Software licenses

Closed-source, open-source, free

Most end-user software is closed-source (proprietary).

- The source code is either
  - never revealed (most commonly), or
  - only made available to select customers (rarely).

- Customers either:
  - get the executable only, or
  - get obfuscated code (e.g. interpreted languages), or
  - access the service over the network only (e.g. “cloud” services).

Example of closed-source software

Only the executable is distributed:

- Operating systems: Microsoft Windows; Android, iOS, MacOS (except kernel)
- Office suites: Microsoft Office, iWork
- Creative software: Adobe suite, Autodesk suite, Final Cut Pro, Pro Tools, Logic Pro
- Development software: Visual Studio; XCode (except compiler)
- Server-side / enterprise: Microsoft IIS, Oracle Database
- Almost all videogames
- Almost all mobile apps

Example of closed-source software

Over the network / obfuscated code:

- Office suites: Microsoft 365, Google Workspace
- Collaboration software: Zoom, Teams, Skype, Slack, Discord
- Enterprise software: SAP
- Generative AI: OpenAI ChatGPT / DALL-E / Sora, Midjourney, Anthropic Claude, Google Gemini

Free software

- “free” as in freedom (not “free lunch”)
- defined by the Free Software Foundation (FSF, est. 1985)
- software attached with a license (uses copyright law)
- gives freedoms (rights) to the user, to:
  - run the software as they wish
  - study and modify the software as they wish
  - redistribute (original and modified versions)
- based on the philosophy that all software should be free, to protect its users
- the “GNU” project is FSF’s software collection

Open-source software

- defined by the Open Source Initiative (OSI, est. 1998)
- software attached with a license (uses copyright law)
- specifies how software can be distributed:
  - no restrictions on redistribution
    - no discrimination against specific users, fields, products, other software, other technologies
  - source code must be available
  - derived works must be allowed
  - but modifications can be required to be clearly delineated

FSF vs. OSI

- FSF: for the user's sake, all software should be free on ethical grounds
  – free software licenses are a means to that end

- OSI: help businesses and developers publish and disseminate their open-source software
  – pragmatically, we do not want to add hurdles if they impair practical use

In practice?

The FSF and OSI each maintain a list of “approved” license.

Most FSF-approved free software licenses are also OSI-approved open-source licenses. And vice-versa.

The difference lies in the licenses each organization **promotes**

The “baseline” FSF license

The GNU General Public License (**GPL**):

- any user who receives the executable must be provided the source code as well upon request
- any derivative work is automatically covered by the GPL (the GPL is “viral”)
- dynamic linking with GPL software counts as derivative work

Amended FSF licenses

- The “more permissive” GNU Lesser General Public License (**LGPL**):
  - adds exception to allow dynamic linking with non-GPL software

- The “more restrictive” GNU Affero General Public License (**AGPL**)
  - definition of “user” includes over-the-network interactions

Typical open-source licenses

- Most popular: Apache License, BSD License, MIT License
- “permissive licenses”: fewer constraints on derivative work
  - unmodified parts still covered by the original license
  - but modified parts are not, can even be closed source
- some require acknowledgement of the original work (authors and/or project)
- differences among permissive licenses are minor (but important to lawyers)

Example projects

- “free software” (GPL-type licenses)
  - Linux kernel, **GPL**
  - GNU project, **GPL**
  - gcc, **GPL**
  - glibc (gcc’s standard C library), **LGPL**
  - git, **GPL**
- “open source” (permissive licenses)
  - Apache web server, **Apache**
  - NGINX web server, **BSD**
  - LLVM, **Apache**
  - Chrome (more precisely: chromium), **BSD**
  - Node.js, Angular, React, **MIT**

Source-available software

"**Source-available software**":
Source code is available, but **not** under a license approved by either FSF or OSI.

Examples:

- Unreal Engine
- Redis
- Legacy software:
  - Apple DOS,
  - Word 1
  - Photoshop 1
  - Sun Java

End-user cost

Whether or not customers pay for software is **orthogonal** to source availability.

cost	free / open-source	closed-source
0	Chrome, Gimp, VLC, Blender	TikTok, Whatsapp, Discord
> 0	Red Hat Enterprise Linux	Photoshop, Maya, Ableton

Commercial, non-commercial

Whether or not developers are commercial entities is **orthogonal** to source availability.

developers	free / open-source	closed-source
non-commercial	FSF's GNU system Blender, Krita, LibreOffice	most amateur code until 2010s some government software legacy scientific software
commercial	Chrome, Ubuntu, Red Hat, NGINX, Docker, GitLab, LLVM	Microsoft Windows, Microsoft 365 iWork, Adobe suite, Autodesk, …

*the distinction between commercial and non-commercial is often blurry

How can commercial software be free / open-source?

