existing applications, suites, frameworks and business objects, or b) media components and c) metadata components:

• **Existing applications**, such as DTP, SQL databases, mediabanks and colour systems will be "IDL-ized and Java Brewed" to act as interoperable components and suites, providing a new market for software products. More and more often they run on web servers. The wave-formed baseline in **Fig 2** indicates the increasing share of the component-based CORBA world as the less open systems vanish.

• **Media components** and their frameworks (i.e. preassembled component suites) are marketed already and they are based on existing Java Beans and OpenDoc components which seem promising, as reported earlier [7, 8]. In this report we update these examples in **Appendices 1 and 2.**

• **Business objects** and application frameworks for other than media businesses have media-related **views** as their basic client features, e.g. for design display, documentation or distribution of e.g. cars, machines or other real-life objects, modelled and managed by components and business object models. Media exposure and advertising are needed also in the future business. The publishers of professional media will get more open and portable **component-based** information from their ad customers, e.g. for the local car distributor not in static text-files, prints or images but in the form of live, late-bound visual containers, or other components shared with the car factory, its international distributors and local dealers.

• **PMS** servers and databases will operate using mainly process **metadata.** They will appear as component suites combining product models, workflow, tracking, scheduling and cost-estimation. These may inherit generic features from the common services and facilities of the CORBA world. How this happens is anticipated in some detail in reference [4], starting with thenaming conventions and life-cycle features.

• Document management uses component-based **compound documents**, e.g. OpenDoc (or OLE or later some new Java container technology) for visual design and prototyping, for language-independent basic coding and for limitless web distribution, i.e. for full openness required in the network.

We may call this picture, presented roughly in **Fig 2**, a first-version general model for **CMDM**, component-based media data management. So far this model describes, sorry to admit, on a fairly high level of abstraction how the components and object buses, like ORBs, behave and interoperate in media production, in the future Object Web, among the global CORBA-compliant web servers and the network of the distributed objects.

Tracking and PMS Components

To describe this new media production trend we have used partly terms of the client/server and database period - in which we still live - and also the central concepts of the **intranet** technologies and the CORBA **object world**. We try to bring this description closer to reality. **Fig 2** was redrawn into **Fig 3** - with small ellipses presenting for the mobile PMS components. This helps us to comprehend the role of basic PMS functions.

They must work as distributed **components** as well as the respective media objects, components and servers which they are tracking and managing. They must interoperate with the CORBA world (**Fig 3**) and use its intranet and component benefits for product modelling, naming, life-cycle, workflow and messaging. We do this by giving some brief examples of existing early demo products of the OpenDoc and Java Beans families, in **Appendices 1 and 2**.

Fig 3 points out that the PMS components may register process events on all servers of a workflow (e.g. an intranet-based production line) and might be connected also to the global PMS or to the administration databases. If the workflow would be extended to cover the customers' servers (extranet) the firewalls between the companies would allow PMS messaging.

The messages of metadata, e.g. the tracking formats, will be **components as well**. They are already objects [1, 2, 8, 9 and 15] shown by some early examples like IFRAtrack and CIP3 [15]. When designed as CORBA components they may inherit some behaviour from the Common Object Services as mentioned by Orfali et alii [3, 4]. PMS components combine the formatting and messaging without any need to install permanent software items into servers and processes.

