US006044466A

United States Patent [19]

Anand et al.

[11] Patent Number: 6,044,466

[45] Date of Patent: Mar. 28, 2000

[54] **FLEXIBLE AND DYNAMIC DERIVATION OF PERMISSIONS**

[75] Inventors: **Rangachari Anand**, Teaneck, N.J.; **Frederique-Anne Giraud**, Ossining, N.Y.; **Nayeem Islam**, Thornwood, N.Y.; **Trent Ray Jaeger**, Croton-on Hudson, N.Y.; **Jochen Liedtke**, Ossining, N.Y.

[73] Assignee: **International Business Machines Corp.**, Armonk, N.Y.

[21] Appl. No.: **08/979,807**

[22] Filed: **Nov. 25, 1997**

[51] **Int. Cl.^{7}** **G06F 12/14**
[52] **U.S. Cl.** **713/200**
[58] **Field of Search** 713/200, 201; 709/225, 229; 364/222.5, 286.4, 286.5; 711/163; 707/9

[56] **References Cited**

U.S. PATENT DOCUMENTS

5,265,221	11/1993	Miller	395/725
5,335,346	8/1994	Fabbio	395/600
5,649,099	7/1997	Theimer et al.	395/187.01
5,911,143	6/1999	Deinhart et al.	707/103
5,924,094	7/1999	Sutter	707/10
5,925,126	7/1999	Hsieh	713/200
5,966,715	10/1999	Sweeney et al.	707/203
5,968,177	10/1999	Batten-Carew et al.	713/201

OTHER PUBLICATIONS

Yellin, "Low Level Security in Java", http://java.sun.com/sfaq/verifier.html, pp 1–12, Dec. 1996.

Bank, "Java Security", http://www–swiss.ai.mit.edu/~jbank/javapaper.html, pp 1–11, Dec. 1995.

Thomas, "The Navigator Java Environment: Current Security Issues", http://developer.netscape.com/docs/manuals/javasecurity.html, pp. 1–3, Jan. 1996.
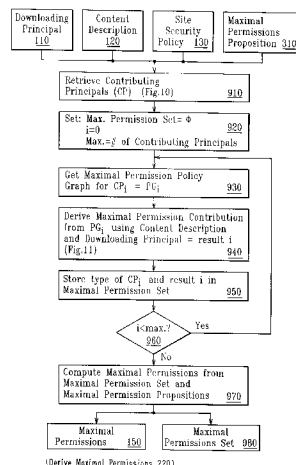
Jaeger et al., "Implementation of a Discretionary Access Control Model for Scrip–based Systems", 8th IEEE Computer Security Foundations Workshop Proceedings, pp 70–84, Jun. 1995.

Sandhu et al., "Role–Based Access Control: A Multi–Dimensional View", IEEE 10th Annual Computer Security Applications Conference Proceedings, pp 54–62, Dec. 1994.

Anand et al., "A Flexible Security Model for Using Internet Content", IEEE The Sixteenth Symposium on Reliable Distributed Systems Proceedings, pp 89–96, Oct. 1997.

Unknown, "HotJava(tm): The Security Story", http://java-.sun.com/sfaq/may95/security.html, pp. 1–7, May 1995.

Trent Jaeger, Aviel D. Rubin, Atul Prakash, "Building Systems That Flexibly Control Downloaded Executable Content", USENIX Association, 6th USENIX Security Symposium, pp. 131–148.

Nayeem Islam, Rangachari Anand, Trent Jaeger and Josyula R. Rao, "A Flexible Security System For Using Internet Content", IEEE Software, Sep./Oct. 1997, pp. 52–59.

Primary Examiner—Robert W. Beausoliel, Jr.
Assistant Examiner—Stephen C. Elmore
Attorney, Agent, or Firm—Scully, Scott, Murphy & Presser; Kevin N. Jordan, Esq.

[57] **ABSTRACT**

A dynamic derivation mechanism is defined which enables limited permissions to be dynamically and flexibly derived for executables based upon their authenticated description. The dynamic derivation mechanism uses the authenticated description to determine the maximal permissions that individual principals can delegate to the content. A principal's maximal permissions for content define a superset of the rights that that principal will actually delegate to that content. Although the maximal permissions are derived from predefined specifications, the specifications can be sensitive to runtime state on the downloader's system or previous delegations to enable the dynamic (i.e., runtime) derivation. Multiple principals can delegate a subset of their maximal permissions for the executable content. The mechanism uses policy for combining the delegated permissions into the content's runtime permissions.

