WALK-UP PRINTING

FIELD OF THE INVENTION

The present invention relates to printing systems, and in particular printing systems involving interactive paper, computer publishing, computer applications, human-computer interfaces, and information appliances.

CROSS-REFERENCES

10/815621	10/815612	10/815630	10/815637	10/815638	10/815640	10/815642
10/815643	10/815644	10/815618	10/815639	10/815635	10/815647	10/815634
10/815632	10/815631	10/815648	10/815641	10/815645	10/815646	10/815617
10/815620	10/815615	10/815613	10/815633	10/815619	10/815616	10/815614
10/815636	10/815649	11/041650	11/041651	11/041652	11/041649	11/041610
11/041609	11/041626	11/041627	11/041624	11/041625	11/041556	11/041580
11/041723	11/041698	11/041648	10/815609	10/815627	10/815626	10/815610
10/815611	10/815623	10/815622	10/815629	10/815625	10/815624	10/815628
10/913375	10/913373	10/913374	10/913372	10/913377	10/913378	10/913380
10/913379	10/913376	10/913381	10/986402	IRB013US	11/172815	11/172814
10/409876	10/409848	10/409845	11/084769	11/084742	11/084806	09/575197
09/575195	09/575159	09/575132	09/575123	6825945	09/575130	09/575165
6813039	09/693415	09/575118	6824044	09/608970	09/575131	09/575116
6816274	09/575139	09/575186	6681045	6678499	6679420	09/663599
09/607852	6728000	09/693219	09/575145	09/607656	6813558	6766942
09/693515	09/663701	09/575192	6720985	09/609303	6922779	09/609596
6847883	09/693647	09/721895	09/721894	09/607843	09/693690	09/607605
09/608178	09/609553	09/609233	09/609149	09/608022	09/575181	09/722174
09/721896	10/291522	6718061	10/291523	10/291471	10/291470	6825956
10/291481	10/291509	10/291825	10/291519	10/291575	10/291557	6862105
10/291558	10/291587	10/291818	10/291576	6829387	6714678	6644545
6609653	6651879	10/291555	10/291510	10/291592	10/291542	10/291820
10/291516	6867880	10/291487	10/291520	10/291521	10/291556	10/291821
10/291525	10/291586	10/291822	10/291524	10/291553	6850931	6865570
6847961	10/685523	10/685583	10/685455	10/685584	10/757600	10/804034
10/793933	6889896	10/831232	10/884882	10/943875	10/943938	10/943874
10/943872	10/944044	10/943942	10/944043	10/949293	10/943877	10/965913
10/954170	10/981773	10/981626	10/981616	10/981627	10/974730	10/986337
10/992713	11/006536	11/020256	11/020106	11/020260	11/020321	11/020319
11/026045	11/059696	11/051032	11/059674	NPA19NUS	11/107944	11/107941
11/082940	11/082815	11/082827	11/082829	11/082956	11/083012	11/124256

11/123136	11/154676	11/159196	NPA225US	09/575193	09/575156	09/609232
09/607844	6457883	09/693593	10/743671	11/033379	09/928055	09/927684
09/928108	09/927685	09/927809	09/575183	6789194	09/575150	6789191
10/900129	10/900127	10/913328	10/913350	10/982975	10/983029	6644642
6502614	6622999	6669385	6827116	10/933285	10/949307	6549935
NPN004US	09/575187	6727996	6591884	6439706	6760119	09/575198
09/722148	09/722146	6826547	6290349	6428155	6785016	6831682
6741871	09/722171	09/721858	09/722142	6840606	10/202021	10/291724
10/291512	10/291554	10/659027	10/659026	10/831242	10/884885	10/884883
10/901154	10/932044	10/962412	10/962510	10/962552	10/965733	10/965933
10/974742	10/982974	10/983018	10/986375	11/107817	11/148238	11/149160
09/693301	6870966	6822639	6474888	6627870	6724374	6788982
09/722141	6788293	09/722147	6737591	09/722172	09/693514	6792165
09/722088	6795593	10/291823	6768821	10/291366	10/291503	6797895
10/274817	10/782894	10/782895	10/778056	10/778058	10/778060	10/778059
10/778063	10/778062	10/778061	10/778057	10/846895	10/917468	10/917467
10/917466	10/917465	10/917356	10/948169	10/948253	10/948157	10/917436
10/943856	10/919379	10/943843	10/943878	10/943849	10/965751	11/071267
11/144840	11/155556	11/155557	09/575154	09/575129	6830196	6832717
09/721862	10/473747	10/120441	6843420	10/291718	6,789,731	10/291543
6766944	6766945	10/291715	10/291559	10/291660	10/409864	NPT019US NP
10/537159	NPT022US	10/410484	10/884884	10/853379	10/786631	10/853782
10/893372	10/893381	10/893382	10/893383	10/893384	10/971051	10/971145
10/971146	10/986403	10/986404	10/990459	11/059684	11/074802	10/492169
10/492152	10/492168	10/492161	10/492154	10/502575	10/683151	10/531229
10/683040	NPW009USNP	10/510391	10/919260	10/510392	10/919261	10/778090
09/575189	09/575162	09/575172	09/575170	09/575171	09/575161	10/291716
10/291547	10/291538	6786397	10/291827	10/291548	10/291714	10/291544
10/291541	6839053	10/291579	10/291824	10/291713	6914593	10/291546
10/917355	10/913340	10/940668	11/020160	11/039897	11/074800	NPX044US
11/075917	11/102698	11/102843	6593166	10/428823	10/849931	11/144807
6454482	6808330	6527365	6474773	6550997	10/181496	10/274119
10/309185	10/309066	10/949288	10/962400	10/969121	UP21US	UP23US
09/517539	6566858	09/112762	6331946	6246970	6442525	09/517384
09/505951	6374354	09/517608	6816968	6757832	6334190	6745331
09/517541	10/203559	10/203560	10/203564	10/636263	10/636283	10/866608
10/902889	10/902833	10/940653	10/942858	10/727181	10/727162	10/727163
10/727245	10/727204	10/727233	10/727280	10/727157	10/727178	10/727210
10/727257	10/727238	10/727251	10/727159	10/727180	10/727179	10/727192

10/727274	10/727164	10/727161	10/727198	10/727158	10/754536	10/754938
10/727227	10/727160	10/934720	10/296522	6795215	10/296535	09/575109
6805419	6859289	09/607985	6398332	6394573	6622923	6747760
6921144	10/884881	10/943941	10/949294	11/039866	11/123011	11/123010
11/144769	11/148237	10/922846	10/922845	10/854521	10/854522	10/854488
10/854487	10/854503	10/854504	10/854509	10/854510	10/854496	10/854497
10/854495	10/854498	10/854511	10/854512	10/854525	10/854526	10/854516
10/854508	10/854507	10/854515	10/854506	10/854505	10/854493	10/854494
10/854489	10/854490	10/854492	10/854491	10/854528	10/854523	10/854527
10/854524	10/854520	10/854514	10/854519	10/854513	10/854499	10/854501
10/854500	10/854502	10/854518	10/854517	10/934628	11/003786	11/003354
11/003616	11/003418	11/003334	11/003600	11/003404	11/003419	11/003700
11/003601	11/003618	11/003615	11/003337	11/003698	11/003420	11/003682
11/003699	11/071473	11/003463	11/003701	11/003683	11/003614	11/003702
11/003684	11/003619	11/003617	10/760254	10/760210	10/760202	10/760197
10/760198	10/760249	10/760263	10/760196	10/760247	10/760223	10/760264
10/760244	10/760245	10/760222	10/760248	10/760236	10/760192	10/760203
10/760204	10/760205	10/760206	10/760267	10/760270	10/760259	10/760271
10/760275	10/760274	10/760268	10/760184	10/760195	10/760186	10/760261
10/760258	11/014764	11/014763	11/014748	11/014747	11/014761	11/014760
11/014757	11/014714	11/014713	11/014762	11/014724	11/014723	11/014756
11/014736	11/014759	11/014758	11/014725	11/014739	11/014738	11/014737
11/014726	11/014745	11/014712	11/014715	11/014751	11/014735	11/014734
11/014719	11/014750	11/014749	11/014746	11/014769	11/014729	11/014743
11/014733	11/014754	11/014755	11/014765	11/014766	11/014740	11/014720
11/014753	11/014752	11/014744	11/014741	11/014768	11/014767	11/014718
11/014717	11/014716	11/014732	11/014742	11/097268	11/097185	11/097184
10/728804	10/728952	10/728806	10/728834	10/729790	10/728884	10/728970
10/728784	10/728783	10/728925	10/728842	10/728803	10/728780	10/728779
10/773189	10/773204	10/773198	10/773199	6830318	10/773201	10/773191
10/773183	10/773195	10/773196	10/773186	10/773200	10/773185	10/773192
10/773197	10/773203	10/773187	10/773202	10/773188	10/773194	10/773193
10/773184	11/008118	11/060751	11/060805	MTB40US	11/097308	11/097309
11/097335	11/097299	11/097310	11/097213	11/097212	10/760272	10/760273
10/760187	10/760182	10/760188	10/760218	10/760217	10/760216	10/760233
10/760246	10/760212	10/760243	10/760201	10/760185	10/760253	10/760255
10/760209	10/760208	10/760194	10/760238	10/760234	10/760235	10/760183
10/760189	10/760262	10/760232	10/760231	10/760200	10/760190	10/760191
10/760227	10/760207	10/760181	10/407212	10/407207	10/683064	10/683041
6750901	6476863	6788336	6623101	6406129	6505916	6457809