- software has zero price, sell support and services (Ubuntu, Red Hat, NGINX)
- software costs money, convince customers not to redistribute it (Red Hat)
- open-core: basic functionality is open-source, sell advanced features (NGINX)
- open-sourced software accesses proprietary services (Chrome)
- open-sourced software is not core business (LLVM)

Use cases

- Closed-source:
  - source code is your “secret sauce”
  - customers willing to pay
- Source-available:
  - make distribution and debugging easier
  - force customers to pay (fixed price or share of profits)
- Open-source permissive licenses:
  - encourage wider adoption
  - encourage commercial entities to participate
- Free software GPL-type licenses:
  - protect users (ethical grounds)
  - force downstream developers to reciprocate

Patents

In most countries:

- Contrary to copyright law (protects creative processes) patents are not a fundamental right
- Patents are a pragmatic compromise for promoting innovation.

  The bargain is:

  - Share your innovation with the patent office (as opposed to keeping it secret)
  - Get N-year exclusivity on commercialization

Patents and software innovation

- Software innovation is quick: 20 years are an eternity
- Ideas are cheap, execution is everything
- Software is close to mathematics (discovered, not invented)
- Patent disclosures do not include code! They don’t actually help anyone.

Stuff that has been patented

- Buy with a single click (Amazon, 1999)
- Automatically make email addresses and phone numbers clickable (Apple, 1995)
- Fourier (1768–1830) series for compression (Fraunhofer Institute, 1991)

US005960411A

United States Patent [19]

Hartman et al.

[11] **Patent Number: 5,960,411**

[45] **Date of Patent: Sep. 28, 1999**

[54] **METHOD AND SYSTEM FOR PLACING A PURCHASE ORDER VIA A COMMUNICATIONS NETWORK**

[75] Inventors: **Peri Hartman**; **Jeffrey P. Bezos**; **Shel Kaphan**; **Joel Spiegel**, all of Seattle, Wash.

[73] Assignee: **Amazon.com, Inc.**, Seattle, Wash.

[21] Appl. No.: **08/928,951**

[22] Filed: **Sep. 12, 1997**

[51] **Int. Cl.**[6] **G06F 17/60**

[52] **U.S. Cl.** **705/26**; 705/27; 345/962

[58] **Field of Search** 705/26, 27; 380/24, 380/25; 235/2, 375, 378, 381; 395/188.01; 345/962

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,937,863	6/1990	Robert et al.	380/4
5,204,897	4/1993	Wyman	380/4
5,260,999	11/1993	Wyman	384/4
5,627,940	5/1997	Rohra et al.	395/12
5,640,501	6/1997	Turpin	395/768
5,640,577	6/1997	Scharmer	395/768
5,664,111	9/1997	Nahan et al.	705/27
5,715,314	2/1998	Payne et al.	380/24
5,715,399	2/1998	Bezos	705/27
5,727,163	3/1998	Bezos	705/27
5,745,681	4/1998	Levine et al.	395/200.3
5,758,126	5/1998	Daniels et al.	395/500

FOREIGN PATENT DOCUMENTS

0855659 A1	1/1998	European Pat. Off.	G06F 17/30
0855687 A2	1/1998	European Pat. Off.	G07F 19/00
0845747A2	6/1998	European Pat. Off.	G06F 17/60
0883076A2	12/1998	European Pat. Off.	G06F 17/60
WO 95/30961	11/1995	WIPO	G06F 17/60
WO 96/38799	12/1996	WIPO	G06F 17/60
WO 98/21679	5/1998	WIPO	G06F 17/60

OTHER PUBLICATIONS

Jones, Chris. "Java Shopping Cart and Java Wallet; Oracles plans to join e–commerce initiative." Mar. 31, 1997, InfoWorld Media Group.

"Pacific Coast Software Software creates virtual shopping cart." Sep. 6, 1996. M2 Communications Ltd 1996.

"Software Creates Virtual Shopping Cart." Sep. 5, 1996. Business Wire, Inc.

Terdoslavich, William. "Java Electronic Commerce Framework." Computer Reseller News, Sep. 23, 1996, CMP Media, Inc., 1996, pp. 126, http://www.elibrary.com/id/101/101/getdoc . . . rydocid=902269@library_d&dtype=0–0&dinst=. [Accessed Nov. 19, 1998].

"Internet Access: Disc Distributing Announces Interactive World Wide." Cambridge Work–Group Computing Report, Cambridge Publishing, Inc., 1995, http://www.elibrary.com/id/101/101/getdoc . . . docid=1007497@library_a&dtype=0–0&dinst=0. [Accessed Nov. 19, 1998].