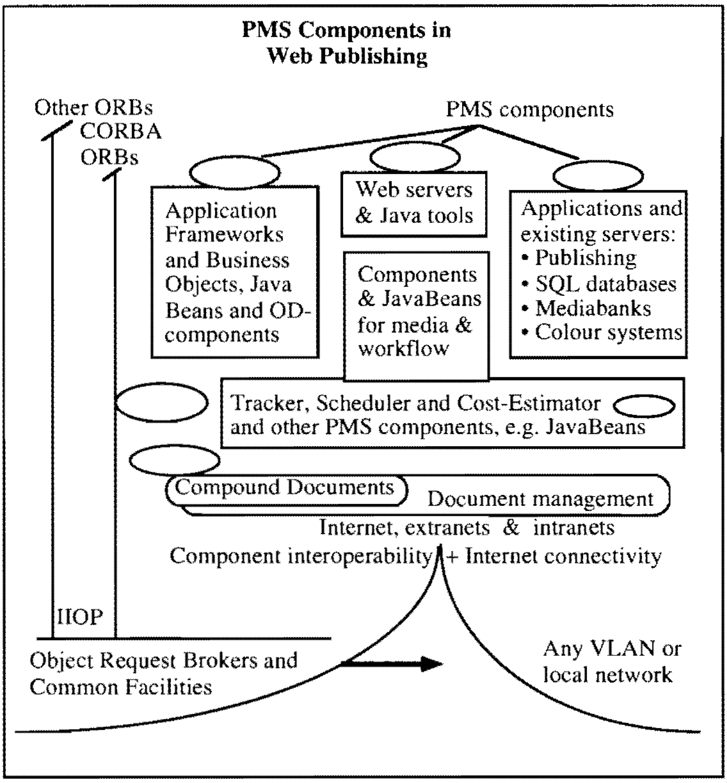

PMS Components in Web Publishing
Other ORBs
CORBA ORBs
PMS components
Application Frameworks and Business Objects, Java Beans and OD-components
Web servers & Java tools
Components & JavaBeans for media & workflow
Applications and existing servers:
• Publishing
• SQL databases
• Mediabanks
• Colour systems
Tracker, Scheduler and Cost-Estimator and other PMS components, e.g. JavaBeans
Compound Documents
Document management
Internet, extranets & intranets
Component interoperability + Internet connectivity
IIOP
Object Request Brokers and Common Facilities
Any VLAN or local network

Fig 3 Tracking and PMS components (⬭) are e.g. JavaBeans.
In Object Web Publishing (web or print) they must interoperate with other frameworks, web servers, documents and applications with a custom-specific style in the web sites, extranets and intranets of users. Interaction means to capture, analyse and report of relevant production data based on CORBA/IIOP interoperation. Media workflow and its PMS may form special component suites for printed and web media.

Components would also be the most persistent way to feed the administration frameworks or later a possible Publisher's Business Object with dynamic **production metadata** (i.e. timing reports, bookings, schedules, consumption of material and resources, waste reports, machine settings, job cost estimates, product structures and models). There is also the question of how much metadata we need and how far the real-time goals are set. As long as these features are provided by traditional software, we will continue to lack PMS.

Finally, we refer to some early **examples** of live OpenDoc and JavaBeans components and their suites. Some of them are related to publishing and design. We have separated these examples in **Appendices 1 and 2**. The examples speak for themselves and show that many early components, as they appear today, have the astonishing features of interoperability and dynamics. We point out that there is a transient period in the CORBA world with changes in the positioning [4] of the object models (mainly between Java and OpenDoc) and the respective component families. The Java-related tools seem to be the most active development path.

We must also mention, that an ActiveX component-based editorial system, developed by a Swedish company, was reported recently in the Seybold Report on Publishing Systems [14]. It is less open than the OD or Java Beans products might be and uses its own protocol. We mention it here to show the trend.

As pointed out in a recent paper [8] there are several componentrs in OpenDoc and Java families which belong to the desktop publishing category, i.e. word processors, document design, database connectors, image and colour design and portable documents. Our two Appendices concentrate on the principles of component interoperation. The near future will offer an increasing variety of component suites and media production business objects.

Conclusions

We recommend that publishers in web and printed media start to follow up and use component-based media data management, CMDM. The WWW and the intranets have triggered this new method and tool set. These lead to improved workflow solutions and more flexible management of data and processes.

There will be two possible segments of markets for the **CMDM**. Let us start from the C-end, of this acronym, i.e. from the **components**, and then conclude with the M-end, i.e. the **management**.

• The components are a remarkable change in the software architecture and imply a more persistent tool set for media production and for related media business processes than the DTP - the existing applications. Business objects and frameworks are a higher level of component families. Consortia for developing CMDM and component-based suites are recommended to avoid overlapping vendor profiles of the future tool sets of media production. Components of the Java family - now Java Beans - will be the leading trend.

• Management of data and processes is a built-in feature of the distributed component concept and CORBA standards. The existing consortia develop a number of business objects [16] in which components have been augmented to process objects and business objects to model and correspond the real-life objects, such as workflows, products, orders, transactions, prices, cost estimates and other company functions.

