Poseidon House
Castle Park
Cambridge CB3 0RD
United Kingdom

TELEPHONE: **Cambridge (0223) 323010**
INTERNATIONAL: **+44 223 323010**
FAX: **+44 223 359779**
E-MAIL: **apm@ansa.co.uk**

ANSA Phase III

IPS - An Information Publishing System

Ed Oskiewicz, Nigel Edwards

Abstract

IPS is a distributed information publishing system, it explores issues which arise when publishing information for commercial gain within a world wide web-like hypertext system. The objective in designing IPS is to understand how to try to make such a system dependable. This paper gives a problem statement and identifies the major components of IPS. It is particularly concerned with the mechanisms by which information is accessed, issues such as charging policies are also introduced.

This document was written as an internal document to record progress in the design of an application scenario. Comments on any aspect of it would be appreciated.

Chapter 1 describes the motivation for this work and describes the problems which need to be solved. A brief description of the functionality of IPS is given. Chapter 2 identifies the major components and explains how they interact. Finally chapter 3 gives an object model using Rumbaugh's Object Modelling Technique. This explains the major responsibilities of the components and further explores the relationships between them.

APM.1171.00.06 **Draft** 25 April 1995

Request for Comments (confidential to ANSA consortium for 2 years)

Distribution:
Supersedes: APM.1096
Superseded by:

IPS - An Information Publishing System

IPS - An Information Publishing System

Ed Oskiewicz, Nigel Edwards

APM.1171.00.06

25 April 1995

The material in this Report has been developed as part of the ANSA Architecture for Open Distributed Systems. ANSA is a collaborative initiative, managed by Architecture Projects Management Limited on behalf of the companies sponsoring the ANSA Workprogramme.

The ANSA initiative is open to all companies and organisations. Further information on the ANSA Workprogramme, the material in this report, and on other reports can be obtained from the address below.

The authors acknowledge the help and assistance of their colleagues, in sponsoring companies and the ANSA team in Cambridge in the preparation of this report.

Architecture Projects Management Limited

Poseidon House
Castle Park
CAMBRIDGE
CB3 0RD
United Kingdom

TELEPHONE UK	(01223) 515010
INTERNATIONAL	+44 1223 515010
FAX	+44 1223 359779
E-MAIL	apm@ansa.co.uk

Contents

1 IPS

IPS is a distributed information publishing system. Our objective in designing such a system is to understand the options for making such a system dependable. This paper gives a problem statement and identifies the major components of IPS. It is particularly concerned with the mechanisms by which information is accessed. Future papers will explore in more detail the engineering components needed to make these mechanisms dependable; others will look at how information is published and updated and how this is made dependable.

The design presented here, is a refinement of the design presented in [EDWARDS 94a]. This early design was analysed in [EDWARDS 94b]. This resulted in some modifications to improve the dependability (for example the introduction of the audit service).

The remainder of this paper is structured as follows. Chapter 1 describes the motivation for this work and describes the problems which need to be solved. A brief description of the functionality of IPS is given. Chapter 2 identifies the major components and explains how they interact. Finally chapter 3 gives an object model using a methodology based on that in [RUMBAUGH 91]. This explains the major responsibilities of the components and further explores the relationships between them.

1.1 Motivation

The aim of designing IPS is to investigate solutions to several generic service management problems and also to help develop a methodology for developing dependable systems. The service management problems contained in this scenario include:

- publishing information
- controlling the availability and consistency of published information
- charging for accessing published information and maintaining the integrity and availability and consistency of the charging information
- controlling the copying of published information

The information units (unit of publication) can be regarded as network elements.

The World Wide Web (WWW) already has some of the required functionality: it is a distributed hypermedia system. In the WWW it is common for information services to go off the air and bounce requests. There is no attempt to automatically maintain consistency: it is possible to find that a hyperlink is pointing at outdated or unavailable information. The WWW (and in particular the Mosaic user interface) can be used to demonstrate a possible user interface, freeing us to concentrate on dependability issues.

This application turns information into commodity which is traded electronically. This means that dependability becomes very important especially in addressing issues of vendors charging for information, updating information (possibly frequently) and maintaining the consistency of hyperlinks and a rapidly changing dynamic environment. These are issues which are not addressed in the worldwide web (but then that is free).

By including audio and video and allowing customised information packages to be built, this scenario subsumes the video on demand and “ME-TV” concepts which are being discussed by the industry. “ME-TV” is analogous to putting together your own information package using (video) information which is published by other suppliers. Alternatively a software agent could put the package together for you [HEILEMANN 94].

1.2 Problem Statement

The objective is to develop a distributed information system in which different organisations can publish information electronically using hypermedia.

Typical of such organisations is a market research company which wants to publish information. Customers should be able to browse through what is available. The charges for browsing the information will be related to its value. So a small charge (nothing) is made for browsing the top level description of what is available. Viewing more detailed (and valuable) information is more expensive. Each unit of information has a cost associated with it.

The information is the company's primary source of revenue, so it is important that it is highly available. In a global information market it is not feasible to take down information servers to reconfigure them: somebody somewhere will want to use it (and pay for it). This means they need to be available 7 days a week 24 hours a day, so there is no planned unavailability.

Integrity of the information is also vital it must not be corrupted under any circumstances. The billing information must not be lost and must relate accurately the information which the customer has accessed. Some of this information may be maintained or available from the systems environment (e.g. account databases), other information must be maintained explicitly by the system itself.