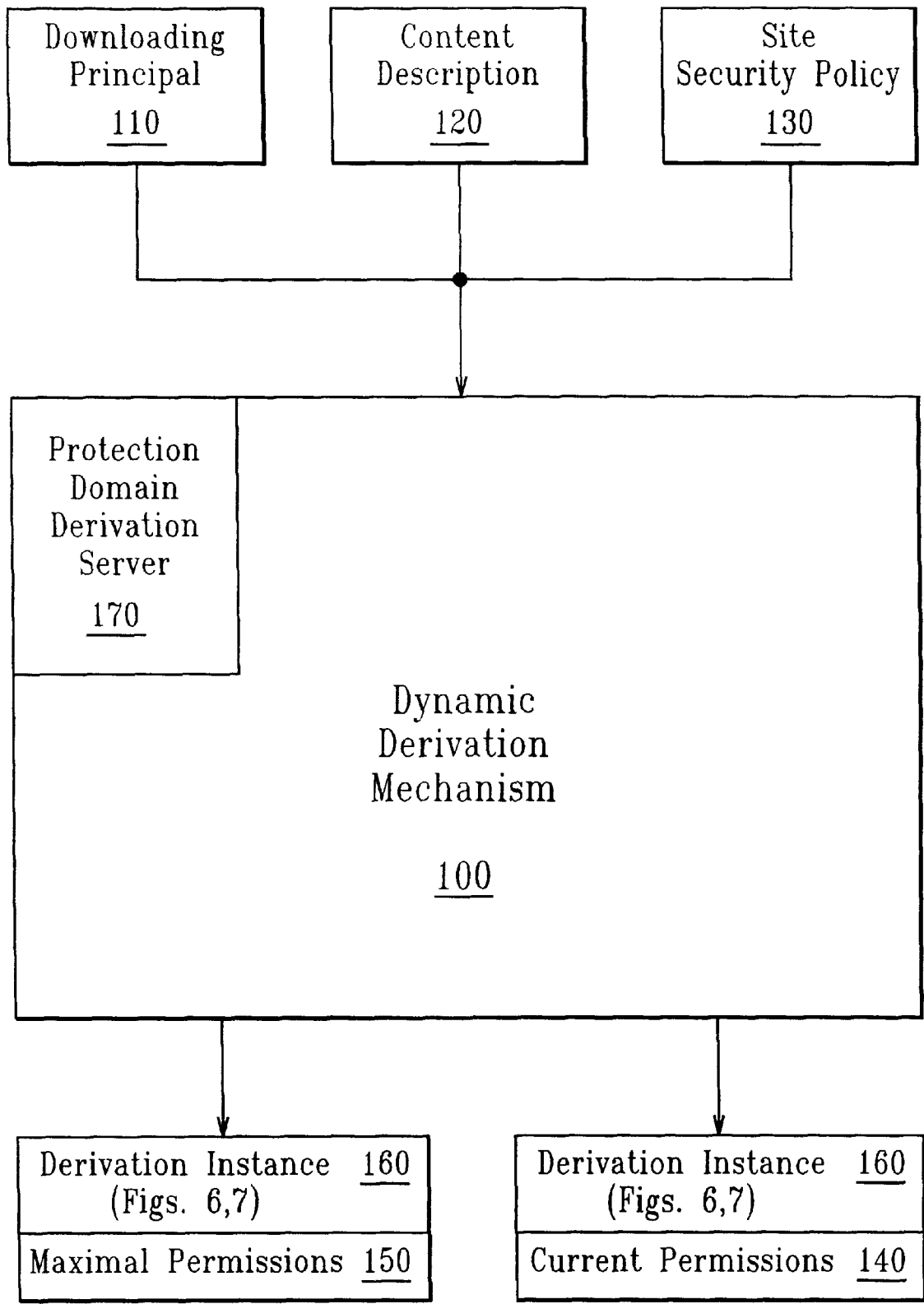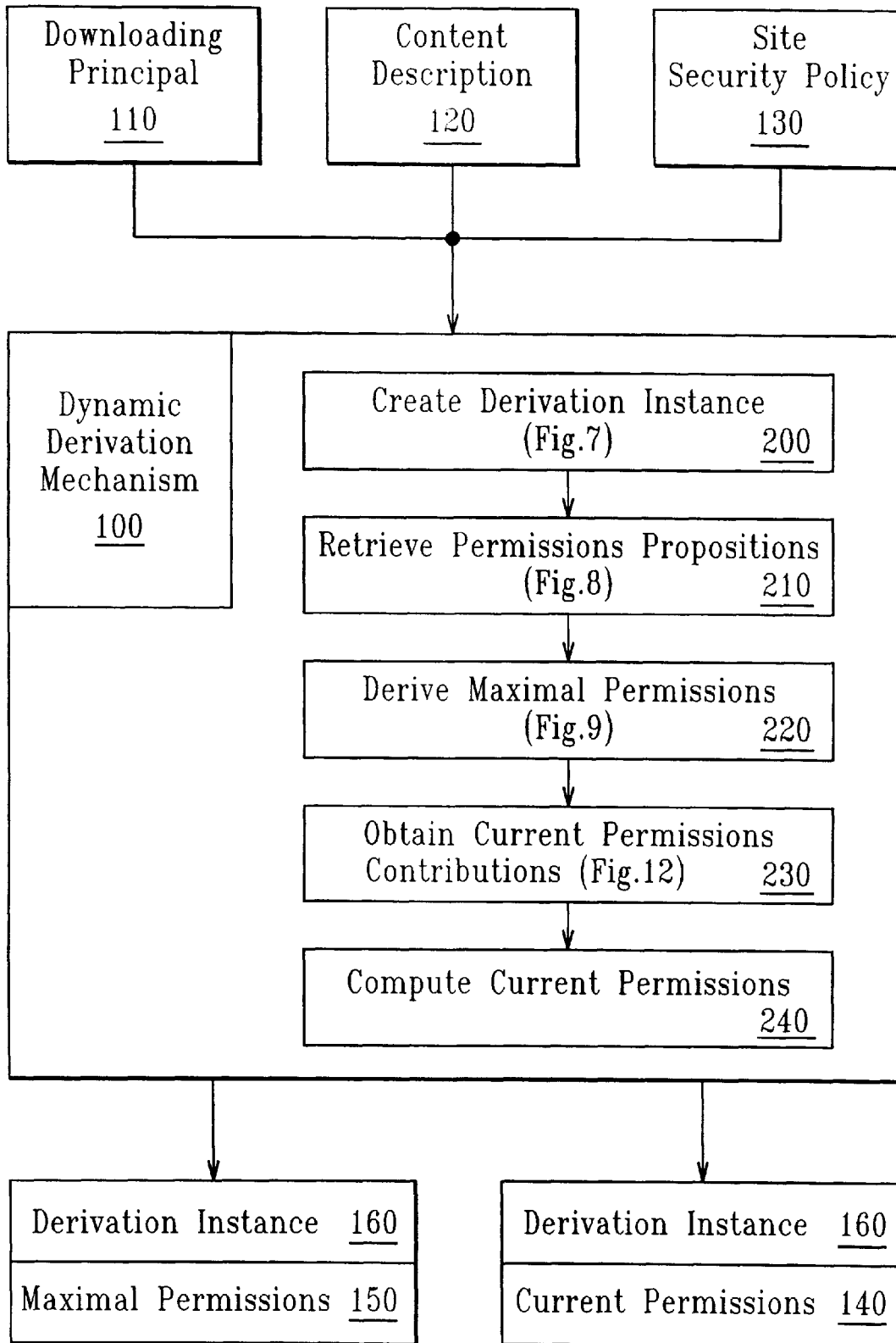