(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第3992136号
(P3992136)
(45)発行日　平成19年10月17日(2007.10.17)　(24)登録日　平成19年8月3日(2007.8.3)

(51)Int.Cl.　F I
G06F 21/22 (2006.01)　G06F 9/06 66ON

請求項の数 13 (全 17 頁)

(21)出願番号　特願2001-382592(P2001-382592)
(22)出願日　平成13年12月17日(2001.12.17)
(65)公開番号　特開2003-186687(P2003-186687A)
(43)公開日　平成15年7月4日(2003.7.4)
審査請求日　平成15年9月2日(2003.9.2)

特許法第３０条第１項適用　平成１３年１０月３１日～１１月２日　社団法人情報処理学会主催の「コンピュータセキュリティシンポジウム２００１」において文書をもって発表

(73)特許権者　593165487
学校法人金沢工業大学
石川県石川郡野々市町扇が丘７番１号
(74)代理人　100105924
弁理士　森下　賢樹
(72)発明者　服部　進実
石川県石川郡野々市町扇が丘７番１号　学校法人金沢工業大学内
(72)発明者　千石　靖
石川県石川郡野々市町扇が丘７番１号　学校法人金沢工業大学内

審査官　石川　正二

最終頁に続く

(54)【発明の名称】ウイルス検出方法および装置

(57)【特許請求の範囲】
【請求項１】
データベース更新部が、ウイルス特有の動作に係る特徴コードにその特徴コードがウイルスとして使用された場合の危険性を示す重みを関連づけてデータベースに登録する工程と、
検査部が、検査対象ファイルをトレースして、前記データベースに登録された前記特徴コードとその重みを収集する工程と、
前記検査部が、前記収集された特徴コードをモジュールレベル、サブルーチンレベル、命令コードレベルおよびオペランドレベルのいずれかに分類する工程と、
危険度算出部が、前記モジュールレベルおよびサブルーチンレベルに分類された特徴コードの重みにもとづいて、その特徴コードがウイルス活動のトリガーになりうる危険度を算出する工程と、
前記危険度算出部が、前記命令コードレベルおよびオペランドレベルに分類された特徴コードの重みにもとづいて、その特徴コードによるウイルス活動に対する危険度を算出する工程と、
前記危険度算出部が、前記サブルーチンごとに、前記２種類の危険度を組み合わせて、前記サブルーチンの危険度を算出し、算出したサブルーチンの危険度にもとづいて、前記検査対象ファイルの危険度を算出する工程とを含むことを特徴とするウイルス検出方法。
【請求項２】
前記データベース更新部が、前記収集された特徴コードに関して、前記データベースに

格納された前記重みを更新する工程をさらに含むことを特徴とする請求項１に記載のウイルス検出方法。

【請求項３】

前記データベース更新部が、前記検査対象ファイルにウイルスが検出された場合に前記重みを更新することを特徴とする請求項２に記載のウイルス検出方法。

【請求項４】

前記危険度算出部は、前記算出したサブルーチンの危険度の内、もっとも高い危険度を前記検査対象ファイルの危険度とすることを特徴とする請求項１に記載のウイルス検出方法。

【請求項５】

前記データベース更新部は、前記モジュールレベルおよびサブルーチンレベルの特徴コードの重みを、その特徴コードの名前から把握された重みに設定し、前記命令コードレベルおよびオペランドレベルの特徴コードの重みを前記命令コートレベルの特徴コードの操作対象から把握された重みに設定することを特徴とする請求項１に記載のウイルス検出方法。

【請求項６】

ウイルス特有の動作に係る特徴コードにその特徴コードがウイルスとして使用された場合の危険性を示す重みを関連づけて格納したデータベースと、

検査対象ファイルをトレースして、前記データベースに登録された前記特徴コードとその重みを収集し、収集した特徴コードをモジュールレベル、サブルーチンレベル、命令コードレベルおよびオペランドレベルのいずれかに分類する検査部と、

前記モジュールレベルおよびサブルーチンレベルに分類された特徴コードの重みにもとづいて、その特徴コードがウイルス活動のトリガーになりうる危険度を算出し、前記命令コードレベルおよびオペランドレベルに分類された特徴コードの重みにもとづいて、その特徴コードによるウイルス活動に対する危険度を算出し、前記サブルーチンごとに、前記２種類の危険度を組み合わせて、前記サブルーチンの危険度を算出し、算出したサブルーチンの危険度にもとづいて、前記検査対象ファイルの危険度を算出する危険度算出部とを含むことを特徴とするウイルス検査装置。

【請求項７】

前記危険度に応じて、ウイルスの動作パターンを構成する前記特徴コードに関して、前記データベースに格納された前記重みを更新する更新部をさらに含むことを特徴とする請求項６に記載のウイルス検査装置。

【請求項８】

前記データベースは、ウイルスの動作パターンを格納し、前記更新部は、前記検査部により収集された特徴コードの組み合わせから特定された動作パターンを前記データベースに格納されたウイルスの動作パターンと比較して、その類似度に応じて、前記特定された動作パターンを構成する前記特徴コードの重みを更新することを特徴とする請求項７に記載のウイルス検査装置。

【請求項９】

前記危険度算出部は、前記モジュールレベルおよびサブルーチンレベルと前記命令コードレベルおよびオペランドレベルとで前記重みの評価を異ならせて前記モジュールレベルおよびサブルーチンレベルの危険度と前記命令コードレベルおよびオペランドレベルの危険度を算出することを特徴とする請求項６に記載のウイルス検査装置。

【請求項１０】

前記データベースに登録された前記モジュールレベルおよびサブルーチンレベルの特徴コードは、その特徴コードの名前から把握された重みに設定され、前記命令コードレベルおよびオペランドレベルの特徴コードは、前記命令コートレベルの特徴コードの操作対象から把握された重みに設定されることを特徴とする請求項６または９に記載のウイルス検査装置。

【請求項１１】

検査部が、ウイルス特有の動作に係る特徴コードにその特徴コードがウイルスとして使用された場合の危険性を示す重みを関連づけて登録したデータベースを参照して、検査対象ファイルから前記データベースに登録された前記特徴コードとその重みを収集する工程と、

前記検査部が、前記収集された特徴コードをモジュールレベル、サブルーチンレベル、命令コードレベルおよびオペランドレベルのいずれかに分類する工程と、

危険度算出部が、前記モジュールレベルおよびサブルーチンレベルに分類された特徴コードの重みにもとづいて、その特徴コードがウイルス活動のトリガーになりうる危険度を算出する工程と、

前記危険度算出部が、前記命令コードレベルおよびオペランドレベルに分類された特徴コードの重みにもとづいて、その特徴コードによるウイルス活動に対する危険度を算出する工程と、

前記危険度算出部が、前記サブルーチンごとに、前記２種類の危険度を組み合わせて、前記サブルーチンの危険度を算出し、算出したサブルーチンの危険度にもとづいて、前記検査対象ファイルの危険度を算出する工程とをコンピュータに実行させることを特徴とするコンピュータプログラム。