As well as reflecting the value of the information provided (server specific charging), charges needs to be made on the basis of the identity of the person accessing the information (client specific charging). It would be unusual for the owner of information to be charged for accessing it. In addition some clients may be entitled to discounts because of the high volume of information they have accessed. Charging options are explored separately in §1.3. For the initial design charging information is maintained by the system and periodically written to an accounts database. Billing and payment is outside of the scope of the system. Other, more sophisticated, payment options could be used (see §1.4).

Facilities need to be available for publishing information and updating or withdrawing existing information. Note that when information is updated, it may be referenced and will have references to other information. These references need to be kept up to date.

What constitutes information could vary in granularity from a book to a report of the latest share price. Information may be in the form of various kinds of media: text, image, video or audio. Large units of information will be updated infrequently, small ones may be updated very frequently (order of minutes). Very new, information may be charged at a premium rate: the fresher it is the more it is worth.

Some customers need to be notified when information is updated (e.g. share price). This taken together with the notion of customers having contexts (see §1.4), allows customised information packages to be built. For example, a business analyst specializing in the pharmaceutical industry would have the latest press reports and share prices of the industry available. Certain events (e.g. rapid share price change or year-end figure publication) might trigger alarms to bring it to the analyst's attention. Use of the notifier might be chargeable.

1.3 Charging options

Client needs to know what accesses are likely to cost, some options include: published rate card, query before read, etc. Servers are expected to charge no more than they quote. A client's browser may be programmed to display a running total of costs amassed so far and also set up to authorise automatically accesses within certain limits or pertaining to certain categories.

Payment technology is either cash, e.g. coin slot in browser (which must be trusted to be honest) or credit: the user may have an account with periodic settlement or quote a credit card to debit. Whatever method is used the server still needs to check that credit limits have not been exceeded or prepayments used up.

1.3.1 Payment styles

- Prepayment - just saves on external credit checks?
- Pay per view - coin-operated or debit users credit card for each access.
- Pay in arrears - send the user periodic statements, suspicious users will keep their own access records for reconciliation.

1.3.2 Charge granularity

Revenue is derived from selling **access to** information to the user. The principle should be that if clients get something of value they should have to pay for it.

Possible options for charging granularity include:

- Whole InformationUnit charging — when the InformationUnit is small or if the provider wants to keep things simple.
- Inter InformationUnit charging — browse within an InformationUnit and pay for what you look at.

Time based charging — for stream based InformationUnits (e.g. audio or video) it will be simpler to charge based on time units (actually you'd be charging for bits sent down the wire?) A useful feature would be to consume the stream over multiple sessions with the system remembering where you left off, e.g combine this with coin-operated prepayment.

1.3.3 Charging variations

Not all InformationUnits will cost the same amount nor will all clients be charged identically for accessing a given InformationUnit. The possibilities seem to be to base discounts on who the user is, what the InformationUnit is, what the user has done already. The charging policy should be as simple and uniform as possible otherwise users will get confused (c.f. air fares). Also some items may attract a sales tax, others may not. Here are some possibilities.

- Client specific discounts: some clients may have an automatic discount applied to some of their accesses. Also, clients may offer to pay less than the published price.
- Bulk access discounts: accesses get cheaper the more you make, could be banded with accesses becoming progressively cheaper.
- 3 for the price of 2 discount schemes: occasional accesses are offered at a discount — these may be time limited.
- Time based charging: on and off peak rates.
- Load based charging: price varies with demand.
- Freshness based charging: e.g. share prices and news stories, the newer it is the more expensive it is; note however that very old items may attract an antique premium pricing.

1.4 Possible Enhancements

A useful facility for customers would be to have their own contexts consisting of a a log of the information which they have accessed. This means that customers would not have to start from the same place each time they used the system — the context would remember where they were. The context could be maintained by the customer's machines (e.g. writing a file into their home directory); alternatively some third party could provide this service and charge for it. A structure would need to be imposed on the context to make it manageable (e.g. structure by subject or time of access). If customers are mobile and may use different browsers, then there would need to be some means for customers to find their context.

Customers could be allowed to make printed copies of textual information on their own site. They could be charged for each copy.

Customers could also make electronic copies of the information. Rather than charging for copying, the (much simpler) alternative accepts that you cannot really control copying, but charges on the basis of use. The local copy would not allow itself to be read unless it can log accesses to a valid access log (so that charges can be made). This is the approach advocated in [MORI 90]. One security issue which would need solving is how to prevent customers copying information, editing it and then publishing it as their own.

Note: Is it possible to prevent anybody except the owner editing information?

Some customers will have accounts, so charges will be incurred to these accounts. Other customers may wish to pay by providing access to a source of revenue (e.g. charge or credit card number). Authorisation and billing to the relevant card company would be on-line.

The description in this paper assumes only one organisation publishes information. In a real system many organisations may publish information.

Organisations may compete with each other (e.g. providing a particular movie more cheaply than a rival organisation). Some organisations may make revenue by providing references to information provided by others and charging customers for these references.

If the objective was to design and build and information service, we would pursue the following issues. There are no obvious extra dependability issues. The provision of an information service can be broken down into a number of pieces which could be provided by separate organisations.

- Basic data provision; gathering and making available the raw data.
- Navigation (discovery); linking related pieces of data.
- Presentation (visualisation); formatting (textual or graphical)
- Summarising; producing abstracts or digests
- Billing; keeping the accounts, collecting and distributing the money.