6550895 6457812 10/296434 6428133 6746105

The disclosures of these co-pending applications are incorporated herein by cross-reference. Some applications are temporarily identified by their docket number. This will be replaced by the corresponding USSN when available.

BACKGROUND OF THE INVENTION

In office environments, documents to be printed are typically sent via local computer networks to one of a number of printers connected to the network. The nominated printer is usually the most convenient to the user but unless the user goes to collect the document immediately after sending the print job, the printed document waits in the collection tray. If the document is sensitive, there is a risk that its contents will be disclosed to others passing the printer.

SUMMARY OF THE INVENTION

According to a first aspect, the present invention provides a computer network for a plurality of users, the computer network comprising:

a server;

a printer;

a network user identifier for a network user to carry on their person; and,

a printer identifier associated with the printer; wherein during use,

the network user identifier and the printer identifier interact such that any of the network user's pending printouts are sent to the printer for printing when the network user is proximate the printer.

By keeping print jobs on the network until the user is at a printer, the printed document does not sit in a collection tray until the user retrieves it. This reduces the risk of others seeing any sensitive documents. If all printers connected to the network have sensors for identifying individual users, then users will not need to select a printer (or designate a default printer). Print jobs can be collected from the most convenient printer regardless of a users current location in the office.

The Netpage system is comprehensively described in the cross referenced documents as well as the Detailed Description below. This system uses a paper- and pen-based interface to computer-based and typically network-based information and applications. The user can request print jobs by 'clicking' an interactive element on a Netpage document with a Netpage pen and therefore may be remote from any of the networked printers or even the office when print jobs are requested. According the invention is particularly suited to the Netpage system and will be described with particular reference to its operation within this environment. However, it will be appreciated that the invention has much broader application than Netpage and is not limited or restricted to printing Netpage documents.

Optionally, the network comprises a plurality of said printers, each printer associated with one of the printer identifiers respectively; and

a plurality of said network user identifiers, each uniquely identifying different network users.

Optionally, each of the network user identifiers is a token and each of the printer identifiers has a token reader such that the user presents their token to the token reader associated with one of the printers to request actual printing of their queued printouts via that printer.

Optionally, the tokens are a short-range RFID tag, a smartcard or a magnetic stripe card.

Optionally, the token reader notifies a walk-up-handling application on the server of the user's proximity to the associated printer which in turn initiates printing.

Optionally, each of the printer identifiers is a token and each of the network user identifiers has a token reader associated with the user. Optionally, the token reader is an electronic stylus with an optical sensor, and the tokens are a surface each of the printers with coded data disposed on it, the coded data being readable by the optical sensors of each users' electronic stylus.

Optionally, the pending printouts are maintained in a queue by the server and each pending printout has a priority such that higher-priority printouts are printed before earlier-queued but lower-priority printouts.

Optionally, the token readers are associated with respective printers such that when the user presents their token to the reader it reads the token and identifies both the user and the printer to the server. Optionally, the token identfies the user explicitly. Optionally, the token has a token identifier and the server performs a database lookup to translate the token identifier into a user identity. Optionally, the token reader identifies the printer explicitly. Optionally, the reader has a reader identifier and the server performs a database lookup to translate the reader identifier into a printer indentity.

Optionally, the token reader and the printer are separate devices which have an electrical connection. Optionally, the token reader is physically built into the printer. Optionally, the reader informs the printer that the user has presented a token and the printer then explicitly retrieves the user's pending printouts for printing.

Optionally, the token is a security access or identification badge or card.

BRIEF DESCRPTION OF THE DRAWINGS

Embodiments of the invention will now be described by way of example only with reference to the accompanying drawings in which:

Figure 1 shows the data flow between Netpage publishers and applications, Netpage services, and Netpage devices;

Figure 2 is a diagram of the range of content type within a Netpage document;

Figure 3 shows a Netpage document with a physical structure consisting of a sequence of numbered pages;

Figure 4 shows a printout consisting of a series of impressions;

Figure 5 is a diagram showing a user with a pen and default printer;

Figure 6 shows the pen events recorded in a digital ink stream;

Figure 7 shows the form data submitted to an application;

Figure 8 shows a dynamic element for use as a document element;

Figure 9 shows a dynamic object linked to an existing impression;

Figure 10 shows the relationship between the document, printout and digital ink stores;

Figure 11 shows the fundamental flow of data in the Netpage system in greater detail than Figure 1;

Figure 12 shows the data flow associated with reprinting impressions;

Figure 13 shows the data flow associated with printing;

Figure 14 shows a bifurcated general printing data flow;

Figure 15 shows the data flow associated with walk-up printing;

Figure 16 shows the data flow associated with the establishment of a printout queue;

Figure 17 shows the different levels of network distribution and access possible within the Netpage system;

Figure 18 shows the data flow if the user has a token read by a reader associated with the printer;

Figure 19 shows the data flow if the user has a reader for reading the token associated with the printer;

Figure 20 shows the data flow if the user has a reader that reads the printer token but then uses the printer reader to connect to the Netpage server;

Figure 21 shows the data flow between a privately hosted network and a publicly hosted network;

Figure 22 shows a PC or device hosted Netpage system;

Figure 23 shows the structure of a complete tag;

Figure 24 shows a symbol unit cell;

Figure 25 shows nine symbol unit cells;

Figure 26 shows the bit ordering in a symbol;

Figure 27 shows a tag with all bits set;

Figure 28 shows a tag group made up of four tag types;

Figure 29 shows the continuous tiling of tag groups;

Figure 30 shows the interleaving of codewords A, B, C & D with a tag;

Figure 31 shows a codeword layout; and

Figure 32 shows a tag and its eight immediate neighbours labelled with its corresponding bit index.

DETAILED DESCRIPTION

As discussed above, the invention is well suited for incorporation in the Assignee's Netpage system. In light of this, the invention has been described as a component of a broader Netpage architecture. However, it will be readily appreciated that the invention is also applicable to other computer networks.

NETPAGE ARCHITECTURE

OVERVIEW

Figure 1 shows the interaction between Netpage publishers, applications, services and devices. The Netpage document service 1 accepts a document 2 from a Netpage publisher 3 or other Netpage application 4, and produces a printout 5 via a Netpage printer 6. A printout 5 consists of a series of impressions on either or both sides of a series of paper sheets. In addition to reproducing the graphic content of the document 2 on paper, the printer 6 also lays down a coordinate grid in the form of an array of invisible millimetre-scale tags 7 (see USSN 10/309,358 cross referenced above). Each tag encodes the two-dimensional coordinates of its location on the impression as well as the impression's unique identifier. When a tag is optically imaged by a Netpage pen 8 (see below and USSN 10/815,636 cross referenced above) the pen is able to identify the corresponding impression as well as its own position relative to the impression. When the user of the pen 8 moves the pen relative to the coordinate grid 7, the pen generates a stream of positions. This stream is referred to as digital ink 9. A digital ink stream also records when the pen makes contact with a surface and when it loses contact with a surface, and each pair of these so-called pen down and pen up events delineates a stroke drawn by the user using the pen.