【請求項１２】

データベース更新部が、マクロウイルス特有の動作をとるために必要なコードと、そのコードがウイルスとして使用された場合の危険性を示す値を一つのレコードとしてデータベースに登録する工程と、

検査部が、マクロを一行ずつトレースし、前記登録したコードとその重みを収集し、コードの組み合わせを記録する工程と、

前記検査部が、前記収集された特徴コードをモジュールレベル、サブルーチンレベル、命令コードレベルおよびオペランドレベルのいずれかに分類する工程と、

前記検査部が、前記モジュールレベルおよびサブルーチンに分類された特徴コードの重みにもとづいて、その特徴コードがウイルス活動のトリガーになりうる危険度を算出する工程と、

前記検査部が、前記命令コードレベルおよびオペランドレベルに分類された特徴コードの重みにもとづいて、その特徴コードによるウイルス活動に対する危険度を算出する工程と、

前記検査部が、前記サブルーチンごとに、前記２種類の危険度を組み合わせて、前記サブルーチンの危険度を算出し、算出したサブルーチンの危険度にもとづいて、前記検査対象ファイルの危険度を算出する工程と、

前記検査部が、前記検査対象ファイルの危険度が基準となる値を超えたとき、そのマクロはウイルスであると判断する工程と、

前記データベース更新部が、ウイルスを検出した場合、収集したコードを基に重みを増減させて前記データベースを更新する工程とを含むことを特徴とするウイルス検出方法。

【請求項１３】

前記更新する工程は、過去に検出したウイルスとパターン比較を行い、類似点が多ければ、収集したコードの重みを増加させることを特徴とする請求項１２に記載のウイルス検出方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

この発明はコンピュータウイルスの検出技術、とくにコンピュータのプログラムファイルやデータファイルに感染するウイルスを検出する方法、装置およびシステムに関する。

【０００２】

【従来の技術】

情報処理振興事業協会（ＩＰＡ）が公表している国内のウイルス被害届出状況によると、１９９７年を境に急激に被害が増加しだしたことがわかる。１９９０年から１９９６年の

間では、年間１，０００件を越した年はわずか一年だけであったのに対し、１９９７年からは２，０００件を下回ることは無くなり、２０００年では１１，１０９件と加速的に増加している。

【０００３】

このようにウイルス被害件数が急増してきた要因は、パソコンやネットワークが広く普及したことによる感染機会の増加や、個人や企業のウイルス対策意識が低いことの他、ウイルス対策の技術的な面にもその要因があるといえる。なぜなら、１９９７年の被害の急増はマクロウイルスという新たな種のウイルスの出現によるものであり、近年突発的に被害が大きく増えたのも、ＶＢＳ（ビジュアルベーシック（商標）スクリプト）ウイルスやＷｉｎｄｏｗｓ（商標）ウイルスなどの新種の出現によるものだからである。

【０００４】

【発明が解決しようとする課題】

このように、従来のウイルス対策は非常に新種ウイルスに対して弱いことを意味しており、これはウイルス検出方法に問題があるといえる。従来のウイルス検出方法は、過去に発見されたウイルスに固有のコードとのパターンマッチングに依存しており、新たなウイルスが出現するたびにウイルス定義ファイルを更新する手間があるため、ユーザの対応に遅れが生じる。

【０００５】

本発明はこうした状況に鑑みてなされたものであり、新種ウイルスまたは変種ウイルスを効果的に検出するウイルス検出技術を提供することを目的とする。

【０００６】

【課題を解決するための手段】

本発明のある態様はウイルス検出方法に関する。この方法は、ウイルス特有の動作に係る特徴コードに危険性を示す重みを関連づけてデータベースに登録する工程と、検査対象ファイルをトレースして、前記データベースに登録された前記特徴コードを収集する工程と、前記収集された特徴コードの組み合わせにもとづいて、各特徴コードに関連づけられた前記重みを評価して、前記検査対象ファイルの危険度を算出する工程とを含む。ここで検査対象ファイルはプログラムファイルと、文書やマクロなどのデータファイルとを含む。

【０００７】

前記収集された特徴コードに関して、前記データベースに格納された前記重みを更新する工程をさらに含んでもよい。前記重みの更新は、前記検査対象ファイルにウイルスが検出された場合になされてもよい。前記検査対象ファイルにウイルスが含まれるかどうかの判定を外部から与えてもよい。また前記ウイルスの検出は、前記危険度にもとづいてなされてもよい。たとえば危険度が所定の基準値を超えた場合に、前記検査対象ファイルにウイルスが含まれると判定してもよい。

【０００８】

前記収集された特徴コードを階層的なレベルに分類した上でレベルによって前記重みの評価を異ならせて前記危険度を算出してもよい。階層的なレベルは、たとえばモジュール、サブルーチン、命令コード、およびオペランドといった処理コードの階層構造のレベルであり、モジュール、サブルーチンのような処理ルーチン単位レベルと、命令コード、オペランドのような下位のプリミティブなコマンドレベルとで重みの評価の仕方を異ならせ、その評価の組み合わせで前記危険度を算出してもよい。

【０００９】

本発明の別の態様はウイルス検査装置に関する。この装置は、ウイルス特有の動作に係る特徴コードに危険性を示す重みを関連づけて格納したデータベースと、検査対象ファイルをトレースして、前記データベースに登録された前記特徴コードを収集し、収集した特徴コードの組み合わせを動作パターンとして特定する検査部と、前記収集した特徴コードの組み合わせと前記重みにもとづいて前記動作パターンの危険度を算出する危険度算出部とを含む。

【００１０】

前記危険度に応じて、前記動作パターンを構成する前記特徴コードに関して、前記データベースに格納された前記重みを更新する更新部をさらに含んでもよい。たとえば前記危険度が所定の基準値以上である場合に、前記動作パターンをウイルスと判定して、前記重みを更新するが、前記危険度が基準値に満たない場合には、前記重みを更新しないようにしてもよい。

【００１１】

前記データベースは、ウイルスの動作パターンを格納し、前記更新部は、前記特定された動作パターンを前記データベースに格納されたウイルスの動作パターンと比較して、その類似度に応じて、前記特定された動作パターンを構成する前記特徴コードの重みを更新してもよい。また前記特定された動作パターンの危険度が基準値以上である場合、この動作パターンを新たなウイルスの動作パターンとして前記データベースに格納してもよい。

【００１２】

前記検査部は、前記収集した特徴コードを処理ルーチン単位とプリミティブなコマンド単位とに階層分けして分類し、前記危険度算出部は前記階層によって前記重みの評価を異ならせて前記危険度を算出してもよい。また前記危険度算出部は、前記収集した特徴コードを命令と操作対象の種別により区別し、その命令と操作対象の種別の組み合わせによって前記重みの評価を異ならせて前記危険度を算出してもよい。たとえば、ファイルの自動オープンの命令のように、命令自体にリスクを伴うものや、システムフォルダ内のファイル、テンプレートファイル、実行形式のファイルなどに対するアクションのように、操作対象にリスクを伴うものがあり、命令と操作対象の種別の組み合わせによって重みの評価の仕方を変えてもよい。