1.5 Scope of the problem

The reason for designing this system is to understand the dependability issues. Hence there are a number of issues which will not be discussed:

- Federating different information models — different organisations may have different formats for their information
- Performance is assumed to be adequate
- Actual payment — customers may want the facility to receive one bill for all the information accessed and have their payment disbursed to the information owners

In addition, although we have tried to provide a very flexible charging model the purpose of this investigation is to explore what it needs to make an application like IPS dependable. It is not an objective to investigate charging models and options for their own sake.

1.6 Services available now

A brief description of some information systems for business news which are available now appear in [SHILLINGFORD 93]:

- Reuter Company Newsyear — a newswire providing real-time news structured according to business categories e.g. telecoms, petro-chemical, pharmaceutical, banking, insurance, automobile, energy, environment, etc.
- London Stock Exchange's Topic database — latest share movements
- Reuter Textline — a database of multi-national corporations
- Datastream — current share prices
- FT-Profile — a database of business information

The charging models for these information systems is very simple: you rent a lease line and pay a subscription. There is no notion of hyperlinks, notification or customising “information packages”. Hence the service management problems are much simpler.

Note: The statement on charging model is based on very limited information. We would appreciate more information.

Other information services are also available on a commercial basis. For example, press agencies such as UPI sell news, and this can be automatically fed into an indexed information service with restricted access. Another very well known information service available in France is Minitel. This provides access to a large number of information services over the telephone network.

Note: Another service which was announced in April 1994 is Uncover [Sunday Times], this includes the facility to fax copies of articles to subscribers. (Awaiting information pack for more details).

WWW is already being used for commercial transactions: it is possible to purchase goods with a credit card. Publishers are being to look at selling encryption keys to give access to online information. In IPS it is hoped to explore some of the options such publishers would have for making their services highly available and reliable, and also making charging dependable and flexible.

2 Event Model for IPS

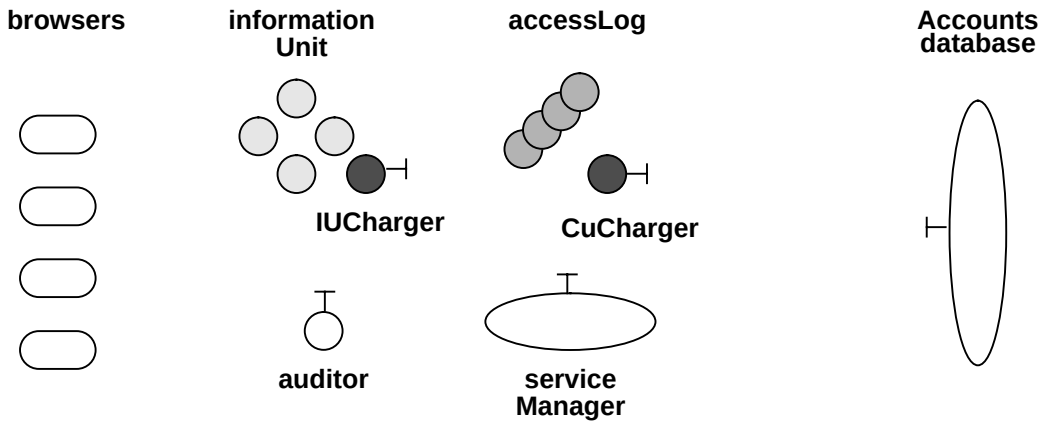
This chapter gives an overview of the principal objects which comprise IPS and shows some of the interactions necessary to enable clients to access information. §2.1 gives an overview of IPS and explains the roles of the objects. §2.2 shows the interactions necessary to enable clients to receive a charging facility. §2.3 shows the interactions which take place when information is accessed and §2.4 shows the interactions which log accesses so that bills can be prepared. A detailed object model for IPS is given in §3.

The auditor and service manger are shown for completeness; their primary function is concerned with managing the systems dependability which will be covered in future papers. This paper concentrates on actions which take place to access information

2.1 An overview of IPS

Figure 2.1 illustrates a simplified IPS system with only one publishing organisation. All client access occurs via a **browser**, this is a display device (maybe a PDA) which is capable of depicting information and which is equipped with some means of accepting payment (e.g. coin or credit card slot, or a keyboard to input credit card details). All client actions are translated into browser actions.

Figure 2.1: A pictorial view of IPS



Information is published in addressable quantities called **informationUnits**, these are typically replicated for availability and load sharing reasons. Each client is represented within the system by an **accessLog**: this object looks after a variety of client generated data, principally a history of past accesses. The **accessLog** is replicated to store volatile information more reliably.

Both **accessLogs** and **IUs** have associated charging policies which between them determine how much an access will cost. The **IUCharger** implements the

publishers charging policy for this **informationUnit**, the **CuCharger** provides details of any discounts relating to this client.

Because browsers may be unreliable it is possible for the system to charge clients for information which wasn't received because of a browser crash. To fix the inconsistency, the system contains **auditor** objects which can modify the stored history. Management in IPS is provided by **serviceManagers** which are responsible for instantiating and configuring **informationUnits, accessLogs** and the chargers, and also for controlling their availability. Finally there is a (legacy) accounts database which stores information about clients in order to maintain their accounts with the publisher.