The Netpage tag pattern 7 is typically printed using an invisible infrared ink, while visible graphic content is printed using colored inks which are transparent in the infrared part of the spectrum. The Netpage pen 8

incorporates a conventional marking nib which utilises an infrared-transparent ink so as not to obscure the tag pattern 7.

Because the impression identifiers (tags) manifest themselves in printed impressions, they are engineered to be unique among all Netpage systems, and therefore rely on a global allocation mechanism.

The document 2 may include an input description 11 which defines command and form data 12. The commands are instructions that may be activated by the user and the forms have designated fields that may be filled in by the user. Both commands and form fields have active zones, i.e. areas of the page where they capture user input.

The Netpage digital ink service 13 accepts digital ink 9 from a Netpage pen 8. Since the pen typically only has a short-range communications capability, it forwards the digital ink 9 to the Netpage digital ink service 13 via a Netpage relay 14 which has a longer-range communications capability. Typical relays include mobile phones, PDAs and personal computers.

The digital ink service 13 uses the impression identifier 7 in the digital ink 9 to retrieve the corresponding impression and input description 11 from the document service 1, and attempts to assign each individual digital ink stroke to a form of the input description 11. Once it detects that the user of the pen 8 has designated a form submission command, it interprets the digital ink 9 assigned to the form and submits the resultant form data 12 to the application associated with the command.

In order to allow the digital ink service to interpret pen input in relation to a particular impression, the document service 1 keeps a copy of every input description 11 it prints.

In order to allow a user to fill in a form over an arbitrarily long time, the digital ink service 13 retains a copy of all digital ink 9 it receives, at least until the digital ink is interpreted and submitted to an application 4. The digital ink service 13 optionally retains all digital ink 9 indefinitely, to allow digital ink searching of both form content and document annotations.

The Netpage pen 8, or a simpler Netpage pointer, may be incorporated directly into a hand-held device such as a mobile phone or PDA. Conversely, the pen may incorporate a long-range communications capability and not need a separate relay.

Since the relay device 14 typically incorporates an interactive display 15, the digital ink service 13 may identify the interactive display 15 to a target application 4 to allow the application to communicate directly with the interactive display, thus allowing an interaction initiated via paper and pen to lead to a richer screen-based interaction, and generally allowing the development of hybrid paper- and screen-based applications which make the most of both media.

In the presence of multiple distributed digital ink services 13 a pen 8 (or its relay 14) may use a name service to resolve the network address of a target digital ink service, based on pen identifier and possibly impression identifier. In the presence of multiple distributed document services 1, a digital ink service 13 uses a name service to resolve the network address of a document service, based on impression identifier.

Although the above description centres on a forms-based interpretation of digital ink and subsequent delivery of form data to an application, the digital ink service also supports streaming delivery of digital ink to an application. This allows an application to be more directly responsive to pen input. In streaming mode the digital ink service delivers both stroke digital ink and intervening "hover" digital ink to allow the application to provide real-time positional feedback to the user via a display.

OBJECT MODEL

The object model is a logical model relating to the external interfaces of the Netpage services. It is not intended as an implementation model.

Document

Figure 2 is a class diagram showing a document 2 comprising a visual description 16 and an input description 11. For a given document, either description may be empty. Each document 2 is uniquely identified 18.

The visual description 16 is a collection of visual elements 20 representing static 22 and dynamic elements 24. Static elements represent textflows 26, images 28, graphics 30 etc. Dynamic elements 24 are described below.

The input description 11 is a collection of forms 32, each of which consists of a collection of input elements 34 representing commands 36 and fields 38. Forms 32 may overlap both physically and logically, and the same input element 34 may participate in multiple forms. Each input element 34 has a zone 40 which defines the area within which it captures input. Each form 32 is associated with a target application 42. The application 42 receives submissions of the form 32. The application 42 is identified by an address 44.

As illustrated in the class diagram in Figure 3, a document 2 has a physical structure which consists of a sequence of numbered pages, each page 46 assigned a page number 54. The document elements 48 are each assigned a specific position 52 in the sequence of pages. Since a single document element 48 may span a number of pages 46, it may have a corresponding number of page elements 50, each defining the position 52 of a fragment of the document element 48.

Printout.

Referring to the class diagram in Figure 4, a printout 5 consists of a series of impressions 58 assigned a printout ID 56. With "N-up" printing, multiple pages 46 may appear on a single impression 58, while with "poster" printing a single page 46 may span multiple impressions 58. A page impression 64 uses a transform 66 to represent the position, scale and rotation of a page 46 on an impression 58.

Each impression 58 is identified by a unique identifier 60 which is encoded with the impression's coordinate grid when the impression is printed.

Once actually printed (or pending printing in the 'walk-up scenario' described below), the impression 58 is associated with both the printer 6 on which it was printed and the user 62 who requested it, if known.

As shown in Figure 5, a pen 8 is owned by a single user 62 but a user may own any number of pens 8. Accordingly, the user 62 is assigned a user ID and other user details 68, and likewise, each pen 8 and printer 6 has a pen ID and details 70, and printer ID and details 72. A user 62 optionally has a default printer 6.

The class diagram in Figure 6 illustrates a single digital ink stream 74 associated with a pen 8, consisting of a sequence of pen events 76. Each pen event is timestamped 78 using a nib force sensor. Successive segments 80 within the stream 74 relate to different impressions 58. Each segment is assigned a number 82. Each sequence of pen events 76 is a series of pen position events 88 between a pen down event 84 and a pen up event 86. This defines a stroke drawn by the user of the pen. Often a succession of strokes relates to the same impression 58, and usually a segment boundary corresponds to a stroke boundary. However, a stroke may also traverse multiple impressions 58, and a stream 74 may contain "hover" pen events between strokes.

The class diagram in Figure 7 shows form data 12 submitted to an application consists of a collection of field values 90. The form data 12 is associated with a unique form instance 92 appearing in a printout 5. An application may specify a transaction identifier when the form instance 92 is first created (as part of a printout). The transaction identifier 94 is submitted together with the form data 12, allowing the target application to use it to index a unique transaction context.

The digital ink service 13 (see Figure 1) supports a form lifecycle wherein a form may only be submitted once, may expire, may become frozen after being signed, and may be voided. The form instance reflects the status of the form with respect to the form lifecycle.

As illustrated in the class diagram in Figure 8, a document 2 may also include dynamic elements 24. Each dynamic element has an associated dynamic object 96, which in turn has associated object data 98 and a (typically type-specific) object application 99. A dynamic element 24 may be activated in place using a device such as a Netpage viewer (see USSN 09/722,175 cross referenced above), or may be activated on an arbitrary interactive display, such as the interactive display 15 associated with the relay 14 (see Figure 1), or may be activated via the Netpage Explorer (described below).

Examples of dynamic objects and their related applications include an audio clip and an audio player, a video clip and a video player, a photo and a photo viewer, a URL and a Web browser, an editable document and a word processor, to name just a few.

As illustrated in the class diagram in Figure 9, a dynamic object 96 may also be dynamically linked to an arbitrary location on an existing impression, e.g. by being "pasted" onto a virtual view of the impression or onto the impression itself.

Figure 10 shows the relationships between the three stores nominally maintained by the Netpage document service 1 and the Netpage digital ink service 13 (see Figure1), with navigational qualifiers.

Apart from the document store 100, the printout store 102 and the digital ink store 104, the Netpage services may have additional stores for registered users 62, pens 8 and printers 6, identifier allocation, and service address resolution (not shown).

FUNCTIONS

The processes and stores described in the following sub-sections are meant to elucidate functionality rather than imply implementation.

Form Input

Figure 11 shows the fundamental flow of data in the Netpage System in more detail than Figure 1. The document service 1 allows an application 4 to lodge a document 2 and to separately transmit a print request 106 to print the document 2. It retains a copy of each lodged document in the document store 100, and retains a copy of the document's input description, if any, in the document store 100. When it prints a document 2 to a specified printer 6, it records the printout 5 in the printout store 102.