【００１３】

本発明のさらに別の態様はコンピュータプログラムに関する。このプログラムは、ウイルス特有の動作に係る特徴コードに危険性を示す重みを関連づけて登録したデータベースを参照して、検査対象ファイルから前記データベースに登録された前記特徴コードを収集する工程と、前記収集された特徴コードの組み合わせにもとづいて、各特徴コードに関連づけられた前記重みを評価して、前記検査対象ファイルの危険度を算出する工程とをコンピュータに実行させる。

【００１４】

前記データベースはサーバに設けられ、ネットワークを介して前記データベースを参照してもよい。また当該プログラムがインストールされたユーザ端末と前記サーバを含むシステムが構成されてもよい。また当該プログラムはネットワーク上のユーザ端末を巡回してウイルス検査を行うモバイルエージェントとして構成されてもよい。

【００１５】

本発明のさらに別の態様もウイルス検出方法に関する。この方法は、マクロウイルス特有の動作をとるために必要なコードと、そのコードがウイルスとして使用された場合の危険性を示す値を一つのレコードとしてデータベースに登録する工程と、マクロを一行ずつトレースし、前記登録したコードを収集し、コードの組み合わせを記録する工程と、収集したコードの組み合わせから、各重みを基にしてウイルスであるかどうかの判断値となる危険度を算出する工程と、危険度が基準となる値を超えたとき、そのマクロはウイルスであると判断する工程と、ウイルスを検出した場合、収集したコードを基に重みを増減させて前記データベースを更新する工程とを含む。前記更新する工程は、過去に検出したウイルスとパターン比較を行い、類似点が多ければ、収集したコードの重みを増加させてもよい。

【００１６】

なお、以上の構成要素の任意の組み合わせ、本発明の表現を方法、装置、サーバ、システム、コンピュータプログラム、記録媒体などの間で変換したものもまた、本発明の態様として有効である。

【００１７】

【発明の実施の形態】

本発明では、新種ウイルスまたは変種ウイルスを効果的に検出することを目的とし、新種ウイルス検出に適すとされるヒューリスティック検査法の概念を取り入れたウイルス検出システムを提案、構築する。また、実施の形態では新種、変種ウイルスの多さと、この先感染機会が増えるであろうことからマクロウイルスに対象を絞っている。

【００１８】

まず検査対象としたマクロウイルスについて、マクロウイルスの特徴、感染メカニズム、およびマクロウイルス検出方法を述べる。

【００１９】

マクロウイルスは、文書ファイルなどのデータファイルに付加しているマクロ部分に感染するウイルスである。コンピュータで扱うファイルは大きく分けると「プログラムファイル」と「データファイル」の２種類があり、従来のウイルスはデータファイルには感染しないという定説があったがマクロウイルスの登場により覆されている。現在、マクロ機能を持つアプリケーションプログラムは多数存在するが、中でもシェアが大きく、マクロ機能に制限が少なく安全性が低いものはウイルス製作者の標的にされている。また、ＯＬＥ（Object Linking and Embedding）オートメーション機能やＤＤＥ（Dynamic Data Exchange）機能を使って、異なるアプリケーションプログラム間の感染が可能であり、これにより感染力の非常に強いウイルスを出現させる結果となっている。

【００２０】

マクロウイルスはその作成が非常に容易であるということも重要な点である。従来のウイルスと比較するとはるかに多くの新種、変種が存在するのはこのためである。従来のウイルスの作成には、アセンブラなどの低級言語によるプログラミングやＯＳの知識などが必要であったが、マクロウイルスの場合は、マクロ言語そのものかビジュアルベーシック（Visual Basic）（商標）についての知識をある程度持っていれば十分である。

【００２１】

マクロウイルス自体はデータファイルであるため、マクロを動作させるアプリケーションを介して感染や発病をする。以下にマクロウイルスが感染を広げていく流れを示す。

（１）電子メールやダウンロードにより外部からコンピュータにウイルスが侵入する。

（２）ファイルを開くためにアプリケーションプログラムを起動する。

（３）アプリケーションプログラムがファイルを読込みマクロを実行する。

（４）ウイルスマクロが実行されると、標準テンプレートファイルに自分自身のコピーを書込む。

（５）その後、そのアプリケーションプログラムに読込まれたファイルには、ウイルスに感染したテンプレートが適応され、そのファイルのマクロもウイルスとなる。

【００２２】

先に述べたように、マクロウイルスには非常に多くの変種ウイルスが存在しており、従来の単にコードを比較するだけのウイルス検出方法では、全ての変種ウイルスを捕らえることは困難である。

【００２３】

そこで、本実施の形態ではマクロウイルスが感染を広げるために標準テンプレートに必ず自分自身のコピーを書込むことなど、マクロウイルスが特有の動作パターンを持つことに着目し、これらの動作パターンを捕らえることでマクロウイルスを検出する。また本実施の形態では、このマクロウイルス特有の動作パターンを捕らえるために、これから述べるヒューリスティック検査法の概念を取り入れている。

【００２４】

ヒューリスティック検査法というのは、検査対象のプログラムの中からコンピュータウイルス特有の動作パターンに必要となるコードを収集し、そのプログラム中にどれだけそのコードが含まれるか、そのコードがウイルスである可能性はどれだけあるかを、ウイルス検出システム自身に試行錯誤させてウイルスであるかどうかを判断させる技術である。

【００２５】

この技術を使えば従来のコード比較によるウイルス検出とは違い、新種ウイルスが出現す

る度にウイルス定義ファイルを更新する手間が無くなり、対応の遅れをなくすことができる。また、ウイルスの特性だけを取り上げるので、新種や変種ウイルスの検出には最適である。

【００２６】

本実施の形態に係るウイルス検出システムではヒューリスティック検査法の概念を取り入れ以下のようにしてウイルス特有の動作パターンを捕らえ、ウイルスであるかどうかの判定をしている。

（１）マクロウイルス特有の動作をとるために必要なコードと、そのコードがウイルスとして使用された場合の危険性を示す値(以下、重みという)などを１レコードとしてデータベースに登録する。