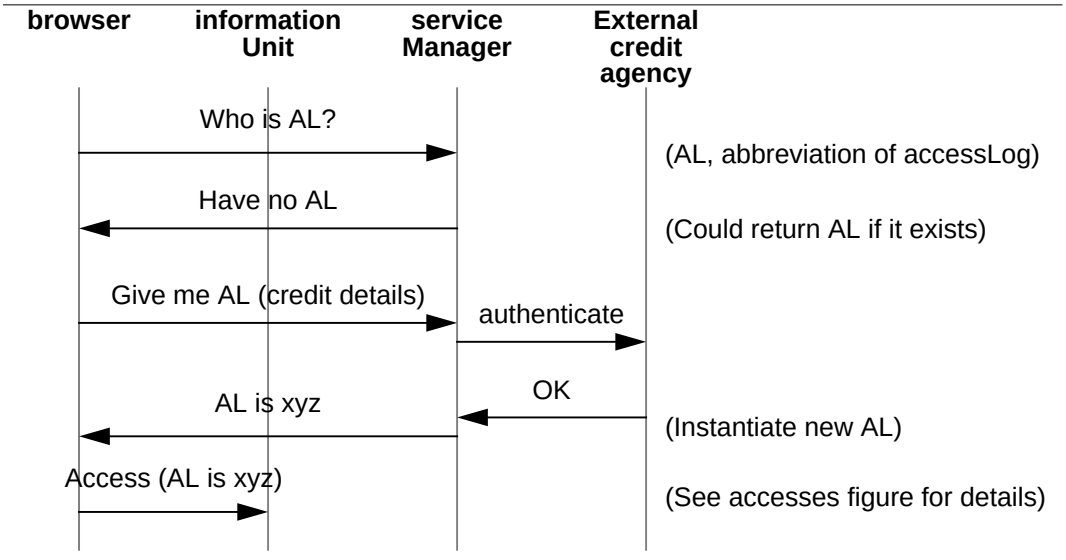
2.2 Acquiring an access log

To access information, a client must generally provide some form of payment. This will usually be some form of credit arrangement, e.g. by providing a credit card to debit or by having an account with the publisher. The role of the **accessLog** is to record accesses made by the user and authorise the **informationUnit** to permit those accesses. Thus with each access the client must identify its **accessLog** in case the access is chargeable.

If the client has used this information provider previously the **accessLog** will already exist and it must be indentified with each (non pre-paid) access. This is done by querying the **serviceManager** as shown in figure 2.2. Note that sufficiently intelligent clients may choose to cache the names of their **accessLogs** to avoid the query overhead.

The first time a client contacts an information publisher it will need to provide sufficient information to enable an **accessLog** to be created. This process is shown in figure 2.2.

Figure 2.2: Explicit accessLog creation

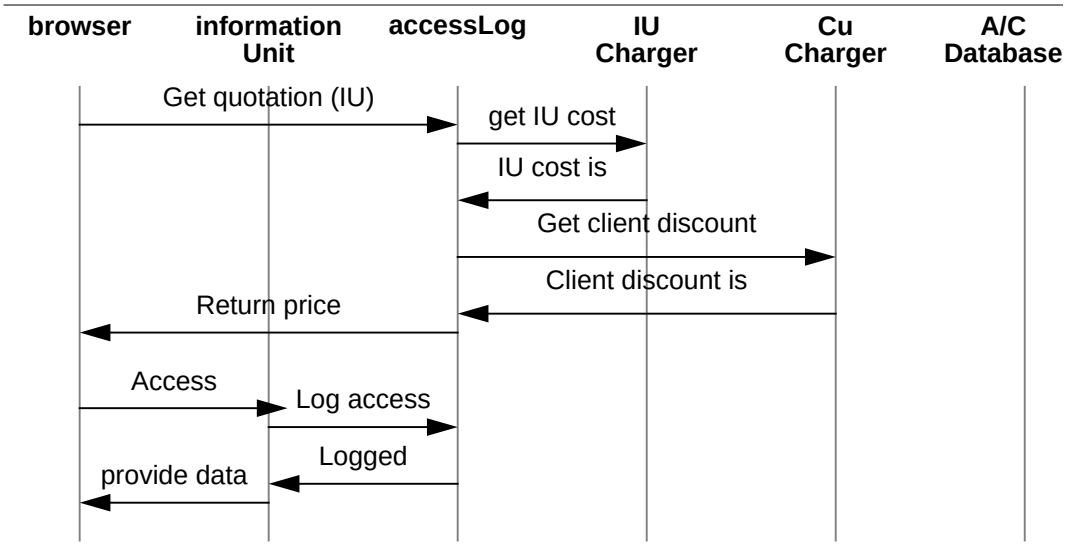


2.3 Accessing information

As noted above, clients must pay for information which they access. Where the cost is unpredictable or likely to be high the client may first wish to establish

what the cost is before deciding whether to make the access. This process is shown in figure 2.3.

Figure 2.3: Accesses with and without quotations



The client first asks the **accessLog** for a quotation (provided in conjunction with the information unit and customer chargers). The resulting price is then returned with the client who may then make the access including the price quoted (useful where prices are rapidly changing). Alternatively the client may make an access including the maximum which it is prepared to pay thus potentially saving latency if the client's credit is good and the cost is acceptable to the publisher.