The digital ink service 13 accepts digital ink 9 from a pen 8 via a relay 14, and retains a copy of received digital ink in the digital ink store 104. It uses the impression identifier 60 in the digital ink 9 to retrieve the corresponding impression 58 and input description from the document service 1. It then assigns each individual digital ink stroke to an element of the input description such as a command or a form field, according to the position and extent of the stroke and the active zone of the input element. Once it detects

that the user of the pen 8 has designated a form submission command, the digital ink 9 assigned to each field is interpreted 108 according to field type, and the resultant form data 12 is submitted to the application 4 associated with the command. For example, the digital ink service 13 interprets a mark in a checkbox as a check mark; it converts handwritten text in a text field into a string of text characters using intelligent character recognition; and it compares a handwritten signature in a signature field with the recorded signature of the user of the pen, and, if the signatures match, digitally signs the form data on behalf of the user.

Reprinting Impressions

The Netpage system supports reprinting of previously printed impressions, with or without any drawings or handwriting captured via those impressions. It thus supports source-independent document reproduction. Figure 12 illustrates the flow of data in response to a reprint request 110 from an application 4. When the document service 1 reprints a set of impressions 58 it optionally includes any drawings and handwriting captured via those impressions, and retrieves the corresponding digital ink from the digital ink store 104 in the digital ink service 13 (subject to visibility and access). It records a new printout to record the impression identifiers assigned to the reprinted impressions 112.

General Printing

Netpage system acts as a virtual filing cabinet for any printed document, whether produced by a Netpage-aware application or not. A Netpage-aware application has the advantage that it can include input descriptions in its documents, while a non-Netpage-aware application benefits from its printed documents supporting searchable annotations and source-independent reprinting.

Figure 13 illustrates the flow of data in response to a general printing request from a non-Netpage-aware application 114. A Netpage-aware printer driver 116 converts platform-specific drawing commands 118 into a Netpage-compatible document 2 which it lodges with the document service 1, and then sends a print request 106 for the document service 1 to print the document 2 via a specified printer 6.

Figure 14 illustrates the corresponding flow of data when the printer is not accessible by the document service 1. Here the printer driver 116 still lodges the document 2 with the document service 1 and records the printout 5 in the printout store 102, but actually prints the documents 2 directly via the specified printer 6.

Walk Up Printing

When a user requests printing of a document via a conventional user interface, it is usually convenient to specify a target printer. In the Netpage system, however, printing often occurs in response to user input via a printed form, and it may be inconvenient to specify a target printer. In some environments, such as in a home containing a single printer, the desired target printer may be inferred. In other environments, such as in an office containing multiple networked printers, the desired target printer may not be so easily inferred. In such environments it is useful to allow a user to specify a target printer by walking up to it.

Figure 15 shows the flow of data in a walk-up environment. All print (and re-print) requests 120 from the Netpage application 4 are typically deferred. In response to a deferred print request 120, the document service 1 records a printout 5 in the printout store 102 to capture impression-related information, and places the printout in a printout pending queue 122 for the requesting user.

In one possible configuration, each printer 6 has an associated token reader 124, and the user presents a token 126 to the token reader to request actual printing of queued printouts via the printer 6. The token 126 may be

a short-range RFID tag, a smartcard, a magnetic stripe card, etc. The token reader 124 notifies a walk-up-handling application 128 of the user's proximity to the printer which in turn initiates printing via the document service 1.

In another possible configuration, the token reader 124 is associated with the user and the token 126 is associated with the printer 6. For example, the token reader 124 may be the user's Netpage pen 8, and the token 124 may be a tag pattern disposed on the printer.

Figure 16 shows the class diagram of the pending printout queue 122 maintained by the document service 1. Each pending printout 128 has a priority 130, allowing higher-priority printouts to be printed before earlier-queued but lower-priority printouts.

The document service can be used to provide walk-up printing for documents which are not encoded with Netpage tags and retained.

In general, the token 126 may be any of a number of passive, semi-passive or active devices, including a surface or object bearing a Netpage tag pattern, linear barcode or two-dimensional barcode; a magnetic stripe card; a smart card or contact-less smart card; or a radio-frequency identification (RFID) tag. The reader 124 may be any reader matched to the type of the token 126, such as an optical reader utilising a scanning laser or a two-dimensional image sensor, as in conventional barcode readers or a Netpage sensing device; a magnetic stripe reader; a smart card reader; or an RFID reader.

As illustrated in Figure 18, in a first configuration the token reader 124 is associated with the printer 6 and the user presents the token 126 to the reader. The reader 124 reads the token 126 and communicates the walk-up event to the Netpage server 1. The walk-up event identifies both the user 62 and the printer 6. The token 126 and hence the walk-up event may identify the user 62 explicitly, or the server may be required to perform a database lookup to translate the token identifier into a user identifier. The reader and hence the walk-up event may identify the printer 6 explicitly, or the server 1 may be required to perform a database lookup to translate the reader identifier into a printer identifier. Once the server 1 has identified the user 62 and the printer 6, it retrieves printouts pending for the user and sends them to the printer to be printed.

Figure 18 shows the reader 124 and the printer 6 as separate devices which are physically associated. The reader 124 may be physically built into the printer 6. It may also be electrically connected to the printer, with the printer delivering the walk-up event to the server. Alternatively and equivalently, the printer 6 may interpret the walk-up event itself, and explicitly retrieve the user's pending printouts for printing.

The user token 126 may be attached to or built into a portable device which the user 62 carries, such as a mobile phone, pen, electronic pen (such as a Netpage pen 8), wallet, security access card or token, or identification badge or card. It may also be stand-alone and purpose-specific.

In the case of a Netpage pen 8, the printer reader 124 may provide a receptacle for receiving the pen, whereby the pen makes electrical contact and establishes a wired communication link (e.g. USB) with the reader to communicate the user identifier to the reader.

As illustrated in Figure 19, in a second configuration the token reader 124 is associated with the user 62 and the user presents the reader to the token 126. The reader 124 reads the token 126 and communicates the walk-up event to the Netpage server 1. The walk-up event identifies both the user 62 and the printer 6. The token 126 and hence the walk-up event may identify the printer 6 explicitly, or the server 1 may be required to perform a database lookup to translate the token identifier into a printer identifier. The reader 124 and hence the walk-up event may identify the user 62 explicitly, or the server 1 may be required to perform a database lookup to translate the reader identifier into a user identifier. Once the server 1 has identified the user 62 and the printer 6, it retrieves printouts pending for the user and sends them to the printer to be printed.

The printer token 126 may be attached to or built into the printer 6 or a device co-located with the printer.

As illustrated in Figure 20, even when the user 62 presents a token reader 125, it may be more convenient for the user reader 125 to rely on the communication link between the token reader 124 on the printer (or printer itself) and the server 1, since this communication link is guaranteed to be present. As in Figure 19, the user 62 presents the reader 125 to the token 127. The reader 125 reads the token 127. From the token it determines a short-range communication link to the printer 6. This may be a personal-area network (PAN) wireless link such as Bluetooth, wireless USB or ZigBee, or a local-area network (LAN) wireless link such as IEEE 802.11 (WiFi). It may also be a short-range optical link such as IrDA. Where the link requires a target address (such as in the case of Bluetooth), the token supplies the target address. For example, if the token 127 on the printer 6 uses a Netpage tag pattern, then the tag pattern encodes the target address instead of an impression ID, x-y location, etc., and flags it as such. Where the link does not require a target address (such as in the case of IrDA), the token 127 merely signals the user's token reader 126 to communicate a user identifier to the printer's token reader 126. Again, if the printer token uses a Netpage tag pattern, then the tag pattern flags the command to communicate the user identifier to the printer reader 124. If a range of communication link types are supported, then the token 127 (e.g. tag pattern) can identify a particular link type. The printer reader 124 receives the user identifier from the user reader 125 and communicates the walk-up event to the Netpage server 1. Once the server has identified the user 62 and the printer 6, it retrieves printouts pending for the user and sends them to the printer to be printed.

In the absence of a user token 126 or user reader 125, the user 62 may key a user identifier or job identifier into a keypad associated with the printer 6, with an optional password. The user 62 may also use a display-based input device associated with the printer to select their identity or their pending printout(s) from a list of users or jobs.