（２）マクロを一行ずつトレースし、登録したコードを収集する。このときコードがどのように組み合わさっていたのかを記録する。

（３）収集したコードの組み合わせから、各重みを基にしてウイルスであるかどうかの判断値となる危険度を算出する。

（４）危険度が基準となる値を超えたとき、そのマクロはウイルスであると判断する。

（５）ウイルスを検出した場合、このとき収集したコードや、過去にウイルスを検出した状態を基に重みを増減させてデータベースを更新する。

【００２７】

図１に示すように、本ウイルス検出システムは４つの処理ルーチンで構成されている。各ルーチンの働きとそれらの連携について述べる。

【００２８】

１．マクロ抜き出し処理ルーチン

このルーチンでは、マクロを含むファイルからマクロのみを抜きだす処理を行う。マクロをアプリケーションを介さずに読むにはＯＬＥ２の複合ファイルの仕組みや構造化記憶について理解する必要がある。図２に示すように、ＯＬＥ２の複合ファイルはファイル内の構造がファイルシステムのようになっており、ディレクトリに相当するストレージとファイルに相当するストリームで構成される。マクロや特にＶＢＡ（Visual Basic For Application）を使用しているファイルでは、それを扱うストリームを容易に特定することができるので、これを利用してマクロを再構築する。

【００２９】

ストリームに格納されているマクロの情報は、命令や、変数名を文字列として格納するデータ群や、それをどのように配置するかを表すデータ群で構成されている。このままでは、次のマクロのトレース処理を効率よく行えないため、一度マクロをトレースしやすい形態に再構築する。図３は、マクロを構成する文字列データ群を格納するストリームの一部である。これを規則に従い再構築しテキストに書き出したのが図４である。

【００３０】

２．マクロトレース処理ルーチン

マクロを一行ずつトレースしながら、データベースに登録した特徴コードを検査していく。ここでいう特徴コードとは、ウイルスの自己伝染、発病、潜伏機能を実行するために使われるであろうコードであり、データベースにはこの特徴コードとそれに対応した重みを記録しておく。レコードの詳細については後述の学習手法で説明する。

【００３１】

特徴コードが発見された場合は、図５に示す４つのレベルに分類してそれぞれの重みを危険度算出ルーチンに送る。すなわちモジュールレベル、サブルーチンレベル、命令コードレベル、および引数（以下、オペランドともいう）レベルの４つの階層化されたレベルである。コードがどのレベルであるかの判断はマクロを再構築する際に得た情報を利用する。モジュールレベルとサブルーチンレベルでは、その名前によってウイルス活動のトリガーとしての働きをし、命令コードレベルと引数レベルでは、その組み合わせによって大きくウイルスである危険度が変わる。

【００３２】

３．危険度算出ルーチン
危険度はそのままウイルスらしさを表し、基準値以上のものをウイルスであると判断する。危険度は、先のトレース処理（以下、トレース検査ともいう）で収集した重みを基に以下のような流れで算出する。
（１）まず、モジュールレベルの重みが得られるのでこれを記憶する。
（２）次にサブルーチンレベルの重みが得られるので、ここでウイルス活動のトリガーになり得ないかをモジュールレベルの重みと合わせて算出し、その値をＡとする。
（３）命令コードレベルの重みを得たときに、引数レベルの重みが同時に得られていれば、その組み合わせからウイルスの行動に対する値Ｂを算出する。引数レベルの重みが無い場合は、命令コードレベルの重みをＢとする。
（４）マクロをトレース中にサブルーチンの終了を検出した時点で、ＡとＢの組み合わせから危険度を算出する。全てのサブルーチンについて危険度を算出し、一番高い危険度をそのマクロの危険度とする。

【００３３】
４．データベース更新ルーチン
このルーチンでは、データベースに登録した重みを増減させる。重みは、ウイルスであるかどうかを判定するための基になる重要な値であり、この重みを巧く増減させることで、ウイルス検出精度の向上が期待できる。

【００３４】
重みを増減させるタイミングは、ウイルスを検出した時とし、そのコードがウイルスに使われた頻度、時間間隔を考慮して増減させるが、重みの減少させすぎによる検出ミスは避ける必要がある。その対策としては重みの下限を設定し、重みを増加させる量よりも、減少させる量を少なくしている。

【００３５】
また、後述の学習手法で説明するが、過去にウイルスを検出した時のマクロのパターンもデータベースに記録するため、マクロトレース処理ルーチンから送られたコードも全てデータベースに記録する。

【００３６】
次に学習手法について詳細に説明する。本ウイルス検出システムでは、マクロウイルスの本質的な動作パターンを突き詰めていくことが必要不可欠であり、そのために過去にウイルスを検出した経験を反映させる方法をとっている。また、ここでいう学習とは重みを適切な値に近づけるもので、データベースの自動更新がこれにあたる。

【００３７】
まずデータベースの構成を説明する。データベースは２つのテーブルからなり、１つは、特徴コードと重みの対を格納し、１つは過去に検出したウイルスマクロをパターン化したものを格納している。マクロのパターン化は、特徴コードに通し番号を付け、それを集めたものである。レコードの構成は次のようになっている。

【００３８】
特徴コード用テーブルのレコード
（１）通し番号
（２）基本値
（３）付加値
（４）特徴コードの文字列
（５）出現頻度

【００３９】
マクロパターン化用テーブルのレコード
（１）タイムスタンプ
（２）通し番号列の文字列
（３）出現頻度

【００４０】

これまで述べてきた重みは基本値のことをいっており、付加値は過去の経験を反映させる重みであり、ここで述べる学習処理で使用する。通し番号列は、特徴コード用テーブルの通し番号を集めて文字列にしたものである。

【００４１】

次に図６を参照しながら学習手順を示す。

（１）基本値による危険度を前述のように算出する。このときマクロのパターン化も同時に行う。

（２）次に、過去に検出したウイルスマクロとパターン比較を行う。類似点が多ければ、過去に同じような動作をするウイルスを検出していることになる。また、まったく同じ動作をしている部分があればそれは、ウイルスの本質的な動作であるとみなすことができ、そのコードの付加値を増加させる。

（３）すでにコンピュータ内にウイルスが広まっている場合も考慮し、全てが同じパターンである場合は付加値を増加させない。これは、特定のウイルスの影響を受けて重みが不適切に偏ることを防ぐためである。

（４）危険度に付加値を与え、新たな危険度を算出する。

（５）出現頻度の一番多いものを基準に、その差が一定値以上ついたコードの付加値を減少させる。

【００４２】

図７は、実施の形態に係るウイルス検出装置１０の構成図である。この構成は、ハードウェア的には、任意のコンピュータのＣＰＵ、メモリ、その他のＬＳＩで実現でき、ソフトウェア的にはメモリにロードされたウイルス検出機能のあるプログラムなどによって実現されるが、ここではそれらの連携によって実現される機能ブロックを描いている。したがって、これらの機能ブロックがハードウェアのみ、ソフトウェアのみ、またはそれらの組み合わせによっていろいろな形で実現できることは、当業者には理解されるところである。

【００４３】

ウイルス検出装置１０のマクロ抽出部１４、トレース検査部１６、危険度算出部１８、およびデータベース更新部２０は、ソフトウェア処理としては、それぞれ前述のウイルス検出システムにおけるマクロ抜き出し処理ルーチン、マクロトレース処理ルーチン、危険度算出ルーチン、およびデータベース更新ルーチンを実行するものである。