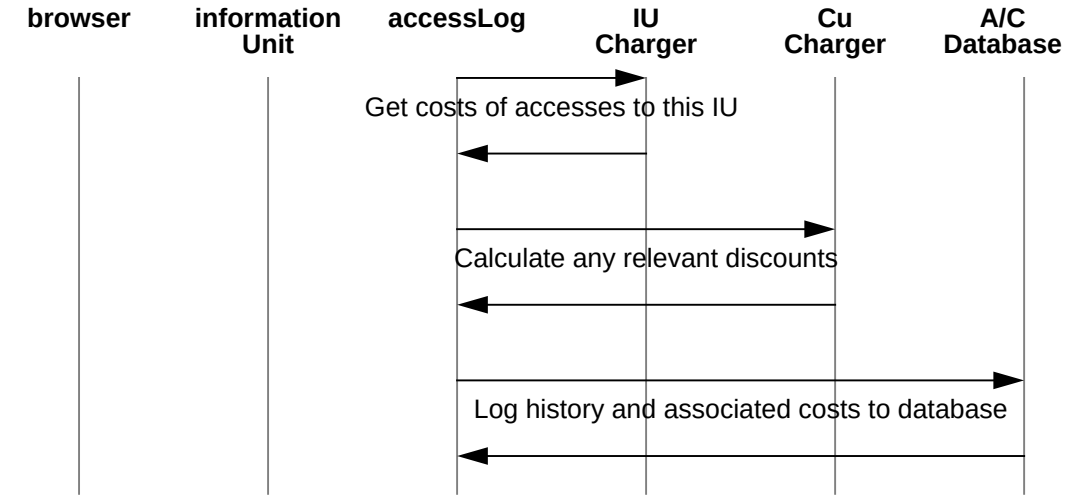
2.4 Charging for past accesses

Periodically, the **accessLog** must transfer the access history to the accounts database as a list of charges. If access was made with an **InformationUnit** generated quotation, then the cost is already known; otherwise the cost must be calculated retrospectively.

Charges can be calculated retrospectively by essentially the same process as obtaining a quotation. The **accessLog** traverses the access history looking for **informationUnits** which have been accessed, but for which no charge has yet been logged. In each case the **IUCharger** is asked for the cost of access then the **CuCharger** is asked for any customer specific discount. Once the **accessLog** knows the cost of access for each **informationUnit** it logs the history and associated costs to the accounts database.

Figure 2.4 shows what happens if the cost of accessing a single **informationUnit** is unknown. In practice the **accessLog** may have to make many invocations of the **CuCharger** and **IUChargers** before it knows the cost of all **informationUnits** accessed and can log this to the database

Figure 2.4: Charging for historic accesses



3 Object model for IPS

This section lists their responsibilities of the objects in IPS (strictly object classes) using the methodology suggested in [SHARBLE 92]. A number of attributes and other associations (not covered by responsibility) are also identified. Taken together this section constitutes the data dictionary described in [RUMBAUGH 91]. The major classes are listed in figure 3.1.

Figure 3.1: Classes

informationUnit	**publisher**	**account**	**organisation**
customer	**IUCharger**	**accessLog**	**notifier**
browser	**CuCharger**	**serviceManager**	**auditor**
context	**CreditAgency**		

3.1 informationUnit

An **informationUnit** is the addressable unit of information in the system. It may contain text, graphics, video or audio. An **informationUnit** may contain references to other **informationUnits**.

The information contained in an **informationUnit** has a value which is assigned by its owner. This is captured in a charger (**IUCharger**) which calculates the charge for accessing a chargeable piece of information in an **informationUnit**. The granularity of a charging is specific to each **informationUnit**, for some **informationUnits**, the granularity may be all the information they contain, for others it may be some portion of the information they contain (e.g. each five minutes of a feature film).

A **browser** can only access an **informationUnit** if the **informationUnit** can log the access with the **browser's accessLog**. If the **browser** presents the **informationUnit** with a quote for the cost of access, the **InformationUnit** is responsible for validating the quote. If the quote is valid, access and charges are logged with the **browser's accessLog**.

Some **informationUnits** will contain an index of what is available. There is therefore no need for any special gateway into the system. **InformationUnits** are owned by an **Organisation**.

Table 3.1: Responsibility of InformationUnit

Responsibility	Collaborator
integrity & availability of information	**serviceManager**
access	**browser**, **accessLog**
validating quotes	

It is assume that an **informationUnit** does not contain a mixture of types, for example both text and images — these would be considered separate **informationUnits**. This avoids the questions of charging for mixed information and the possibility of partial failures of mixed information.

3.2 Customer

Customers have **accounts** or some means of payment. They are not directly responsible for anything. It might be argued that the **customer** is responsible for such things as payment. This is delegated to the **browser**. The purpose of the **customer** class is to show associations with other objects, such as association with **accessLogs** and ownership of **contexts**

Table 3.2: Responsibility of customer

Responsibility	Collaborator

3.3 Browser

The **browser** is the **customer's** interface to the system; for the purposes of this application there is an assumption that a **customer** always accesses IPS using the same **browser**.

It can access **informationUnits** specified by links in **informationUnits** that have already been accessed.

Table 3.3: Responsibility of browser

Responsibility	Collaborator
Updating **customer's context**	**context**
Providing quotes to the **customer**	**accessLog**
Accessing **informationUnits**	**InformationUnit**, **accessLog**, **CuCharger**, **IuCharger**

The **browser** needs to be able to suspend or abort the accessing of an **informationUnit**. It may need to prove the access did or did not take place in the event of a failure. The general principle is that charging needs to relate accurately what actually takes place and needs to be auditable. If the **browser** detects a failure it will invoke the audit service to ensure that the charging information accurately reflects the access.

This document does not discuss how a new **browser** (and its associated **context**, **accessLog** and **CuCharger**) is introduced into the system, or how old **browsers** (and their associated services) are removed. Further work is needed to understand how to do this using the concept of the **serviceManager**; it is important than orphans are not left in the system.

3.4 Context

A **customer's context** is for the convenience of the **customer** (see §1.4). For the purposes of this analysis is considered to reside with or inside the **customer's browser**.