Netpage Explorer

As discussed above, the Netpage system acts as a virtual filing cabinet for any printed document. The Netpage system therefore provides users with a screen-based browser - the Netpage Explorer - for browsing and searching collections of printouts maintained by a document service, and for viewing individual printouts on-screen, including their digital ink. The Netpage Explorer also supports real-time display of streaming digital ink, and so provides a basis for remote conferencing.

As described above, the Netpage System supports the embedding of dynamic objects in documents, and the dynamic linking of dynamic objects to locations on printed impressions. The Netpage Explorer supports viewing of, and interaction with, such objects via the virtual view it provides of printed impressions, as well as the dynamic linking of such objects.

Product Variants

This section describes three Netpage product variants, each reflecting a different level of network distribution and access. Figure 17 shows a system using public Netpage services 134 running on a distributed set of servers on the public Internet 133, and serving applications 4 and users on the public Internet 133. Figure 21 shows a private Netpage system with services 136 (e.g. private Netpage document and digital ink services) running on one or more servers on a private intranet 138, and serving applications 4 and users on the private intranet. Figure 22 shows a personal Netpage system with services 142 running on a single personal computer or other personal device 140.

In each case, pre-printed Netpage content such as magazine adverts, catalogues, brochures, and product item Hyperlabels is typically hosted by public Netpage document services running on the Internet.

In private Netpage systems, security and privacy considerations may motivate the use of a private digital ink service even where the document service is public, as implied in Figures 21 and 22. A private document service may also act as a caching proxy for a public document service.

More generally, security and privacy considerations may motivate routing a user's digital ink to a constrained set of digital ink services, independent of the proliferation of document services. Some national governments may mandate that their citizens' digital ink be routed to national digital ink servers, even when interacting with international document services. A Netpage pen (or its relay) may therefore have knowledge of both a private and a public digital ink service, and may route digital ink pertaining to private impressions to the former and digital ink pertaining to public impressions to the latter. Even when a given pen's digital ink relates to a public impression and is nominally accessible on a public server, this need not imply that the owner of the impression or other users of the impression automatically gain access to that digital ink.

NETPAGE SURFACE CODING SECURITY

INTRODUCTION

The Netpage system uses a surface coding to imbue otherwise passive surfaces with interactivity in conjunction with Netpage sensing devices such as the Netpage pen and the Netpage viewer. When interacting with a Netpage coded surface, a Netpage sensing device generates a digital ink stream which indicates both the identity of the surface region relative to which the sensing device is moving, and the absolute path of the sensing device within the region.

This section defines optional authentication features of the Netpage surface coding, and associated authentication protocols.

SURFACE CODING SECURITY

Surface Coding Background

The Netpage surface coding consists of a dense planar tiling of tags. Each tag encodes its own location in the plane. Each tag also encodes, in conjunction with adjacent tags, an identifier of the region containing the tag.

This region ID is unique among all regions. In the Netpage system the region typically corresponds to the entire extent of the tagged surface, such as one side of a sheet of paper. In the Hyperlabel system the region typically corresponds to the surface of an entire product item, and the region ID corresponds to the unique item ID. For clarity in the following discussion, references to items and item IDs (or simply IDs), correspond to the region ID.

The surface coding is designed so that an acquisition field of view large enough to guarantee acquisition of an entire tag is large enough to guarantee acquisition of the ID of the region containing the tag. Acquisition of the tag itself guarantees acquisition of the tag's two-dimensional position within the region, as well as other tag-specific data. The surface coding therefore allows a sensing device to acquire a region ID and a tag position during a purely local interaction with a coded surface, e.g. during a "click" or tap on a coded surface with a pen.

Cryptography Background

Cryptography is used to protect sensitive information, both in storage and in transit, and to authenticate parties to a transaction. There are two classes of cryptography in widespread use: secret-key cryptography and public-key cryptography. The Netpage and Hyperlabel systems use both classes of cryptography.

Secret-key cryptography, also referred to as symmetric cryptography, uses the same key to encrypt and decrypt a message. Two parties wishing to exchange messages must first arrange to securely exchange the secret key.

Public-key cryptography, also referred to as asymmetric cryptography, uses two encryption keys. The two keys are mathematically related in such a way that any message encrypted using one key can only be decrypted using the other key. One of these keys is then published, while the other is kept private. They are referred to as the public and private key respectively. The public key is used to encrypt any message intended for the holder of the private key. Once encrypted using the public key, a message can only be decrypted using the private key. Thus two parties can securely exchange messages without first having to exchange a secret key. To ensure that the private key is secure, it is normal for the holder of the private key to generate the public-private key pair.

Public-key cryptography can be used to create a digital signature. If the holder of the private key creates a known hash of a message and then encrypts the hash using the private key, then anyone can verify that the encrypted hash constitutes the "signature" of the holder of the private key with respect to that particular message, simply by decrypting the encrypted hash using the public key and verifying the hash against the message. If the signature is appended to the message, then the recipient of the message can verify both that the message is genuine and that it has not been altered in transit.

Secret-key can also be used to create a digital signature, but has the disadvantage that signature verification can also be performed by a party privy to the secret key.

To make public-key cryptography work, there has to be a way to distribute public keys which prevents impersonation. This is normally done using certificates and certificate authorities. A certificate authority is a trusted third party which authenticates the association between a public key and a person's or other entity's identity. The certificate authority verifies the identity by examining identity documents etc., and then creates and signs a digital certificate containing the identity details and public key. Anyone who trusts the certificate authority can use the public key in the certificate with a high degree of certainty that it is genuine. They just

have to verify that the certificate has indeed been signed by the certificate authority, whose public key is well-known.

To achieve comparable security to secret-key cryptography, public-key cryptography utilises key lengths an order of magnitude larger, i.e. a few thousand bits compared with a few hundred bits.

For a detailed discussion of cryptographic techniques, see Schneier, B., *Applied Cryptography*, Second Edition, John Wiley & Sons 1996.

Security Requirements

We define item security to have two related purposes:

- to allow authentication of an item
- to prevent forgery of an item

The greater the difficulty of forgery, the greater the trustworthiness of authentication.

When an item is coded, Netpage surface coding security has two corresponding purposes:

- to allow authentication of a coded item
- to prevent forgery of a coded item with a novel item ID

If a user is able to determine the authenticity of the surface coding of an item, then the user may be able to make an informed decision about the likely authenticity of the item.

If it is intractable to forge the surface coding for a novel ID, then the only tractable way of forging an item with an authentic surface coding is to duplicate the surface coding of an existing item (and hence its ID). If the user is able to determine by other means that the ID of an item is likely to be unique, then the user may assume that the item is authentic.

Since the Netpage surface coding allows meaningful interaction between a sensing device and a coded surface during a purely local interaction, it is desirable for the surface coding to support authentication during a similarly local interaction, i.e. without requiring an increase in the size of the sensing device field of view.

Since no *a priori* relationship exists between creators of authentic coded items and users potentially wishing to authenticate such items, it is undesirable to require a trust relationship between creators and users. For example, it is undesirable to require that creators share secret signature keys with users.

It is reasonable for many users to rely on online access to an authenticator trusted by a creator for the purposes of authenticating items. Conversely, it is desirable to allow authentication to take place in the absence of online access.

Security Discussion

As discussed above in 'Cryptography Background', authentication relies on verifying the correspondence between data and a signature of that data. The greater the difficulty in forging a signature, the greater the trustworthiness of signature-based authentication.

The item ID is unique and therefore provides a basis for a signature. If online authentication access is assumed, then the signature may simply be a random number associated with the item ID in an authentication database accessible to the trusted online authenticator. The random number may be generated by any suitable method, such as via a deterministic (pseudo-random) algorithm, or via a stochastic physical process.

A keyed hash or encrypted hash may be preferable to a random number since it requires no additional space in the authentication database.

In the limit case no signature is actually required, since the mere presence of the item ID in the database indicates authenticity. However, the use of a signature limits a forger to forging items he has actually sighted.

To prevent forgery of a signature for an unsighted ID, the signature must be large enough to make exhaustive search via repeated accesses to the online authenticator intractable. If generated using a key rather than randomly, then the length of the signature must also be large enough to prevent the forger from deducing the key from known ID-signature pairs. Signatures of a few hundred bits are considered secure, whether generated using private or secret keys.