The latter trend implies that we are able to develop the publishing, its business and market, on the top-down principle and to build more open and generic **Publishing Business Objects** which will be more preassembled large systems. They offer the users - various work groups and teams of publishing - a new level of openness, scaleability and flexibility. The chances of the **users to integrate** their own media workflows and production environments is evident.

For the various and very different basic business ideas of publishing and various media (TV, video, web and print) there might be some shared business objects and also enough room for many vendors, specific suites and frameworks.

Acknowledgement

The authors thank their colleagues at VTT Information Technology, for many fruitful discussions on our joint research topics: Information formats, media, messaging, middleware, WWW, intranets, web media and networking.

References

1. Simo Karttunen "Production Data in Media Systems and Press Frontends: Capture, Formats and Database Methods " Paper for the the Conference on Electronic Imaging, Colour Hardcopy and Graphic Arts, 7-11 October 1996, Berlin, Germany; organized by The European Optical Society (EOS), The Society for Imaging Science and Technology (IS&T), and The International Society for Optical Engineering (SPIE).

2. Simo Karttunen "Production Data Capture and Messaging Architecture in Media Production Networks" Paper for the TAGA '96, Dallas, Texas, May 1996.

3. Robert Orfali, Dan Harkey and Jeri Edwards "The Essential Distributed Objects Survival Guide" John Wiley & Sons. Inc., New York, 1996, 605 pages.

4. Robert Orfali and Dan Harkey "Client/Server Programming with Java and CORBA" John Wiley & Sons Inc., New York, 1997, 600 pages. Other information sources on the OpenDoc, CORBA, Java and distributed component technologies can be found on many web sites, e.g. **http://www.omg.org** and those of the leading vendors Apple, IBM, Informix, Netscape, Oracle and Sun.

5. Simo Karttunen "On the Quality of Data Management of Publishers and Printers" TAGA/ IARIGAI '95 Conference, Paris, France, 17-20 September 1995, TAGA Proceedings Vol. 2, pp. 822-35.

6. Simo Karttunen "On Systematic Production Data Messaging between Publishers and Commercial Printers: The Press Frontend" TAGA '95 Conference, Orlando, Florida, 2-5 April 1995, TAGA Proceedings Vol. 1, pp 419-432.

7. CILabs was a member-based multi-vendor consortium validating OpenDoc components, the LiveObjects, representing a central point in the movement from objects to validated and universally interoperable componenets. CILabs and OMG web sites are recommended. Visit <http://www.cilabs.org>. After IBM and Sun announced to support JavaBeans instead of the OpenDoc as the enterprise object model (Infoworld 17 March 1997, p.1). Apple reduced (but did not stop) the funding for developing OpenDoc. CILabs was later closed.

8. Simo Karttunen, Caj Södergård, Heikki Nikulin and Tomonori Yuasa "Process Timing and Colour Management Data Exchange between Intranet Servers in Media Production" Journal of Prepress and Printing Technology, Vol. 1 (1997) No. 1. Manuscript in prepress preparation since March 1997. First refereed manuscript version delivered in June 1997.

9. Johan Stenberg "Global Production Management in Newspaper Production and Distribution - Coordination of Products, Processes and Resources" Thesis for the Degree of Doctor of Technology at the Royal Institute of Technology, Stockholm, (Dep. KTH GT), to be presented in March 1997.

10. Mark Leon "Database Administration Headache Gets Worse" InfoWorld, 2nd December 1996, p.14. **Craig LaGrow** "Java Fires Up Its Database Engines" Another **web database** analysis report on how to bind database functionality to your web sites, visit <http://www.morph.com/cszone1.3html>. Both authors analyse how the distributed components, the frameworks, the CORBA object services and the TME, by IBM/Tivoli (information and systems management), relate to the web databases.

11. Deborah deVoe "When push comes to shove" Special news report, Infoworld, 17 February, 1997, pp. 1 and 15. This brief report lists 6 "push product vendors" of whom the Marimba/Castanet has the most advanced product family. Visit http://www.marimba.com. Introducing some familiar broadcasting metaphora into the Internet and intranet services means quite new hybrid web media. Castanet of Marimba has been announced to become a part of the MacOs 8 and the Netscape Constellation Browser.

12. Metadata Coalition (MC) unites a group of vendors to define metadata and develop neutral standards and formats for users of metadata. Visit MC and its web site http://www.metadata.org

13. Simo Karttunen, Key Muurikka, Heikki Nikulin and Ari Siren "Intranet Distribution and Colour Control of the Currency Handling Guides for Bank Offices" Paper for the Helsinki '97 ICPO-Interpol 9th International Conference, 9-13 June 1997.