Table 3.4: Responsibility of context

Responsibility	Collaborator
maintaining availability of information	
updating information	**browser**

3.5 Publisher

The **publisher** is responsible for introducing information into the system; it is the tool used by an **organisation** to do this. It is responsible for creating and initialising an **informationUnit**, and attaching it to a **serviceManager.** The **publisher** is also responsible for assigning an **IUCharger** to the **informationUnit**. (It uses the **serviceManager** for doing this).

Introducing a new **informationUnit** means that existing informationUnits need to be updated (at least the indexes) so that they have links to the new **informationUnit**. A **publisher** must also assign an owner to the **informationUnit**.

Publishers can also update (replace) existing information with new information this may mean changing the owner or cost as well as the content. Finally **publishers** can withdraw information. **Publishers** need to preserve the integrity of cross-references between information: changes need to be atomic.

Future papers will explore how to update **informationUnits** and how values and charges can be changed. It is hoped to use the concept of the **serviceManager** and interface group [OSKIEWICZ 93].

Table 3.5: Responsibility of publisher

Responsibility	Collaborator
assigning ownership	**informationUnit**, **organisation**
creating **informationUnits**	**informationUnit**, **ServiceManager**
assigning cost	**informationUnit**, **serviceManager**, **IUCharger**
withdrawing information	**organisation**, **informationUnit**(s), **serviceManager**
updating information	**organisation**, **informationUnit**(s), **serviceManager**
Advertising new information	**informationUnits** (indices)

3.6 IUCharger

The **IUCharger** (**informationUnit** charger) is responsible for calculating the cost of access to an **informationUnit**. The cost of accessing it will reflect the charging

policy, different **informationUnits** will have different policies (see §1.3). In particular the cost may be affected by what the **customer** has already done.

Table 3.6: Responsibility of IUCharger

Responsibility	Collaborator
availability of **IUCharger** service	**serviceManager**
calculating **informationUnit** specific cost of access	**customer's accessLog**

3.7 CuCharger

The **CuCharger** (**customer's** Charger) is responsible for calculating **customer** specific costs for accessing an **informationUnit**, such as **customer** specific discounts. These may be affected by what the **customer** has already done.

Table 3.7: Responsibility of CuCharger

Responsibility	Collaborator
availability of **CuCharger** service	**serviceManager**
calculating **customer** specific cost of access	**customer's accessLog**

3.8 CreditAgency

The **creditAgency** is responsible for authenticating a customer's credit details. It is invoked by the **serviceManager** prior to the creation of an **accessLog** for the **customer**. The **creditAgency** is external to IPS.

Table 3.8: Responsibility of CreditAgency

Responsibility	Collaborator
authenticating **customers** credit details	**serviceManager**

3.9 Account

Account is a (legacy) database responsible for billing the **customers** for payment. **AccessLogs** periodically download their charging information to **account**. Very few assumptions can be made about **account's** availability or reliability. It is assumed it does not corrupt data.

Table 3.9: Responsibility of Account

Responsibility	Collaborator
Billing **customers** for payment	**accessLog**

3.10 AccessLog

Each **customer** is associated with an **accessLog**. In a system with multiple information providers a **customer** would have an **accessLog** with each provider they used.

AccessLogs are responsible for maintaining a log of which **informationUnits** have been accessed. This history may be used by the **CuCharger** and the **IUCharger** to calculate charges.

AccessLogs are also responsible for providing **customer's** with quotes for the cost of accessing an **informationUnit** (obtained by using the **CuCharger** and **IUCharger**).

AccessLogs are responsible for accumulating the charges incurred by a **customer** and downloading this information to **account**. Charges may be known at the time of access (because the **customer** has obtained a valid quote) in which case the charge as well as the access is logged. Alternatively charges are calculated retrospectively by processing the log and using the **CuCharger** and **IUChargers** before handing this information to **account**.

The information held by **AccessLogs** needs to be highly available: **customers** may have been charged a lot of money for information which they are entitled to access without further charges. In addition, it is important that the publisher should not lose information about payments to which it is entitled.

Table 3.10: Responsibility of accessLog

Responsibility	Collaborator
integrity and availability of list of **informationUnits** accessed and charges	**ServiceManager**
logging **informationUnits** accessed	**informationUnit**
logging charges	**informationUnit**
Updating **account**	**Account**
providing quotes	**CuCharger**, **IUCharger**, **browser**
Retrospective calculation of costs	**CuCharger**, **IUCharger**

3.11 ServiceManager

ServiceManagers are responsible for creating, initialising and maintaining the availability of services in IPS. Future documents which discuss how the underlying engineering supports the reliability and availability of IPS will look at the role of the **ServiceManager** in more detail.

Table 3.11: Responsibility of ServiceManager

Responsibility	Collaborator
Availability, reliability creation and initialisation of **InformationUnits**	**InformationUnits**
Availability, reliability creation and initialisation of **accessLogs**	**accessLog**
availability, reliability creation and initialisation of **IUChargers**	**IUChargers**
availability, reliability creation and initialisation of **CuChargers**	**CuChargers**
consistency of links between **InformationUnits**	**InformationUnits**

3.12 Organisation

Organisations own information. The information provided may well have references to sources of information provided by other **organisations**. **Organisations** are ultimately responsible for the integrity and availability of the information which they publish and the services which they provide; **serviceManagers** are used for this. In addition they are responsible for pricing and collecting charges for accessing that information.