Limited space within the surface coding tag structure makes it impractical to include a secure signature in a tag. We are therefore motivated to distribute fragments of a signature across multiple tags. If each fragment can be verified in isolation against the ID, then the goal of supporting authentication without increasing the sensing device field of view is achieved. The security of the signature still derives from the full length of the signature rather than from the length of a fragment, since a forger cannot predict which fragment a user will randomly choose to verify. Note that a trusted authenticator can always perform fragment verification, so fragment verification is always possible when online access to a trusted authenticator is available.

Fragment verification requires fragment identification. Fragments may be explicitly numbered, or may more economically be identified by the two-dimensional coordinate of their tag, modulo the repetition of the signature across a continuous tiling of tags.

The limited length of the ID itself introduces a further vulnerability. Ideally it should be at least a few hundred bits. In the Netpage and Hyperlabel surface coding schemes it is 96 bits or less. To overcome this, the ID may be padded. For this to be effective the padding must be variable, i.e. it must vary from one ID to the next. Ideally the padding is simply a random number, and must then be stored in the authentication database indexed by ID. If the padding is deterministically generated from the ID then it is worthless.

Offline authentication of secret-key signatures requires the use of a trusted offline authentication device. The QA chip (see US 6,374,354, issued 16 April 2002) provides the basis for such a device, although of limited capacity. The QA chip can be programmed to verify a signature using a secret key securely held in its internal memory. In this scenario, however, it is impractical to support per-ID padding, and it is impractical even to support more than a very few secret keys. Furthermore, a QA chip programmed in this manner is susceptible to a chosen-message attack. These constraints limit the applicability of a QA-chip-based trusted offline authentication device to niche applications.

In general, despite the claimed security of any particular trusted offline authentication device, creators of secure items are likely to be reluctant to entrust their secret signature keys to such devices, and this is again likely to limit the applicability of such devices to niche applications.

By contrast, offline authentication of public-key signatures (i.e. generated using the corresponding private keys) is highly practical. An offline authentication device utilising public keys can trivially hold any number of public keys, and may be designed to retrieve additional public keys on demand, via a transient online connection, when it encounters an ID for which it knows it has no corresponding public signature key. Untrusted offline authentication is likely to be attractive to most creators of secure items, since they are able to retain exclusive control of their private signature keys.

A disadvantage of offline authentication of a public-key signature is that the entire signature must be acquired from the coding, violating our desire to support authentication with a minimal field of view. A

corresponding advantage of offline authentication of a public-key signature is that access to the ID padding is no longer required, since decryption of the signature using the public signature key generates both the ID and its padding, and the padding can then be ignored.

Acquisition of an entire distributed signature is not particularly onerous. Any random or linear swipe of a hand-held sensing device across a coded surface allows it to quickly acquire all of the fragments of the signature. The sensing device can easily be programmed to signal the user when it has acquired a full set of fragments and has completed authentication. A scanning laser can also easily acquire all of the fragments of the signature. Both kinds of devices may be programmed to only perform authentication when the tags indicate the presence of a signature.

Note that a public-key signature may be authenticated online via any of its fragments in the same way as any signature, whether generated randomly or using a secret key. The trusted online authenticator may generate the signature on demand using the private key and ID padding, or may store the signature explicitly in the authentication database. The latter approach obviates the need to store the ID padding.

Note also that signature-based authentication may be used in place of fragment-based authentication even when online access to a trusted authenticator is available.

Security Specification

Setup per ID range:

- generate public-private signature key pair
- store key pair indexed by ID range

Setup per ID:

- generate ID padding
- retrieve private signature key by ID
- generate signature by encrypting ID and padding using private key
- store signature in database indexed by ID
- encode signature across multiple tags in repeated fashion

Online (fragment-based) authentication (user):

- acquire ID from tags
- acquire position and signature fragment from tag
- generate fragment number from position
- look up trusted authenticator by ID
- transmit ID, fragment and fragment number to trusted authenticator

Online (fragment-based) authentication (trusted authenticator):

- receive ID, fragment and fragment number from user
- retrieve signature from database by ID
- compare supplied fragment with signature
- report authentication result to user

Offline (signature-based) authentication (user):

- acquire ID from tags
- acquire positions and signature fragments from tags
- generate signature from fragments

- retrieve public signature key by ID
- decrypt signature using public key
- compare acquired ID with decrypted ID
- report authentication result to user

NETPAGE SURFACE CODING

INTRODUCTION

This section defines a surface coding used by the Netpage system (described above in 'Netpage Architecture') to imbue otherwise passive surfaces with interactivity in conjunction with Netpage sensing devices such as the Netpage pen and the Netpage viewer.

When interacting with a Netpage coded surface, a Netpage sensing device generates a digital ink stream which indicates both the identity of the surface region relative to which the sensing device is moving, and the absolute path of the sensing device within the region.

SURFACE CODING

The Netpage surface coding consists of a dense planar tiling of tags. Each tag encodes its own location in the plane. Each tag also encodes, in conjunction with adjacent tags, an identifier of the region containing the tag. In the Netpage system, the region typically corresponds to the entire extent of the tagged surface, such as one side of a sheet of paper.

Each tag is represented by a pattern which contains two kinds of elements. The first kind of element is a target. Targets allow a tag to be located in an image of a coded surface, and allow the perspective distortion of the tag to be inferred. The second kind of element is a macrodot. Each macrodot encodes the value of a bit by its presence or absence.

The pattern is represented on the coded surface in such a way as to allow it to be acquired by an optical imaging system, and in particular by an optical system with a narrowband response in the near-infrared. The pattern is typically printed onto the surface using a narrowband near-infrared ink.

TAG STRUCTURE

Figure 23 shows the structure of a complete tag 200. Each of the four black circles 202 is a target. The tag 200, and the overall pattern, has four-fold rotational symmetry at the physical level.

Each square region represents a symbol 204, and each symbol represents four bits of information. Each symbol 204 shown in the tag structure has a unique label 216. Each label 216 has an alphabetic prefix and a numeric suffix.

Figure 24 shows the structure of a symbol 204. It contains four macrodots 206, each of which represents the value of one bit by its presence (one) or absence (zero).

The macrodot 206 spacing is specified by the parameter s throughout this specification. It has a nominal value of $143\mu m$, based on 9 dots printed at a pitch of 1600 dots per inch. However, it is allowed to vary within defined bounds according to the capabilities of the device used to produce the pattern.

Figure 25 shows an array 208 of nine adjacent symbols 204. The macrodot 206 spacing is uniform both within and between symbols 208.

Figure 26 shows the ordering of the bits within a symbol 204.

Bit zero 210 is the least significant within a symbol 204; bit three 212 is the most significant. Note that this ordering is relative to the orientation of the symbol 204. The orientation of a particular symbol 204 within the tag 200 is indicated by the orientation of the label 216 of the symbol in the tag diagrams (see for example Figure 23). In general, the orientation of all symbols 204 within a particular segment of the tag 200 is the same, consistent with the bottom of the symbol being closest to the centre of the tag.

Only the macrodots 206 are part of the representation of a symbol 204 in the pattern. The square outline 214 of a symbol 204 is used in this specification to more clearly elucidate the structure of a tag 204. Figure 27, by way of illustration, shows the actual pattern of a tag 200 with every bit 206 set. Note that, in practice, every bit 206 of a tag 200 can never be set.

A macrodot 206 is nominally circular with a nominal diameter of $(5/9)s$. However, it is allowed to vary in size by $\pm 10\%$ according to the capabilities of the device used to produce the pattern.

A target 202 is nominally circular with a nominal diameter of $(17/9)s$. However, it is allowed to vary in size by $\pm 10\%$ according to the capabilities of the device used to produce the pattern.

The tag pattern is allowed to vary in scale by up to $\pm 10\%$ according to the capabilities of the device used to produce the pattern. Any deviation from the nominal scale is recorded in the tag data to allow accurate generation of position samples.