14. Urban Jönér "First Newspaper System Built Around the Internet: The Internet: Backbone of a New Generation of Newspaper Systems" Seybold Report on Publishing Systems Vol. 26 Nimber 9, 27 Jan 1997, p. 1 and 3-7.

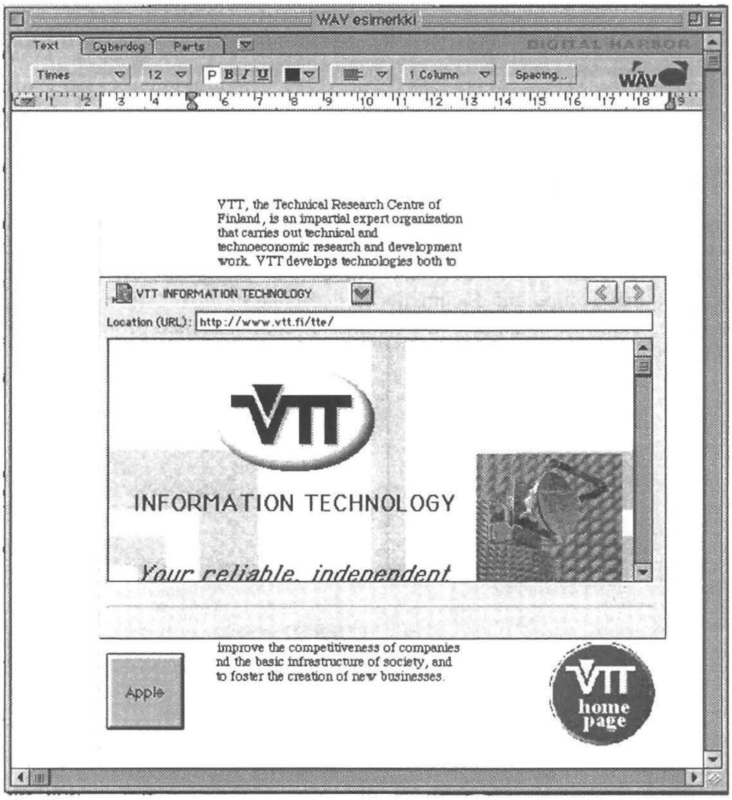
15. PMS formats are new metadata formats which capture and distribute production management data. Both are designed to help the **printed media** and and to solve specific timing and control problems between prepress and printing plants. There are at least two well-known formats: 1.The IFRAtrack (IMF) and 2. the CIP3. Both are object-based and contain product model features. More details of these formats can be found in:
• **Nils Enlund and Philippe Maeght** (IFRA) "A Recommendation for the Interconnection of Production Tracking Systems in Newspaper Production" TAGA/IARIGAI '95 Conference, Paris, France, 17-20 September 1995. Also the original IFRAtrack Recommendation Document by **Bruno Thoyer** : Special Report 6.19. IFRA, Darmstadt, July 1995. <http://www.gt.kth.se>.
• **Stefan Daun, Georg Lucas and Jürgen Schönhut** "Specification of the **CIP3** Print Production Format" Version 1.0. Fraunhofer IGD, Document Imaging Dept. Darmstadt, Germany. IGD's WWW site <http://www.igd.fhg.de>

16. Distributed components and business objects are modern software architectures for web commerce and publishing: E.g. WWW: Lycos Search with key words: Businessobjects OMG distributed objects components.

17. Simo Karttunen and Heikki Nikulin. The authors of this paper have plans to develop the CMDM and the related database and OLAP concepts in much more detail in media production environments to integrate new prototypes, components and compound documents to demonstrate how future publishing might work. This will take place in a number of commissioned projects at VTT Information Technology in 1997-98.

Interoperating OpenDoc components

Sample screenshots of using the OpenDoc aware WAV editor from Digital Harbour together with CyberButton component and Cyberdog browser (Apple´s OD components).