In the present version of IPS only a single information provider is considered. The concept of **organisation** will become much more important if it is enhanced to cope with multiple information providers.

Table 3.12: Responsibility of organisation

Responsibility	Collaborator
pricing information	**informationUnit**, **publisher**, **IUCharger**
assigning **customer** specific charges	**accessLog**, **CuCharger**
publishing information	**publisher**, **ServiceManager**
withdrawing information	**publisher**, **ServiceManager**
updating information	**publisher**, **ServiceManager**
billing and collecting charges	**accessLog**, **accounts**
Availability, reliability creation and initialisation of **InformationUnits, accessLogs, IUChargers** and **CuChargers**	**ServiceManager**

3.13 Notifier

The **notifier** is responsible for notifying **customers** who have registered an interest in an **informationUnit** of significant events in the lifecycle of that **informationUnit**. For example, the **informationUnit** has been updated or withdrawn. Charges are registered for its service: when **customers** receives a notification they will be charged. The **notifier** is responsible for maintaining the consistency and integrity of the database of registered interests, and ensuring that the notifications are reliably delivered. The notification service will be explored more fully in forthcoming documents.

Table 3.13: Responsibility of notifier

Responsibility	Collaborator
consistency & integrity of registration data	
registering interests	**accessLog**
notifying significant lifecycle events	**informationUnit**, **publisher**, **accessLog**

3.14 Auditor

The **auditor** is responsible for ensuring that the charges or logged accesses reflect accurately what has actually taken place. Its role will be explored more

fully in forthcoming documents which consider how the underlying engineering provides a suitable level of dependability for IPS.

Table 3.14: Responsibility of auditor

Responsibility	Collaborator
consistency and accuracy of logged accesses and charges.	**InformationUnits**, **browsers**, **accessLog**

3.15 The Object Model

This section presents the object model of the information system using the notation of [RUMBAUGH 91] (see figures 3.2 and 3.3).

3.16 Acknowledgements

The authors are grateful for discussions with Owen Rees of APM Ltd., and Dave Raggett and Paul Vickers of Hewlett-Packard.