TAG GROUPS

Tags 200 are arranged into tag groups 218. Each tag group contains four tags arranged in a square. Each tag 200 has one of four possible tag types, each of which is labelled according to its location within the tag group 218. The tag type labels 220 are 00, 10, 01 and 11, as shown in Figure 28.

Figure 29 shows how tag groups are repeated in a continuous tiling of tags, or tag pattern 222. The tiling guarantees the any set of four adjacent tags 200 contains one tag of each type 220.

CODEWORDS

The tag contains four complete codewords. The layout of the four codewords is shown in Figure 30. Each codeword is of a punctured 2^4-ary $(8, 5)$ Reed-Solomon code. The codewords are labelled A, B, C and D. Fragments of each codeword are distributed throughout the tag 200.

Two of the codewords are unique to the tag 200. These are referred to as local codewords 224 and are labelled A and B. The tag 200 therefore encodes up to 40 bits of information unique to the tag.

The remaining two codewords are unique to a tag type, but common to all tags of the same type within a contiguous tiling of tags 222. These are referred to as global codewords 226 and are labelled C and D, subscripted by tag type. A tag group 218 therefore encodes up to 160 bits of information common to all tag groups within a contiguous tiling of tags.

REED-SOLOMON ENCODING

Codewords are encoded using a punctured 2^4-ary $(8, 5)$ Reed-Solomon code. A 2^4-ary $(8, 5)$ Reed-Solomon code encodes 20 data bits (i.e. five 4-bit symbols) and 12 redundancy bits (i.e. three 4-bit symbols) in each codeword. Its error-detecting capacity is three symbols. Its error-correcting capacity is one symbol.

Figure 31 shows a codeword 228 of eight symbols 204, with five symbols encoding data coordinates 230 and three symbols encoding redundancy coordinates 232. The codeword coordinates are indexed in coefficient order, and the data bit ordering follows the codeword bit ordering.

A punctured 2^4-ary $(8, 5)$ Reed-Solomon code is a 2^4-ary $(15, 5)$ Reed-Solomon code with seven redundancy coordinates removed. The removed coordinates are the most significant redundancy coordinates. The code has the following primitive polynominal:

$$p(x) = x^4 + x + 1 \quad \text{(EQ 1)}$$

The code has the following generator polynominal:

$$g(x) = (x + \alpha)(x + \alpha^2)\ldots(x + \alpha^{10}) \quad \text{(EQ 2)}$$

For a detailed description of Reed-Solomon codes, refer to Wicker, S.B. and V.K. Bhargava, eds., *Reed-Solomon Codes and Their Applications*, IEEE Press, 1994, the contents of which are incorporated herein by reference.

THE TAG COORDINATE SPACE

The tag coordinate space has two orthogonal axes labelled x and y respectively. When the positive x axis points to the right, then the positive y axis points down.

The surface coding does not specify the location of the tag coordinate space origin on a particular tagged surface, nor the orientation of the tag coordinate space with respect to the surface. This information is application-specific. For example, if the tagged surface is a sheet of paper, then the application which prints the tags onto the paper may record the actual offset and orientation, and these can be used to normalise any digital ink subsequently captured in conjunction with the surface.

The position encoded in a tag is defined in units of tags. By convention, the position is taken to be the position of the centre of the target closest to the origin.

TAG INFORMATION CONTENT

Table 1 defines the information fields embedded in the surface coding. Table 2 defines how these fields map to codewords.

Table 1. Field definitions

field	width	description
per codeword		
codeword type	2	The type of the codeword, i.e. one of A (b'00'), B (b'01'), C (b'10') and D (b'11').

per tag		
tag type	2	The type[1] of the tag, i.e. one of 00 (b'00'), 01 (b'01'), 10 (b'10') and 11 (b'11').
x coordinate	13	The unsigned x coordinate of the tag[2].
y coordinate	13	The unsigned y coordinate of the tag[b].
active area flag	1	A flag indicating whether the tag is a member of an active area. b'1' indicates membership.
active area map flag	1	A flag indicating whether an active area map is present. b'1' indicates the presence of a map (see next field). If the map is absent then the value of each map entry is derived from the active area flag (see previous field).
active area map	8	A map[3] of which of the tag's immediate eight neighbours are members of an active area. b'1' indicates membership.
data fragment	8	A fragment of an embedded data stream. Only present if the active area map is absent.
per tag group		
encoding format	8	The format of the encoding. 0: the present encoding Other values are TBA.
region flags	8	Flags controlling the interpretation and routing of region-related information. 0: region ID is an EPC 1: region is linked 2: region is interactive 3: region is signed 4: region includes data 5: region relates to mobile application Other bits are reserved and must be zero.
tag size adjustment	16	The difference between the actual tag size and the nominal tag size[4], in 10nm units, in sign-magnitude format.

[1]corresponds to the bottom two bits of the x and y coordinates of the tag

[2]allows a maximum coordinate value of approximately 14m

[3]Figure 29 indicates the bit ordering of the map

[4]the nominal tag size is 1.7145mm (based on 1600dpi, 9 dots per macrodot, and 12 macrodots per tag)

region ID	96	The ID of the region containing the tags.
CRC	16	A CRC[5] of tag group data.
total	320	

Figure 32 shows a tag 200 and its eight immediate neighbours, each labelled with its corresponding bit index in the active area map. An active area map indicates whether the corresponding tags are members of an active area. An active area is an area within which any captured input should be immediately forwarded to the corresponding Netpage server for interpretation. It also allows the Netpage sensing device to signal to the user that the input will have an immediate effect.

Table 2. Mapping of fields to codewords

codeword	codeword bits	field	width	field bits
A	1:0	codeword type (b'00')	2	all
	10:2	x coordinate	9	12:4
	19:11	y coordinate	9	12:4
B	1:0	codeword type (b'01')	2	all
	2	tag type	1	0
	5:2	x coordinate	4	3:0
	6	tag type	1	1
	9:6	y coordinate	4	3:0
	10	active area flag	1	all
	11	active area map flag	1	all
	19:12	active area map	8	all
	19:12	data fragment	8	all
C_{00}	1:0	codeword type (b'10')	2	all
	9:2	encoding format	8	all
	17:10	region flags	8	all
	19:18	tag size adjustment	2	1:0

[5]CCITT CRC-16 [7]

C_{01}	1:0	codeword type (b'10')	2	all
	15:2	tag size adjustment	14	15:2
	19:16	region ID	4	3:0
C_{10}	1:0	codeword type (b'10')	2	all
	19:2	region ID	18	21:4
C_{11}	1:0	codeword type (b'10')	2	all
	19:2	region ID	18	39:22
D_{00}	1:0	codeword type (b'11')	2	all
	19:2	region ID	18	57:40
D_{01}	1:0	codeword type (b'11')	2	all
	19:2	region ID	18	75:58
D_{10}	1:0	codeword type (b'11')	2	all
	19:2	region ID	18	93:76
D_{11}	1:0	codeword type (b'11')	2	all
	3:2	region ID	2	95:94
	19:4	CRC	16	all

Note that the tag type can be moved into a global codeword to maximise local codeword utilization. This in turn can allow larger coordinates and/or 16-bit data fragments (potentially configurably in conjunction with coordinate precision). However, this reduces the independence of position decoding from region ID decoding and has not been included in the specification at this time.

EMBEDDED DATA

If the "region includes data" flag in the region flags is set then the surface coding contains embedded data. The data is encoded in multiple contiguous tags' data fragments, and is replicated in the surface coding as many times as it will fit.

The embedded data is encoded in such a way that a random and partial scan of the surface coding containing the embedded data can be sufficient to retrieve the entire data. The scanning system reassembles the data from retrieved fragments, and reports to the user when sufficient fragments have been retrieved without error.

As shown in Table 3, a 200-bit data block encodes 160 bits of data. The block data is encoded in the data fragments of A contiguous group of 25 tags arranged in a 5×5 square. A tag belongs to a block whose integer coordinate is the tag's coordinate divided by 5. Within each block the data is arranged into tags with increasing x coordinate within increasing y coordinate.

A data fragment may be missing from a block where an active area map is present. However, the missing data fragment is likely to be recoverable from another copy of the block.