【００４４】

ウイルスデータベース２６は、特徴コードレコード２８とマクロパターンレコード３０を格納している。特徴コードレコード２８は、ウイルス特有の特徴コードに危険性を示す重みを関連づけたものであり、前述の特徴コード用テーブルのレコードである。マクロパターンレコード３０は、特徴コードの組み合わせで構成されるウイルスのパターンを示すものであり、前述のマクロパターン化用テーブルのレコードである。

【００４５】

特徴コードは図５のように４つのレベルに分けられて重みづけされている。ファイルの自動オープンのようにその動作自体がリスクを伴うようなモジュールやサブルーチンの場合、モジュール名またはサブルーチン名から危険度を把握できる。またシステムフォルダ内のファイルの操作、テンプレートファイルの操作、実行形式のファイルのオープンなど操作対象の種別から危険度を把握できるものもある。したがって命令と操作対象を別々に登録しておいて、その組み合わせについて重みづけをすることが必要である。また特徴コードを階層的にレベル分けして、モジュールやサブルーチンといった大きな処理単位で危険性を判断するとともに、命令コードやオペランドといったよりプリミティブな単位で危険性を判断することが必要である。たとえばファイルの自動オープンを使用せずに、ダイアログを表示して、ユーザにダイアログのボタンをクリックさせることで実質的にファイルの自動オープンを行うことも可能である。したがってモジュール名やサブルーチン名だけに頼った危険性の判断だけでは不十分であり、より下位の命令コードやオペランドのレベルでの危険性の判断が要求される。

【００４６】
図８を参照して、ウイルス検出装置１０によるウイルス検出手順の大まかな流れを説明する。マクロ抽出部１４は、検査対象ファイル１２を読込む（Ｓ１０）。次にマクロ抽出部１４は、検査対象ファイルからマクロ情報を抜き出し、マクロ情報をソースコードの状態に再構築する（Ｓ１２）。トレース検査部１６は再構築されたマクロのソースコードを一行ずつトレースして、ウイルスの特徴コードが含まれていないかどうか検査する（Ｓ１４）。このときトレース検査部１６は、特徴コードの組み合わせから構成される動作パターンを特定する。
【００４７】
危険度算出部１８は、トレース検査部１６により抽出された特徴コードの重みにもとづいて、マクロの危険度を算出し、マクロがウイルスであるかどうかを判定する（Ｓ１６）。ウイルス判定により、ウイルスが検出された場合（Ｓ１８のＹ）、データベース更新部２０は、トレース検査部１６により特定されたマクロの動作パターンを構成する特徴コードに関して、ウイルスデータベース２６の特徴コードレコード２８の重みを更新する（Ｓ２０）。ウイルスが検出されない場合（Ｓ１８のＮ）、ウイルスデータベース２６の更新は行わない。
【００４８】
図９はマクロの再構築処理Ｓ１２の詳細な手順を示すフローチャートである。既に図３に示したように、マクロ情報は検査対象の文書ファイルのバイナリデータ中に散らばっており、このままではトレースが難しい。マクロ抽出部１４は検査対象ファイルからマクロ情報を抽出して、図４のようなソースコードの形に組み上げる再構築の処理を行う。まず検査対象ファイルが構造化記憶を利用したものであるかどうかを調べる（Ｓ３０）。構造化記憶を利用していない場合（Ｓ３０のＮ）、ウイルスの検査を行わずに終了する。構造化記憶を利用している場合（Ｓ３０のＹ）、検査対象ファイルにマクロが含まれるかどうかを調べる（Ｓ３２）。マクロが含まれない場合（Ｓ３２のＮ）、ウイルスの検査を行わずに終了する。マクロが含まれる場合（Ｓ３２のＹ）、マクロ情報を組み上げ、ソースコードの状態にして書き出す（Ｓ３４）。
【００４９】
図１０はマクロのトレース検査処理Ｓ１４の詳細な手順を示すフローチャートである。トレース検査部１６は、再構築されたマクロのソースコードを一行ずつトレースしながら、ウイルスデータベース２６を参照して、登録された特徴コードレコード２８に一致するものがあるかどうか調べ、一致した特徴コードとその重みを収集する（Ｓ４０）。またマクロのサブルーチンごとに、そのサブルーチンを構成する特徴コードの種類を動作パターンとして記憶する（Ｓ４２）。
【００５０】
図１１はウイルス判定処理Ｓ１６の詳細な手順を示すフローチャートである。危険度算出部１８は、トレース検査部１６により抽出された特徴コードの重みにもとづいて、マクロのサブルーチンごとに危険度を算出する。トレース検査部１６は、特徴コードを図５に示した４つのレベルに分類している。危険度算出部１８は、モジュールレベルの重みＭとサブルーチンレベルの重みＳとを組み合わせて、ウイルス活動のトリガーになりうる危険度ＶＴを算出する（Ｓ５０）。たとえば、ＶＴ＝ｍａｘ（Ｍ，Ｓ）とする。次にサブルーチンの一つ一つの命令について、命令コードレベルの重みＩとオペランドレベルの重みＯとを組み合わせて、ウイルス活動に対する危険度ＶＡを算出する（Ｓ５２）。たとえばＶＡ＝Ｉ×Ｏとする。
【００５１】
次にサブルーチンごとに、評価した重みの階層レベルが異なる２種類の危険度ＶＴ、ＶＡを組み合わせてサブルーチンの危険度を算出する（Ｓ５４）。たとえば、ＶＴの値とサブルーチン内の最も大きいＶＡの値の和をそのサブルーチンの危険度とする。マクロのすべてのサブルーチンについてサブルーチンの危険度が算出されると、それらのサブルーチンの危険度の内、もっとも高い危険度をマクロの危険度とする（Ｓ５６）。マクロの危険度

が所定の基準値を越える場合（Ｓ５８のＹ）、そのマクロはウイルスであると判定し（Ｓ６０）、そうでない場合（Ｓ５８のＮ）、そのマクロをウイルスとは判定しない。

【００５２】

図１２はデータベース更新処理Ｓ２０の詳細な手順を示すフローチャートである。データベース更新部２０は、マクロパターン登録部２２と重み更新部２４を含む。マクロがウイルスと判定された場合に、マクロパターン登録部２２はトレース検査部１６により特定されたマクロの動作パターンを新たなマクロパターンレコード３０としてウイルスデータベース２６に登録し、重み更新部２４はその動作パターンを構成する特徴コードに関して、特徴コードレコード２８の重みを更新する。

【００５３】

学習アルゴリズムの基本方針は、多種のウイルスに共通する動作パターンはウイルスの本質的な動作とみなし、その動作パターンを構成する特徴コードの重みを増加させることである。たとえば、ウイルスＡ、Ｂ、Ｃの動作が以下であるとする。