(List continued on next page.)

Primary Examiner—James P. Trammell
Assistant Examiner—Demetra R. Smith
Attorney, Agent, or Firm—Perkins Coie LLP

[57] **ABSTRACT**

A method and system for placing an order to purchase an item via the Internet. The order is placed by a purchaser at a client system and received by a server system. The server system receives purchaser information including identification of the purchaser, payment information, and shipment information from the client system. The server system then assigns a client identifier to the client system and associates the assigned client identifier with the received purchaser information. The server system sends to the client system the assigned client identifier and an HTML document identifying the item and including an order button. The client system receives and stores the assigned client identifier and receives and displays the HTML document. In response to the selection of the order button, the client system sends to the server system a request to purchase the identified item. The server system receives the request and combines the purchaser information associated with the client identifier of the client system to generate an order to purchase the item in accordance with the billing and shipment information whereby the purchaser effects the ordering of the product by selection of the order button.

26 Claims, 11 Drawing Sheets

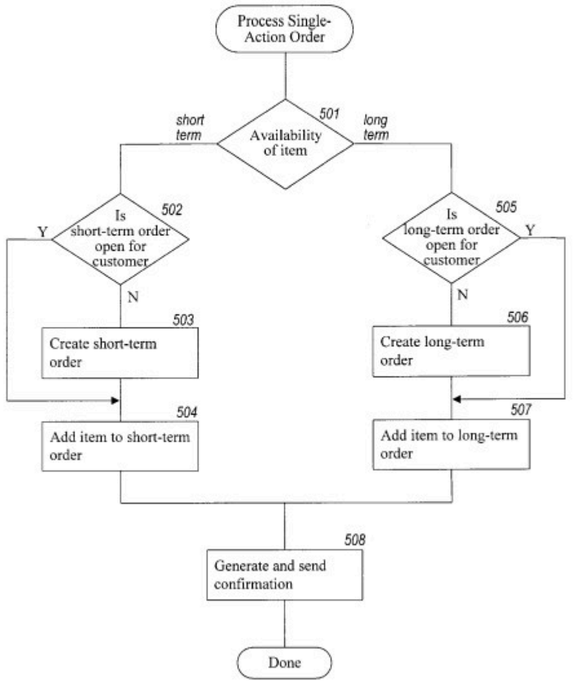

Fig. 5

METHOD AND SYSTEM FOR PLACING A PURCHASE ORDER VIA A COMMUNICATIONS NETWORK

TECHNICAL FIELD

The present invention relates to a computer method and system for placing an order and, more particularly, to a method and system for ordering items over the Internet.

BACKGROUND OF THE INVENTION

The Internet comprises a vast number of computers and computer networks that are interconnected through communication links. The interconnected computers exchange information using various services, such as electronic mail, Gopher, and the World Wide Web ("WWW"). The WWW service allows a server computer system (i.e., Web server or Web site) to send graphical Web pages of information to a remote client computer system. The remote client computer system can then display the Web pages. Each resource (e.g., computer or Web page) of the WWW is uniquely identifiable by a Uniform Resource Locator ("URL"). To view a specific Web page, a client computer system specifies the URL for that Web page in a request (e.g., a HyperText Transfer Protocol ("HTTP") request). The request is forwarded to the Web server that supports that Web page. When that Web server receives the request, it sends that Web page to the client computer system. When the client computer system receives that Web page, it typically displays the Web page using a browser. A browser is a special-purpose application program that effects the requesting of Web pages and the displaying of Web pages.

Currently, Web pages are typically defined using HyperText Markup Language ("HTML"). HTML provides a standard set of tags that define how a Web page is to be displayed. When a user indicates to the browser to display a Web page, the browser sends a request to the server computer system to transfer to the client computer system an HTML document that defines the Web page. When the requested HTML document is received by the client computer system, the browser displays the Web page as defined by the HTML document. The HTML document contains various tags that control the displaying of text, graphics, controls, and other features. The HTML document may contain URLs of other Web pages available on that server computer system or other server computer systems.

The World Wide Web is especially conducive to conducting electronic commerce. Many Web servers have been developed through which vendors can advertise and sell product. The products can include items (e.g., music) that are delivered electronically to the purchaser over the Internet and items (e.g., books) that are delivered through conventional distribution channels (e.g., a common carrier). A server computer system may provide an electronic version of a catalog that lists the items that are available. A user, who is a potential purchaser, may browse through the catalog using a browser and select various items that are to be purchased. When the user has completed selecting the items to be purchased, the server computer system then prompts the user for information to complete the ordering of the items. This purchaser-specific order information may include the purchaser's name, the purchaser's credit card number, and a shipping address for the order. The server computer system then typically confirms the order by sending a confirming Web page to the client computer system and schedules shipment of the items.