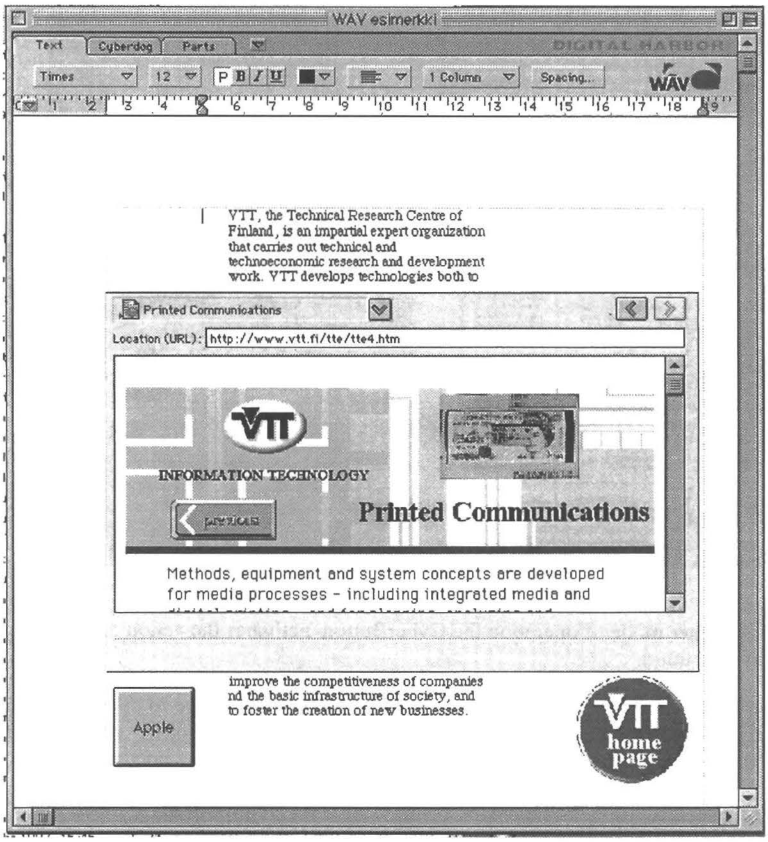


VTT, the Technical Research Centre of Finland, is an impartial expert organization that carries out technical and technoeconomic research and development work. VTT develops technologies both to improve the competitiveness of companies nd the basic infrastructure of society, and to foster the creation of new businesses.

• The WAV Editor is opened with some text in it.

• In the next stage the CyberDog web browser and Apple-labeled CyberButton have been inserted into the document. The inserted web browser and the button are essential parts of the document itself.

• The buttons which are easy to configure could open other files, web pages or databases for supporting the session with the WAV editor.



VTT, the Technical Research Centre of Finland, is an impartial expert organization that carries out technical and technoeconomic research and development work. VTT develops technologies both to

Location (URL): http://www.vtt.fi/tte/tte4.htm

Methods, equipment and system concepts are developed for media processes - including integrated media and

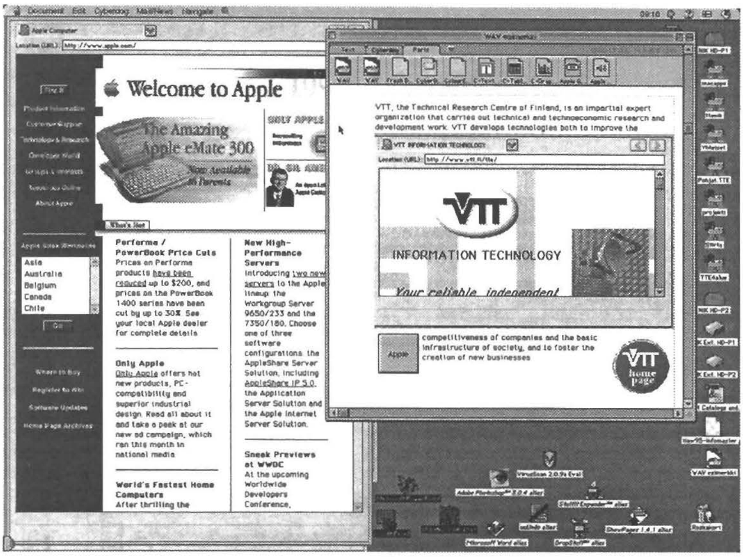
improve the competitiveness of companies nd the basic infrastructure of society, and to foster the creation of new businesses.

• This screenshot shows the use of CyberDog browser inside the WAV editor.