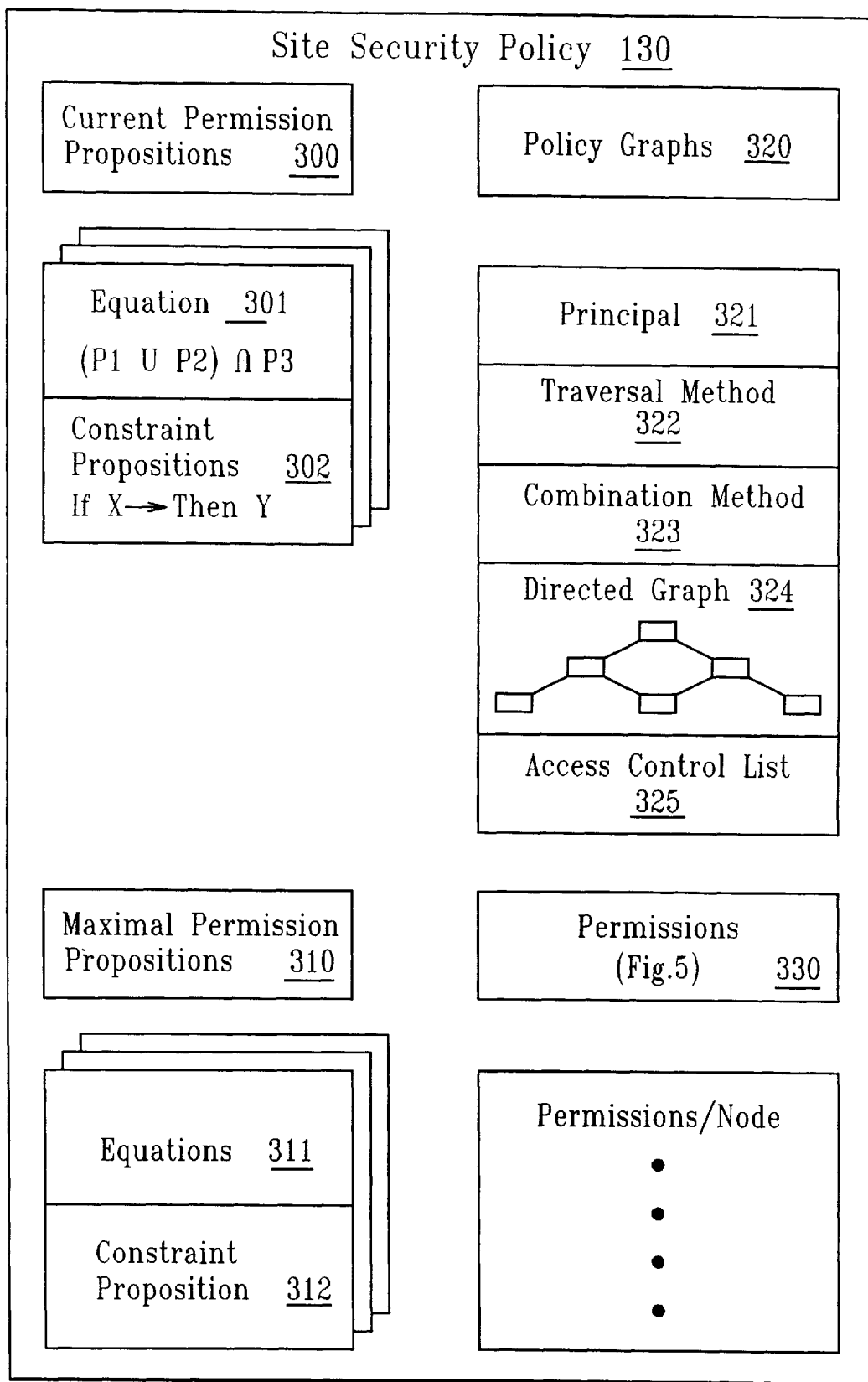