Since the purchaser-specific order information contains sensitive information (e.g., a credit card number), both

vendors and purchasers want to ensure the security of such information. Security is a concern because information transmitted over the Internet may pass through various intermediate computer systems on its way to its final destination. The information could be intercepted by an unscrupulous person at an intermediate system. To help ensure the security of the sensitive information, various encryption techniques are used when transmitting such information between a client computer system and a server computer system. Even though such encrypted information can be intercepted, because the information is encrypted, it is generally useless to the interceptor. Nevertheless, there is always a possibility that such sensitive information may be successfully decrypted by the interceptor. Therefore, it would be desirable to minimize the sensitive information transmitted when placing an order.

The selection of the various items from the electronic catalogs is generally based on the "shopping cart" model. When the purchaser selects an item from the electronic catalog, the server computer system metaphorically adds that item to a shopping cart. When the purchaser is done selecting items, then all the items in the shopping cart are "checked out" (i.e., ordered) when the purchaser provides billing and shipment information. In some models, when a purchaser selects any one item, then that item is "checked out" by automatically prompting the user for the billing and shipment information. Although the shopping cart model is very flexible and intuitive, it has a downside in that it requires many interactions by the purchaser. For example, the purchaser selects the various items from the electronic catalog, and then indicates that the selection is complete. The purchaser is then presented with an order Web page that prompts the purchaser for the purchaser-specific order information to complete the order. That Web page may be prefilled with information that was provided by the purchaser when placing another order. The information is then validated by the server computer system, and the order is completed. Such an ordering model can be problematic for a couple of reasons. If a purchaser is ordering only one item, then the overhead of confirming the various steps of the ordering process and waiting for, viewing, and updating the purchaser-specific order information can be much more than the overhead of selecting the item itself. This overhead makes the purchase of a single item cumbersome. Also, with such an ordering model, each time an order is placed sensitive information is transmitted over the Internet. Each time the sensitive information is transmitted over the Internet, it is susceptible to being intercepted and decrypted.

SUMMARY OF THE INVENTION

An embodiment of the present invention provides a method and system for ordering an item from a client system. The client system is provided with an identifier that identifies a customer. The client system displays information that identifies the item and displays an indication of an action (e.g., a single action such as clicking a mouse button) that a purchaser is to perform to order the identified item. In response to the indicated action being performed, the client system sends to a server system the provided identifier and a request to order the identified item. The server system uses the identifier to identify additional information needed to generate an order for the item and then generates the order.

The server system receives and stores the additional information for customers using various computer systems so that the server system can generate such orders. The server system stores the received additional information in association with an identifier of the customer and provides

… 45 pages …

Neural networks

Weights vs. source code

- Neural network weights are not source code
- Not yet clear how copyright law applies
- Most LLM weights are kept secret
  - OpenAI have not published weights since GPT2 (2019)

Llama

- Llama 3 weights available under the “Meta Llama 3 community license agreement”
  - max. 700 million monthly active users
  - cannot use output to train another LLM
- Llama 3 inference code is open source (MIT license)
- Llama 3 training code is closed source
- Llama 3 training data is kept secret

Stable diffusion

Stable Diffusion weights available under “CreativeML Open RAIL-M” license:

[…]
You agree not to use the Model or Derivatives of the Model:

- In any way that violates any applicable national, federal, state, local or international law or regulation;
- For the purpose of exploiting, harming or attempting to exploit or harm minors in any way;
- To generate or disseminate verifiably false information and/or content with the purpose of harming others;
- To generate or disseminate personal identifiable information that can be used to harm an individual;
- To defame, disparage or otherwise harass others;
- For fully automated decision making that adversely impacts an individual’s legal rights or otherwise creates or modifies a binding, enforceable obligation;
- For any use intended to or which has the effect of discriminating against or harming individuals or groups based on online or offline social behavior or known or predicted personal or personality characteristics;
- To exploit any of the vulnerabilities of a specific group of persons based on their age, social, physical or mental characteristics, in order to materially distort the behavior of a person pertaining to that group in a manner that causes or is likely to cause that person or another person physical or psychological harm;
- For any use intended to or which has the effect of discriminating against individuals or groups based on legally protected characteristics or categories;
- To provide medical advice and medical results interpretation;
- To generate or disseminate information for the purpose to be used for administration of justice, law enforcement, immigration or asylum processes, such as predicting an individual will commit fraud/crime commitment (e.g. by text profiling, drawing causal relationships between assertions made in documents, indiscriminate and arbitrarily-targeted use).