【００５４】

（Ａ）ウイルスＡの動作：
（Ａ－１）ファイルオープンをトリガーとする、
（Ａ－２）レジスタにＸを書込む、
（Ａ－３）標準テンプレートファイルを書き換える。

【００５５】

（Ｂ）ウイルスＢの動作：
（Ｂ－１）ファイルオープンをトリガーとする、
（Ｂ－２）システムファイルを削除する、
（Ｂ－３）標準テンプレートファイルを書き換える。

【００５６】

（Ｃ）ウイルスＣの動作：
（Ｃ－１）アプリケーションの起動をトリガーとする、
（Ｃ－２）ファイルＹを改ざんする、
（Ｃ－３）標準テンプレートファイルを書き換える。

【００５７】

このとき、「標準テンプレートファイルを書き換える」という動作パターンは３種のウイルスＡ、Ｂ、Ｃに共通しており、「ファイルオープンをトリガーとする」という動作パターンは２種のウイルスＡ、Ｂに共通している。そこで「標準テンプレートファイルを書き換える」という動作パターンを構成する特徴コードについては重みの増加量を大きくとり、「ファイルオープンをトリガーとする」という動作パターンを構成する特徴コードについては重みの増加量を小さくとる。

【００５８】

重みは基本値と付加値に分かれている。基本値は特徴コードに対して予想される危険度を示す固定の値であり、付加値は初期値をゼロとして、学習アルゴリズムにより更新される値である。ウイルスに頻繁に使用される特徴コードについては付加値が増加し、ほとんどウイルスに使用されない特徴コードについては付加値が減少する。特徴コードの危険度を示す重みは基本値と付加値の和で与えられるため、過去に発見されたウイルスの動作パターンによって重みが更新されることになる。

【００５９】

具体的な学習手順を説明する。発見されたウイルスの動作パターンについて、その動作パターンを構成する特徴コードの種類ごとに出現回数をカウントする（Ｓ５０）。たとえばウイルスの動作パターンが３種の特徴コードａ、ｂ、ｃから構成される場合、３種の特徴コードａ、ｂ、ｃのそれぞれの出現回数のカウント数を１だけインクリメントする。ウイルスのすべての動作パターンについて、その動作パターンに含まれる特徴コードの出願回数をこのようにカウントする。次に、特徴コードの付加値をこの出現回数のカウント数に比例して増加させる（Ｓ５２）。たとえば出現回数に所定の増加係数をかけた値を新たな

付加値とする。これにより、多種のウイルスに共通する動作パターンの特徴コードほど付加値が大きくなる。

【００６０】

出現回数のカウント数が特定数以上である場合（Ｓ５４のＹ）、カウント数が極端に大きい特徴コードの付加値を減少させる（Ｓ５６）。たとえば特徴コード間の出現回数の差に比例して付加値を減少させる。すなわち出現回数の差に所定の減少係数をかけた値を付加値から差し引く。付加値の減少は、特定のウイルスの影響を受けて重みが偏るのを防ぐために行われる。

【００６１】

以上述べたように、実施の形態では検査対象にマクロウイルスを取り上げ、ヒューリスティック検査法の概念を取り入れた新種ウイルスに有効な検出方法を提案した。マクロウイルスに特有のコードをデータベース化し、その危険性を学習するシステムであるため、単純なコードのパターンマッチングによるウイルス検出の方法とは違って、既存ウイルスの一部を変えた変形型のウイルスにも柔軟に対応でき、またこれまでになかった新種のウイルスであっても検出することが可能となる。

【００６２】

以上、本発明を実施の形態をもとに説明した。実施の形態は例示であり、それらの各構成要素や各処理プロセスの組合せにいろいろな変形例が可能なこと、またそうした変形例も本発明の範囲にあることは当業者に理解されるところである。以下そのような変形例を説明する。

【００６３】

実施の形態では、新種、変種のウイルスが多く出現するマクロウイルスを例にあげてウイルス検出手法を説明したが、マクロウイルスに限定する趣旨ではなく、本発明はプログラムコードに感染するウイルス一般にも適用可能である。

【００６４】

実施の形態では、ウイルス特有のコードを収集したデータベースが検査対象のコンピュータ内に設けられが、このようなデータベースがサーバに設けられ、ネットワークを介して必要なデータを検索したり、ダウンロードする構成でもよい。また本実施の形態のウイルス検出プログラムを有するモバイルエージェントがユーザのコンピュータに配信されて、分散型のウイルス検出が行われてもよい。このようなエージェントを用いる構成においても、ウイルスコードのデータベースはユーザのコンピュータ内に設けられてもサーバに設けられてもよい。またエージェントプログラムのデータ領域にそのようなデータベースの少なくとも一部が含まれていてもよい。

【００６５】

またウイルス検出システムは、コンピュータのハードディスク等をスキャンしてウイルスの検査、除去を行うウイルス検疫ソフトウェアのような独立したアプリケーションとして提供されてもよい。その場合、ＣＤ－ＲＯＭなどの記録媒体から読みとられたデータやネットワークを介してダウンロードされたデータがハードディスクに書込まれるときに、オンデマンドでデータをスキャンしてもよい。また、ウイルス検出システムは、マクロ機能をもつアプリケーションに組み込まれる形態で提供されてもよい。この場合、アプリケーションがマクロファイルをオープンする際、本ウイルス検出システムによりマクロの危険度を測定して、マクロファイルをオープンして実行する前に危険度をユーザに通知して警告を与えてもよい。

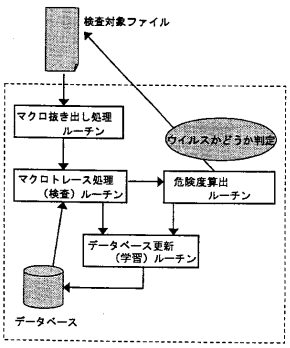
【００６６】

上記の学習手順では、ウイルスが検出された場合に重みの更新が行われたが、ウイルスかどうかの判定は、外部から与えられてもよい。たとえばユーザがウイルスかどうかを判断して、判定結果をウイルス検出システムに与えてもよい。またデータベース更新部は、外部から提供されるウイルス定義ファイルを用いて、危険度算出部によるウイルス判定の成否を評価してもよい。