• The previous browser window (VTT Information Technology, screenshot on page 17) has been changed to another web site with www-address: VTT Information Technology: Printed Communications.

• For instance the web site content of the "VTT Information Technology. Printed Communications" could be updated by the WAV editor.

• Similarily the WAV editor session uses always the updated and latest version of the above web site if that has been inserted as the inside content of the document by the button or by the component browser.



Performa / PowerBook Price Cuts
Prices on Performa products have been reduced up to $200, and prices on the PowerBook 1400 series have been cut by up to 30%. See your local Apple dealer for complete details.

Only Apple
Only Apple offers hot new products, PC-compatibility and superior industrial design. Read all about it and take a peek at our new ad campaign, which ran this month in national media.

World's Fastest Home Computers
After thrilling the

New High-Performance Servers
Introducing two new servers to the Apple lineup: the Workgroup Server 9650/233 and the 7350/180. Choose one of three software configurations: the AppleShare Server Solution, including AppleShare IP 5.0, the Application Server Solution and the Apple Internet Server Solution.

Sneak Previews at WWDC
At the upcoming Worldwide Developers Conference,

VTT, the Technical Research Centre of Finland, is an impartial expert organization that carries out technical and technoeconomic research and development work. VTT develops technologies both to improve the competitiveness of companies and the basic infrastructure of society, and to foster the creation of new businesses.

Location (URL): http://www.vtt.fi/tte/

• In another mode of operation the CyberButton activates the action embedded into the button.

• In this case the button commands to open Apples home page in the CyberDog browser **outside** the document.. In this case the browser is not an integral part of the document (but the the button reference is included in the document).

• In the end of 1996 the OpenDoc component software producers founded the Component 100 (C100). This association was an initiative of the Digital Harbor, the designers of WAV™, to promote component markets.

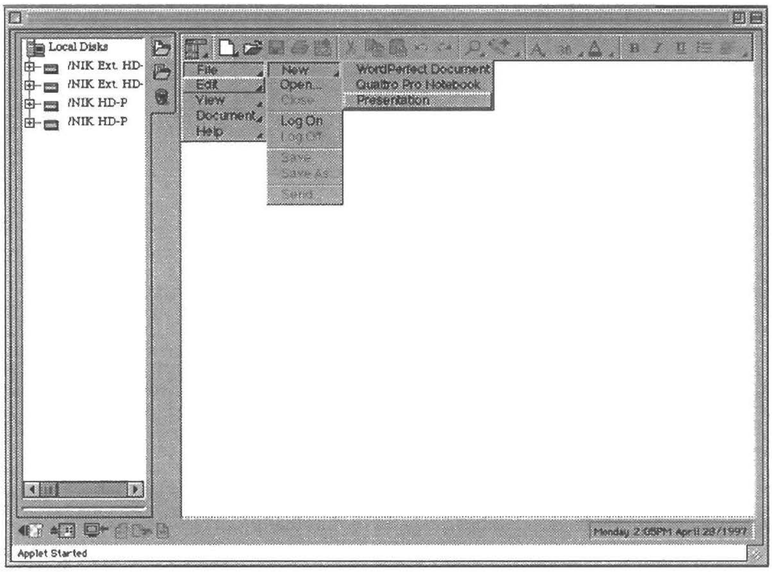
• This Association comprised of twenty-five companies from around the world developing component products including Digital Harbor, Nisus Software Inc., MetaMind Software, Totally Hip, WorldSoft, Adrenaline Software, Corda Inc., SoftLinc, Bare Bones Software, ComGrafix, Inc., Eclipse Services, Hutchings Software, Pictorius, Inc., PreFab Software, Inc., SimCalc Inc., theta group (Germany), Claris, to name a few, started to wortk toward OpenDoc component parts and editors. Some of these were mentioned in our recent report [8].

• Since then Apple has changed its plans to move its component technology resources to Java and the OPENSTEP APIs in the coming Rhapsody operating system. The new direction brings the support from wider general Java and object management developments to the component development work.

• Digital Harbor is already making the transition to Java. In their opinion the Java technologies must be seen as the emerging standard for object oriented programming and will imply the future of component interoperability.

Interoperating Java components

The component technology is rapidly moving into the Java sphere. New and powerful applications are emerging. Some early samples of the first publishing related developments of component software are shown below.