Figure 3.2: Object model: Browser/InformationUnit

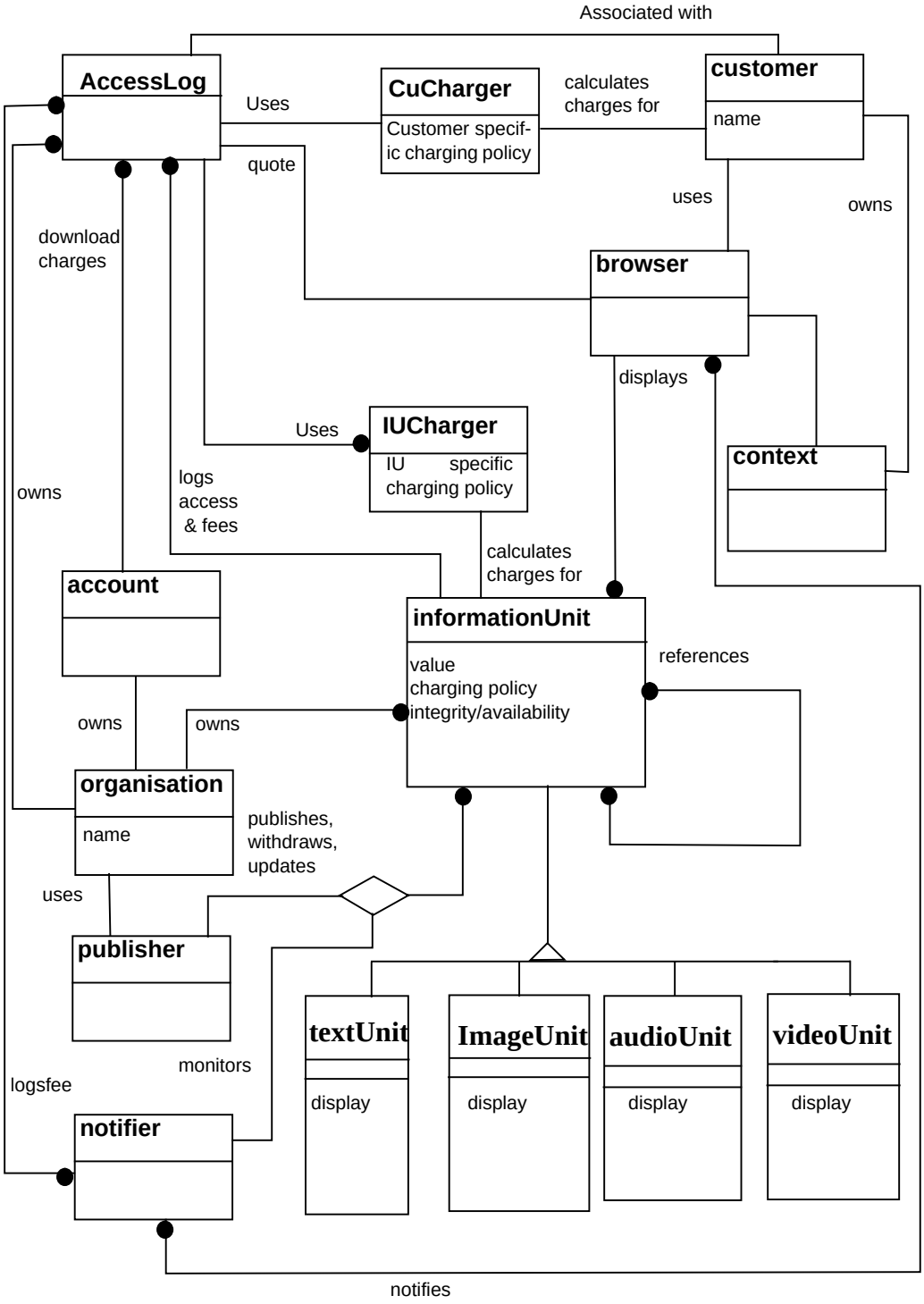
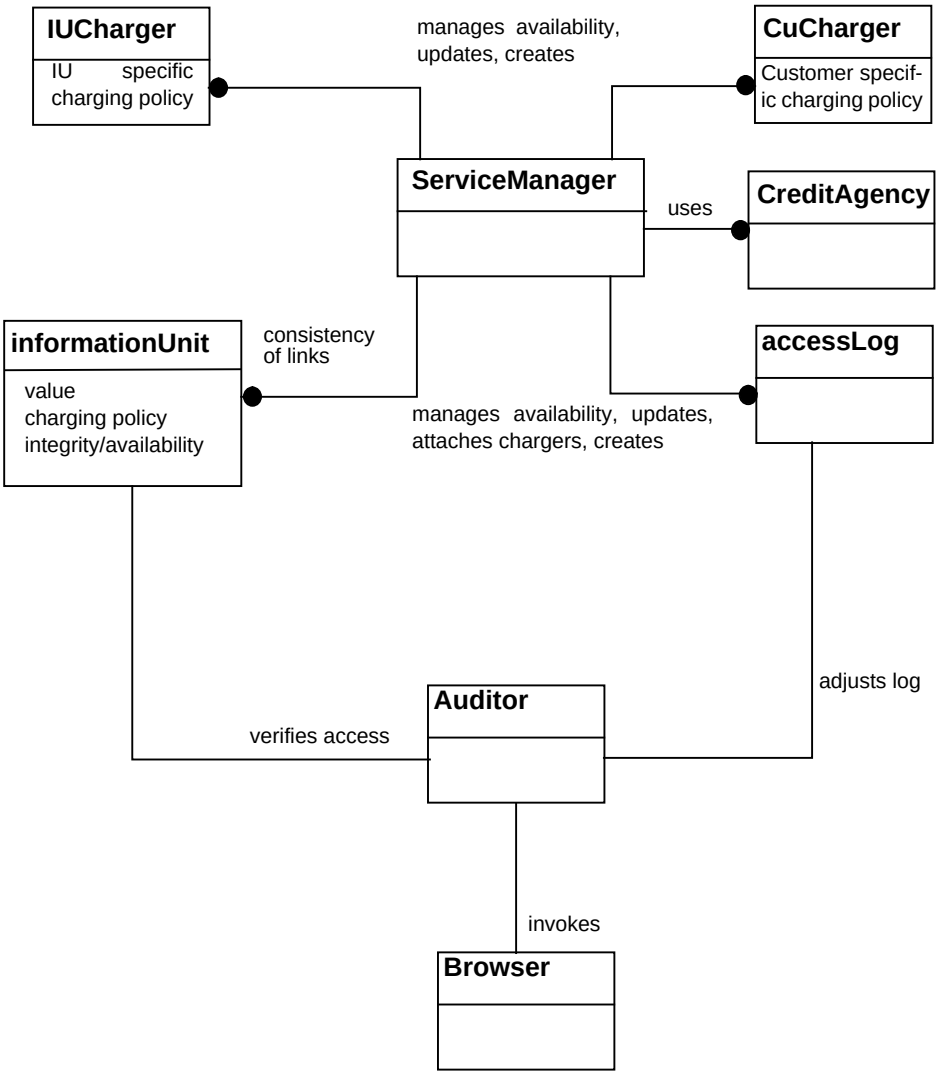


Figure 3.3: Object Model: ServiceManager & Auditor



References

[EDWARDS 94a]

Edwards, N.J., Oskiewicz, E.P., Rees, R.T.O., “A dependability scenario — IPS (Information Publishing System)., APM1096, APM Ltd., Cambridge U.K., 1994, January 1994.

[EDWARDS 94b]

Edwards, N.J., “Expectation analysis for IPS (the dependability scenario)”, APM1119, APM Ltd., Cambridge U.K., 1994, January 1994.

[EDWARDS 93]

Edwards, N.J., Rees, R.T.O., “A Model for Failure in Dependable Systems” APM.1027, November 1993.

[HEILEMANN 94]

John Heilemann, “A Survey of Television”, The Economist, February 12th, 1994.

[MORI 90]

Mori, R., Kawahara, M., “Superdistribution: The Concept and the Architecture”, Transactions of the IEICE; Vol. E-73#7 July 1990; Special Issue on Cryptography and Information Security.

[OSKIEWICZ 93]

Oskiewicz, E.P., Edwards, N.J., “A Model for Interface Groups”, AR002.001, APM Ltd., Cambridge, 1993.

[RUMBAUGH 91]

Rumbaugh J., Blaha, M., Premerlani, W., Eddy, F., Lorenson, W., “Object-Oriented Modeling and Design”, Prentice-Hall International 1991.

[SHARBLE 92]

Sharble R.C., Cohen, S.S.,“The Object-Oriented Brewery: A Comparison of Two Object-Oriented Development Methods”, Document BCS-G4059, Boeing, October 19th 1992.

[SHILLINGFORD 93]

Shillingford, J., “Online Databases — Windows adds appeal”, Financial Times, November 8th 1993.