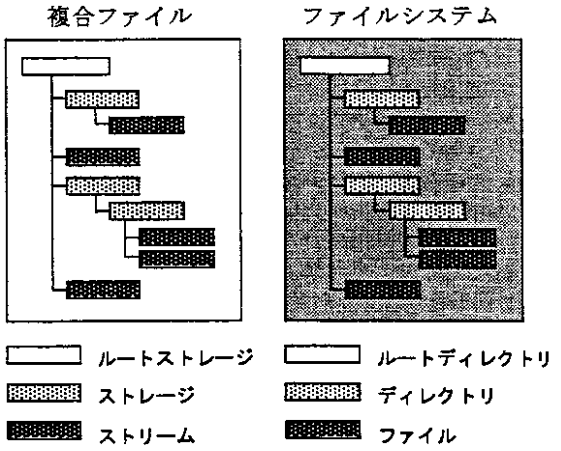
【００６７】

【発明の効果】
本発明によれば、新種または変種のウイルスを効果的に検出することができる。
【図面の簡単な説明】
【図１】　実施の形態に係るウイルス検出システムの全体構成図である。
【図２】　ウイルス検査対象ファイルの一例である複合ファイルのデータ構造を説明する図である。
【図３】　図２の複合ファイルのストリームに格納されているマクロ情報を示す図である。
【図４】　図３のマクロ情報を再構築してテキストに書き出した状態を示す図である。
【図５】　ウイルスの特徴コードに対応づける重みのレベル分けを説明する図である。
【図６】　図１のデータベースの学習手順を説明する図である。
【図７】　実施の形態に係るウイルス検出装置の構成図である。
【図８】　ウイルス検出装置におけるウイルス検出手順を示すフローチャートである。
【図９】　図８のマクロの再構築処理の詳細な手順を示すフローチャートである。
【図１０】　図８のマクロのトレース検査処理の詳細な手順を示すフローチャートである。
【図１１】　図８のウイルス判定処理の詳細な手順を示すフローチャートである。
【図１２】　図８のデータベース更新処理の詳細な手順を示すフローチャートである。
【符号の説明】
１０　ウイルス検出装置、　１２　検査対象ファイル、　１４　マクロ抽出部、　１６　トレース検査部、　１８　危険度算出部、　２０　データベース更新部、　２２　マクロパターン登録部、　２４　重み更新部、　２６　ウイルスデータベース、　２８　特徴コードレコード、　３０　マクロパターンレコード。

【図１】



【図２】



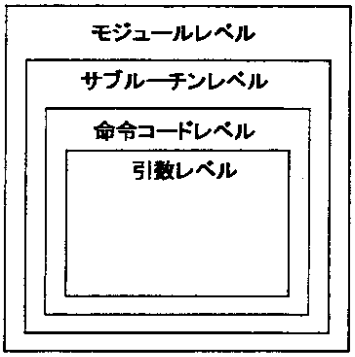
【図３】

```
10 00 08 04 53 61 6D 65 70 6C 65 6E D9 2A .♪....Sameplen♪*
04 04 4D 41 49 4E 88 2C 10 00 09 04 66 69 ....MAIN・....fi
4D 61 63 72 6F 31 54 10 00 09 04 67 6C 6F MacroITT....glo
61 63 72 6F FC A3 10 00 09 04 4D 61 63 72 bMacro.j....Macr
69 6C 65 55 76 10 00 09 04 65 72 72 43 61 oFileUv....errCa
68 74 73 60 10 00 09 00 57 6F 72 64 42 61 ughts`....WordBa
63 F1 92 10 00 0F 00 46 69 6C 65 53 75 60 sic・....FileSum
72 79 49 6E 66 6F 59 EE 10 00 06 00 55 70 maryInfoY・...Up
74 65 D2 FC 10 00 03 04 64 6C 67 2D 84 10 date♪.....dlg-・・
00 44 69 61 6C 6F 67 52 65 63 6F 72 64 92 ...DialogRecord♪
00 09 00 43 75 72 56 61 6C 75 65 73 C8 A4 .....CurValues♪、
09 00 44 69 72 65 63 74 6F 72 79 6D 47 10 ....DirectorymG.
00 46 69 6C 65 4E 61 6D 65 6A C3 10 00 05 ...FileNamejテ...
43 61 73 65 35 0A 10 00 06 80 80 00 FF 03 .UCase5.........
52 69 67 68 74 24 79 31 10 00 0F 80 80 00 ..Right$y1......
00 00 4D 61 63 72 6F 46 69 6C 65 4E 61 6D ....MacroFileNam
9C 4A 10 00 0A 80 80 00 FF 03 00 00 4D 61 e$圧.........Ma
6F 4E 61 6D 65 24 8D 01 10 00 09 00 00 69 croName$・.....i
53 61 76 65 41 73 CE EE 10 00 07 04 53 65 leSaveAs*・...Se
```

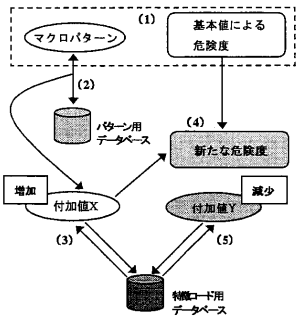
【図４】

```
Sub MAIN
Dim fileMacro
Dim MacroFile
On Error GoTo -1: On Error GoTo errCaught
WordBasic.FileSummaryInfo Update:=1
Dim Word: dlg = WordBasic.DialogRecord.FileSummaryInfo(False)
WordBasic.CurValues.FileSummaryInfo dlg
fileMacro = dlg.Directory + "¥" + dlg.FileName + ":autoOpen"
MacroFile = UCase(WordBasic.[Right$](WordBasic.[MacroFileName$]
If MacroFile = "sample.DOT" Then
MsgBox "sample.doc"
Else
MsgBox "test.doc"
End If
SetString
GoTo bye
errCaught:
bye:
On Error GoTo -1: On Error GoTo 0
End Sub
Sub SetString
Dim i
```