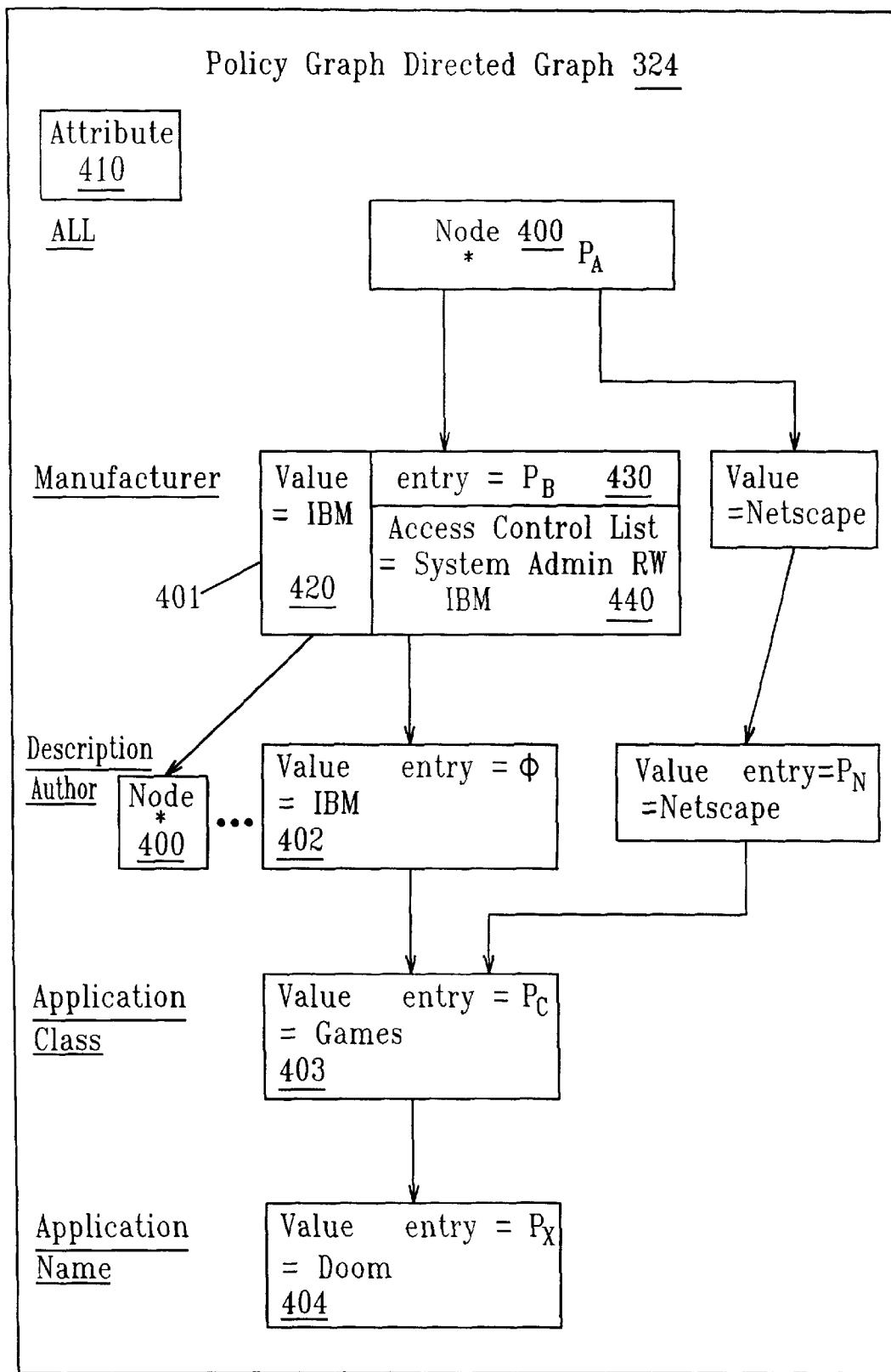**44 Claims, 11 Drawing Sheets**