Data of arbitrary size is encoded into a superblock consisting of a contiguous set of blocks arranged in a rectangle. The size of the superblock is encoded in each block. A block belongs to a superblock whose integer coordinate is the block's coordinate divided by the superblock size. Within each superblock the data is arranged into blocks with increasing x coordinate within increasing y coordinate.

The superblock is replicated in the surface coding as many times as it will fit, including partially along the edges of the surface coding.

The data encoded in the superblock may include more precise type information, more precise size information, and more extensive error detection and/or correction data.

Table 3. Embedded data block

field	width	description
data type	8	The type of the data in the superblock. Values include: 0: type is controlled by region flags 1: MIME Other values are TBA.
superblock width	8	The width of the superblock, in blocks.
superblock height	8	The height of the superblock, in blocks.
data	160	The block data.
CRC	16	A CRC[6] of the block data.
total	200	

CRYPTOGRAPHIC SIGNATURE OF REGION ID

If the "region is signed" flag in the region flags is set then the surface coding contains a 160-bit cryptographic signature of the region ID. The signature is encoded in a one-block superblock.

In an online environment any signature fragment can be used, in conjunction with the region ID, to validate the signature. In an offline environment the entire signature can be recovered by reading multiple tags, and

[6]CCITT CRC-16 [7]

can then be validated using the corresponding public signature key. This is discussed in more detail in Netpage Surface Coding Security section above.

MIME DATA

If the embedded data type is "MIME" then the superblock contains Multipurpose Internet Mail Extensions (MIME) data according to RFC 2045 (see Freed, N., and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) - Part One: Format of Internet Message Bodies", RFC 2045, November 1996), RFC 2046 (see Freed, N., and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) - Part Two: Media Types", RFC 2046, November 1996) and related RFCs. The MIME data consists of a header followed by a body. The header is encoded as a variable-length text string preceded by an 8-bit string length. The body is encoded as a variable-length type-specific octet stream preceded by a 16-bit size in big-endian format.

The basic top-level media types described in RFC 2046 include text, image, audio, video and application. RFC 2425 (see Howes, T., M. Smith and F. Dawson, "A MIME Content-Type for Directory Information", RFC 2045, September 1998) and RFC 2426 (see Dawson, F., and T. Howes, "vCard MIME Directory Profile", RFC 2046, September 1998) describe a text subtype for directory information suitable, for example, for encoding contact information which might appear on a business card.

ENCODING AND PRINTING CONSIDERATIONS

The Print Engine Controller (PEC) supports the encoding of two fixed (per-page) 2^4-ary $(15, 5)$ Reed-Solomon codewords and six variable (per-tag) 2^4-ary $(15, 5)$ Reed-Solomon codewords. Furthermore, PEC supports the rendering of tags via a rectangular unit cell whose layout is constant (per page) but whose variable codeword data may vary from one unit cell to the next. PEC does not allow unit cells to overlap in the direction of page movement.

A unit cell compatible with PEC contains a single tag group consisting of four tags. The tag group contains a single A codeword unique to the tag group but replicated four times within the tag group, and four unique B codewords. These can be encoded using five of PEC's six supported variable codewords. The tag group also contains eight fixed C and D codewords. One of these can be encoded using the remaining one of PEC's variable codewords, two more can be encoded using PEC's two fixed codewords, and the remaining five can be encoded and pre-rendered into the Tag Format Structure (TFS) supplied to PEC.

PEC imposes a limit of 32 unique bit addresses per TFS row. The contents of the unit cell respect this limit. PEC also imposes a limit of 384 on the width of the TFS. The contents of the unit cell respect this limit. Note that for a reasonable page size, the number of variable coordinate bits in the A codeword is modest, making encoding via a lookup table tractable. Encoding of the B codeword via a lookup table may also be possible. Note that since a Reed-Solomon code is systematic, only the redundancy data needs to appear in the lookup table.

IMAGING AND DECODING CONSIDERATIONS

The minimum imaging field of view required to guarantee acquisition of an entire tag has a diameter of $39.6s$ (i.e. $(2 \times (12+2))\sqrt{2}s$), allowing for arbitrary alignment between the surface coding and the field of view. Given a macrodot spacing of $143\mu m$, this gives a required field of view of 5.7mm.

Table 4 gives pitch ranges achievable for the present surface coding for different sampling rates, assuming an image sensor size of 128 pixels.

Table 4. Pitch ranges achievable for present surface coding for different sampling rates; dot pitch = 1600dpi, macrodot pitch = 9 dots, viewing distance = 30mm, nib-to-FOV separation = 1mm, image sensor size = 128 pixels

sampling rate	pitch range
2	−40 to +49
2.5	−27 to +36
3	−10 to +18

Given the present surface coding, the corresponding decoding sequence is as follows:

- locate targets of complete tag
- infer perspective transform from targets
- sample and decode any one of tag's four codewords
- determine codeword type and hence tag orientation
- sample and decode required local (A and B) codewords
- codeword redundancy is only 12 bits, so only detect errors
- on decode error flag bad position sample
- determine tag x-y location, with reference to tag orientation
- infer 3D tag transform from oriented targets
- determine nib x-y location from tag x-y location and 3D transform
- determine active area status of nib location with reference to active area map
- generate local feedback based on nib active area status
- determine tag type from A codeword
- sample and decode required global (C and D) codewords (modulo window alignment, with reference to tag type)
- although codeword redundancy is only 12 bits, correct errors; subsequent CRC verification will detect erroneous error correction
- verify tag group data CRC
- on decode error flag bad region ID sample
- determine encoding type, and reject unknown encoding
- determine region flags
- determine region ID

- encode region ID, nib x-y location, nib active area status in digital ink
- route digital ink based on region flags

Note that region ID decoding need not occur at the same rate as position decoding.

Note that decoding of a codeword can be avoided if the codeword is found to be identical to an already-known good codeword.

The above description is purely illustrative and the skilled worker in this field will readily recognize many variations and modifications that do not depart from the spirit and scope of the broad inventive concept.

We Claim:

1. A computer network for a plurality of users, the computer network comprising:

a server;

a printer;

a network user identifier for a network user to carry on their person; and,

a printer identifier associated with the printer; wherein during use,

the network user identifier and the printer identifier interact such that any of the network user's pending printouts are sent to the printer for printing when the network user is proximate the printer.

2. A computer network according to claims 1 wherein the network has a plurality of said printers, each printer associated with one of the printer identifiers respectively; and

a plurality of said network user identifiers, each uniquely identifying different network users.

3. A computer network according to claims 2 wherein each of the network user identifiers is a token and each of the printer identifiers has a token reader such that the user presents their token to the token reader associated with one of the printers to request actual printing of their queued printouts via that printer.

4. A computer network according to claims 3 wherein the tokens are a short-range RFID tag, a smartcard or a magnetic stripe card.

5. A computer network according to claims 4 wherein the token reader notifies the server of the user's proximity to the associated printer which in turn initiates printing.

6. A computer network according to claims 2 wherein each of the printer identifiers is a token and each of the network user identifiers has a token reader associated with the user.

7. A computer network according to claims 6 wherein the token reader is an electronic stylus with an optical sensor, and the tokens are a surface on each of the printers with coded data disposed on it, the coded data being readable by the optical sensors of each user's electronic stylus.

8. A computer network according to claims 1 wherein the pending printouts are maintained in a queue by the server and each pending printout has a priority such that higher-priority printouts are printed before earlier-queued but lower-priority printouts.

9. A computer network according to claims 3 wherein the token readers identify both the user and the printer to the server when the user presents their token to the reader.

10. A computer network according to claims 9 wherein the token identfies the user explicitly.

11. A computer network according to claims 9 wherein the token has a token identifier and the server performs a database lookup to translate the token identifier into a user identity.

12. A computer network according to claims 9 wherein the token reader identifies the printer explicitly.

13. A computer network according to claims 9 wherein the reader has a reader identifier and the server performs a database lookup to translate the reader identifier into a printer indentity.

14. A computer network according to claims 3 wherein the token reader and the printer are separate devices which have an electrical connection.

15. A computer network according to claims 3 wherein the token reader is physically built into the printer.

16. A computer network according to claims 3 wherein the token reader informs the printer that the user has presented a token and the printer then explicitly retrieves the user's pending printouts for printing.

17. A computer network according to claims 3 wherein the token is a security access or identification badge or card.