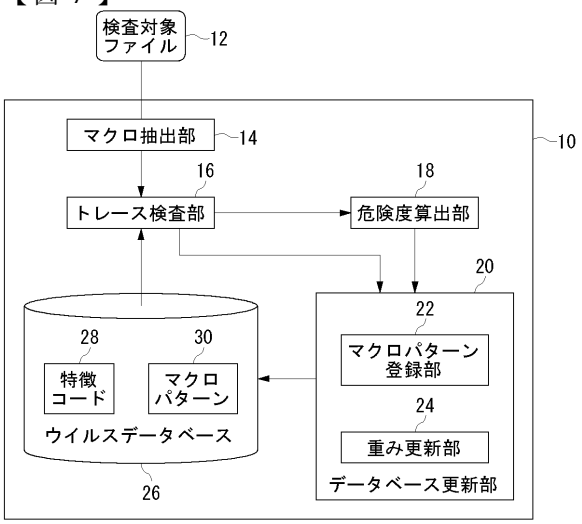
【図５】



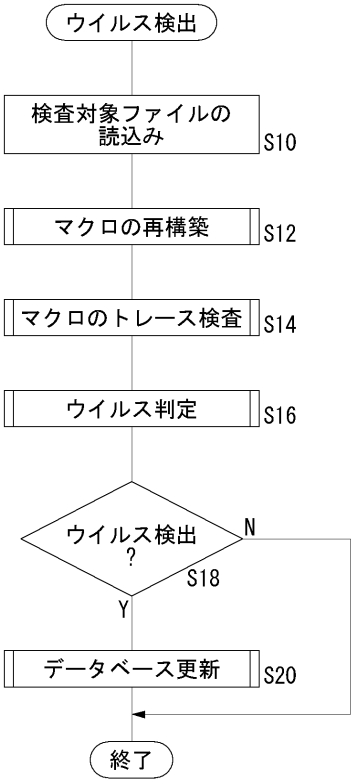
【図６】



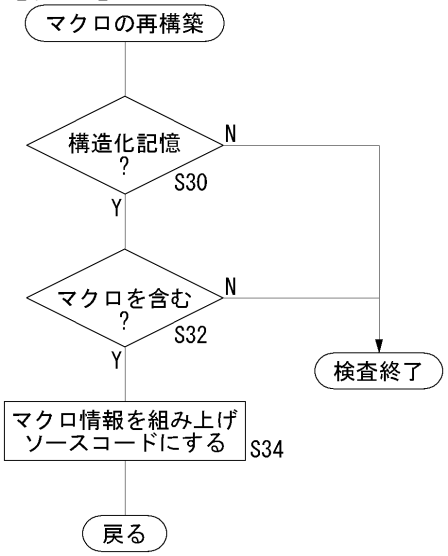
【図７】



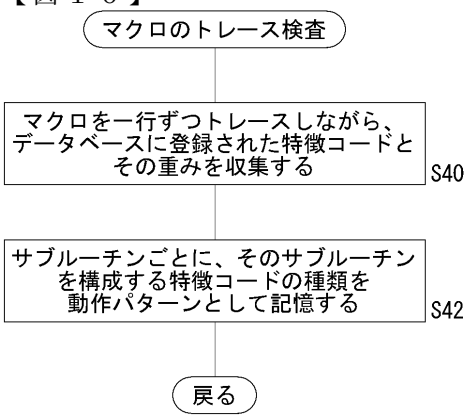
【図８】


ウイルス検出
検査対象ファイルの読込み
S10
マクロの再構築
S12
マクロのトレース検査
S14
ウイルス判定
S16
ウイルス検出？
S18
N
Y
データベース更新
S20
終了

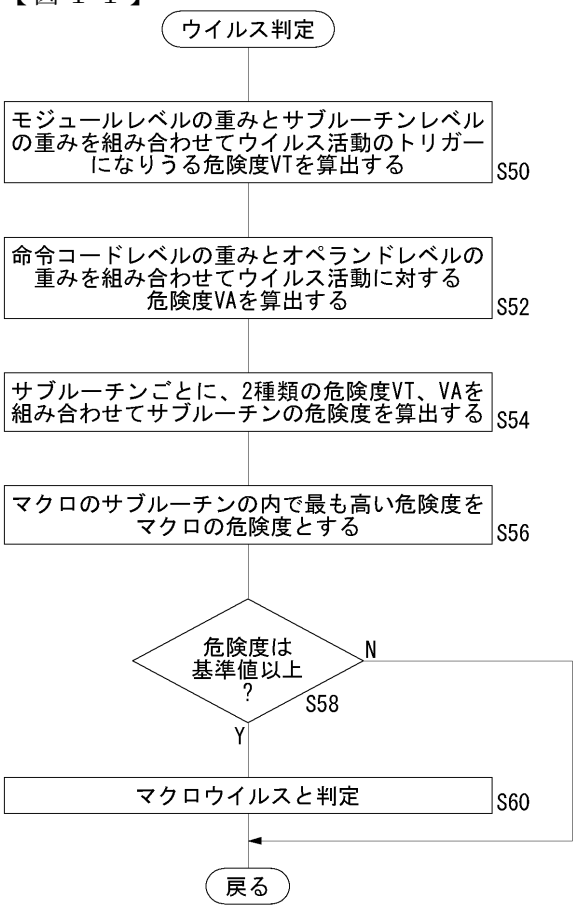
【図９】


マクロの再構築
構造化記憶？
S30
N
Y
マクロを含む？
S32
N
Y
検査終了
マクロ情報を組み上げソースコードにする
S34
戻る

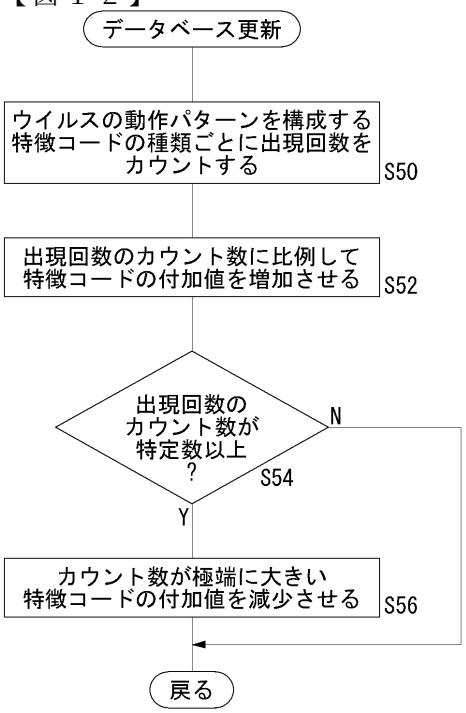
【図１０】


マクロのトレース検査
マクロを一行ずつトレースしながら、データベースに登録された特徴コードとその重みを収集する
S40
サブルーチンごとに、そのサブルーチンを構成する特徴コードの種類を動作パターンとして記憶する
S42
戻る

【図１１】


ウイルス判定
モジュールレベルの重みとサブルーチンレベルの重みを組み合わせてウイルス活動のトリガーになりうる危険度VTを算出する
S50
命令コードレベルの重みとオペランドレベルの重みを組み合わせてウイルス活動に対する危険度VAを算出する
S52
サブルーチンごとに、2種類の危険度VT、VAを組み合わせてサブルーチンの危険度を算出する
S54
マクロのサブルーチンの内で最も高い危険度をマクロの危険度とする
S56
危険度は基準値以上？
S58
N
Y
マクロウイルスと判定
S60
戻る

【図１２】


データベース更新
ウイルスの動作パターンを構成する特徴コードの種類ごとに出現回数をカウントする
S50
出現回数のカウント数に比例して特徴コードの付加値を増加させる
S52
出現回数のカウント数が特定数以上？
S54
N
Y
カウント数が極端に大きい特徴コードの付加値を減少させる
S56
戻る

フロントページの続き

(56)参考文献 渡部 章, "コンピュータウイルス事典", 日本, 株式会社オーム社, １９９３年１１月２５日, 第1版, p.463, 単行本1997-00312-001
"モバイルワクチンシステムの学習機能の構築と検証", 佐藤 伸介, 千石 靖, 服部 進実, 第60回(平成12年前期)全国大会講演論文集(3) , 日本, 社団法人情報処理学会, ２０００年 3月１４日, p.475, p.476, 予稿集2000-00013-001

(58)調査した分野(Int.Cl., ＤＢ名)
G06F 21/22