FIG.1

```
┌──────────────────┐  ┌──────────────────┐  ┌──────────────────┐
│   Downloading    │  │     Content      │  │       Site       │
│    Principal     │  │   Description    │  │  Security Policy  │
│       110        │  │       120        │  │       130        │
└──────────────────┘  └──────────────────┘  └──────────────────┘
```

```
┌─────────────────────────────────────────────────────────────┐
│              ┌───────────────────────────────────────────┐   │
│ Dynamic      │   Create  Derivation  Instance            │   │
│ Derivation   │           (Fig.7)              200         │   │
│ Mechanism    └───────────────────────────────────────────┘   │
│   100        ┌───────────────────────────────────────────┐   │
│              │  Retrieve  Permissions  Propositions       │   │
│              │            (Fig.8)             210          │   │
│              └───────────────────────────────────────────┘   │
│              ┌───────────────────────────────────────────┐   │
│              │   Derive  Maximal  Permissions             │   │
│              │            (Fig.9)             220          │   │
│              └───────────────────────────────────────────┘   │
│              ┌───────────────────────────────────────────┐   │
│              │  Obtain  Current  Permissions              │   │
│              │  Contributions (Fig.12)        230          │   │
│              └───────────────────────────────────────────┘   │
│              ┌───────────────────────────────────────────┐   │
│              │  Compute  Current  Permissions             │   │
│              │                                240          │   │
│              └───────────────────────────────────────────┘   │
└─────────────────────────────────────────────────────────────┘
```

```
┌──────────────────────────────┐  ┌──────────────────────────────┐
│ Derivation  Instance   160   │  │ Derivation  Instance   160   │
├──────────────────────────────┤  ├──────────────────────────────┤
│ Maximal  Permissions  150    │  │ Current  Permissions  140    │
└──────────────────────────────┘  └──────────────────────────────┘
```

FIG. 2

Site Security Policy　130

Current Permission Propositions　300

Policy Graphs　320

Equation　301

(P1 U P2) ∩ P3

Constraint Propositions　302

If X → Then Y

Principal　321

Traversal Method 322

Combination Method 323

Directed Graph　324

Access Control List 325

Maximal Permission Propositions　310

Permissions (Fig.5)　330

Equations　311

Constraint Proposition　312

Permissions/Node

FIG. 3

Policy Graph Directed Graph 324

Attribute
410

ALL

Node 400
* P_A

Manufacturer

Value
= IBM

420

401

entry = P_B 430
Access Control List
= System Admin RW
IBM 440

Value
=Netscape

Description
Author

Node
*
400

...

Value entry = φ
= IBM
402

Value entry=P_N
=Netscape

Application
Class

Value entry = P_C
= Games
403

Application
Name

Value entry = P_X
= Doom
404

FIG. 4

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.