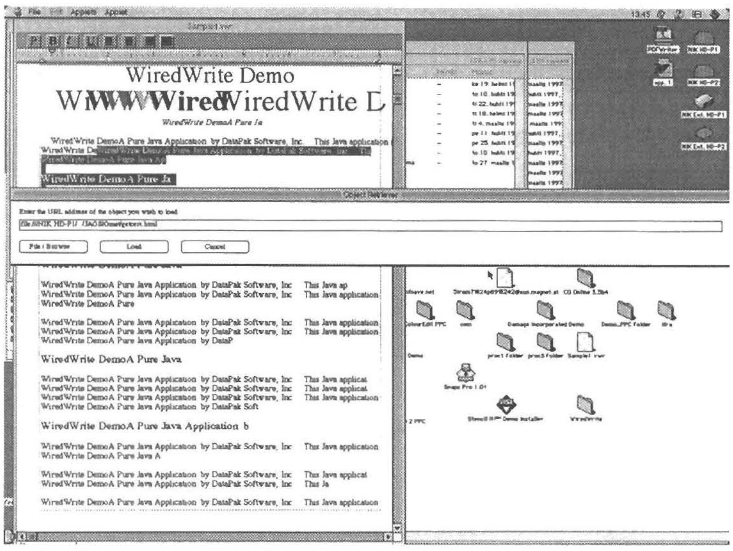
• A screenshot of **Corel Office for Java**, a Java based office suite, transformed from the well known Corel Office™.

• Application builds its own desktop environment and is thus separated from the OS it runs on. Desktop runs over the Java runtime engine powered by the operating system of the workstation.

• In the upper part of the desktop we have the tool bars and document handling functions which seem very normal and complete.

• The resources are shown on the left hand column as the connected hard disks.

• The office functions, namely calendar, spreadsheet, presentation and word processor (Word Perfect of Corel), work seamlessly. The speed of execution is very fast because the Java object classes are small, i.e. about a few kilobytes.



WiredWrite Demo

WiWWWWiredWiredWrite D

WiredWrite DemoA Pure Ja

WiredWrite DemoA Pure Java Application by DataPak Software, Inc This Java application

WiredWrite DemoA Pure Ja

Enter the URL address of the object you wish to load

WiredWrite DemoA Pure Java Application by DataPak Software, Inc This Java ap
WiredWrite DemoA Pure Java Application by DataPak Software, Inc This Java application
WiredWrite DemoA Pure

WiredWrite DemoA Pure Java Application by DataPak Software, Inc This Java application
WiredWrite DemoA Pure Java Application by DataPak Software, Inc This Java application
WiredWrite DemoA Pure Java Application by DataP

WiredWrite DemoA Pure Java

WiredWrite DemoA Pure Java Application by DataPak Software, Inc This Java applicat
WiredWrite DemoA Pure Java Application by DataPak Software, Inc This Java applicat
WiredWrite DemoA Pure Java Application by DataPak Software, Inc This Java application
WiredWrite DemoA Pure Java Application by DataPak Soft

WiredWrite DemoA Pure Java Application b

WiredWrite DemoA Pure Java Application by DataPak Software, Inc This Java application
WiredWrite DemoA Pure Java A

WiredWrite DemoA Pure Java Application by DataPak Software, Inc This Java applicat
WiredWrite DemoA Pure Java Application by DataPak Software, Inc This Ja

WiredWrite DemoA Pure Java Application by DataPak Software, Inc This Java application

• This screenshot from WiredWrite™, a Java document builder from J.Stream, Canada, displays the inherent communication capabilities of the Java-based applications. In the screenshot we can see the link bar in the middle of the document, as the top layer of the screenshot.

• The link bar addresses and connects the session and document to any files, web sites, documents and databases - and dynamically during the sessions.

• This desktop (on the left column of the screenshot) looks very simple but various links and functions operate easily and produce complex documents very efficiently.

• WiredWrite™ uses its own portable document format JPD with very complete feature sets for both building and handling structured web documents. It is best described by saying (our expression) that JPD does the same as Word, Frame, PageMill and Acrobat, only **much faster** - because it is pure Java.

• The desktop, shown on the righ hand side, belongs to the workstation in question, a PowerMac/MacOS.

• We recommend to follow up this document builder component and its later similar competitors.