

アプリ開発の実務を踏まえた 不正指令電磁的記録に関する罪の一考察

—コインハイブ事件を契機として—

小 田 啓 太
西 貝 吉 晃

1. 導入

本稿では、アプリ⁽¹⁾開発の実務を踏まえた不正指令電磁的記録に関する罪についての解釈を展開する。

アプリ開発の実務では、過去に作られた多くのプログラムの全部または一部を再利用する形でより複雑なものが開発されている。このとき、依存している先の過去に作られたプログラムは、何十年前に作られたものであったり数日前に作られたものであったり大企業の作ったものであったり個人が作ったものであったりと様々である。コインハイブ事件で問題となったアプリも例外ではない。同事件においては、被告人とは別に仮想通貨のマイニング機能を作成したライブラリ提供元が別に存在している。こうした事情を不正指令電磁的記録に関する罪の規定の解釈・適用上、どのように考えるべきか。

本稿執筆時に上告中である同事件の高裁判決に対しては様々な評価がなされている。本稿は、アプリ開発実務を踏まえて考えると、同事件のプログラムコードについては不正指令電磁的記録性のうちの反意図性が欠けるのではないかと考える。既に本件において反意図性が欠けると

(1) アプリケーションソフトウェアの略。IT用語辞典バイナリ「アプリケーションソフトとは、ソフトウェアの分類のうち、特定の作業を行うために使用されるソフトウェアの総称である。(略)」(<https://www.sophia-it.com>)

している評釈は存在しているが⁽²⁾、ここでは、立法担当者の解説⁽³⁾を基礎にして、同解釈が反意図性要件に求める規範的判断を技術の面から具体化させた解釈論的考察を行う。

(1) コインハイブ事件の概要

ア 事案

コインハイブ事件においては、仮想通貨をマイニングするプログラムコードを自身の管理するウェブサイトを設置する行為が不正指令電磁的記録保管罪(刑法168条の3)に問われた。被告人は、ウェブサイトの閲覧者の同意を得ずに、同人のコンピュータに仮想通貨の取引履歴の承認作業等の演算を行わせて、その演算機能を提供したことによる報酬を取得しようとしていた。本件プログラムコード⁽⁴⁾は、コインハイブから提供されるマイニングを実行するための専用スクリプトをライブラリとして呼び出しており、これはJavaScriptで書かれていた。マイニングの報酬の7割を登録者が、残り3割をコインハイブ側が取得し、サイト閲覧者への還元はなされなかった。被告人は、コインハイブに登録し、上記のスクリプトを自身のサイトに置いた上で、本件プログラムコードに設けられたサイト閲覧時の閲覧者の電子計算機のCPU使用率調整のための設定値を0.5と設定した⁽⁵⁾。このサイトでは、本件プログラムコードが実行され、マイニングが行われることの表示はなされなかった。

-
- (2) 永井善之「不正指令電磁的記録概念について」金沢法学63巻1号79頁(2020)、岡部天俊「不正指令電磁的記録概念と条約適合的解釈：いわゆるコインハイブ事件を契機として」北大法学70巻6号55頁(2020)、渡邊卓也「不正指令電磁的記録に関する罪における反「意図」性の判断」情報ネットワークLR19巻16頁(2020)。
- (3) 杉山徳明＝吉田雅之「『情報処理の高度化等に対処するための刑法等の一部を改正する法律』について(上)」曹時64巻4号(以下「杉山＝吉田」という)1頁(2012)。
- (4) IT用語辞典バイナリ「コードとは、(略)プログラム言語の命令を記述した文書であるソースコードのことである。(略)」
- (5) 裁判所によると、この設定値の場合、マイニングを実行しても閲覧者のPCの処理速度が極端に遅くなるものではなかったようである。

1 審の横浜地判平成31年3月27日判時2446号78頁が反意図性を認めつつも、不正性を認めずに無罪判決を下したので検察官が控訴したところ、東京高判令和2年2月7日判時2446号71頁も、反意図性を認めた(1審と異なり不正性も認めたため有罪判決を下した)。

イ 高裁判決

高裁判決のうちの反意図性についての判示部分に限って紹介したい。まず、高裁は、反意図性について次のように抽象論を展開した。

「プログラムの反意図性は、当該プログラムの機能について一般的に認識すべきと考えられるところを基準とした上で、一般的なプログラム使用者の意思に反しないものと評価できるかという観点から規範的に判断されるべきである。原判決は、本件プログラムコードが、その機能を認識した上で実行できないことから、反意図性を肯定しているが、一般的な電子計算機の利用者は、電子計算機の使用にあたり、実行されるプログラムの全ての機能を認識しているわけではないものの、特に問題のない機能のプログラムが、電子計算機の使用に付随して実行されることは許容しているといえるから、一般的なプログラム使用者が事前に機能を認識した上で実行することが予定されていないプログラムについては、そのような点だけから反意図性を肯定すべきではなく、そのプログラムの機能の内容そのものを踏まえ、一般的なプログラム使用者が、機能を認識しないまま当該プログラムを使用することを許容していないと規範的に評価できる場合に反意図性を肯定すべきである。」

こうした法解釈を踏まえて、あてはめとして次のように述べた。本稿の問題関心から、反意図性に限って、その判断を紹介する。「一般的に、ウェブサイト閲覧者は、ウェブサイトを閲覧する際に、閲覧のために必要なプログラムを実行することは承認していると考えられるが、本件プログラムコードで実施されるマイニングは、ウェブサイトの閲覧のために必要なものではなく、このような観点から反意図性を否定することができる事案ではない。その上、本件プログラムコードの実行によって行われるマイニングは、閲覧者の電子計算機に一定の負荷を与えるものであるのに、このような機能の提供に関し報酬が発生した場合にも閲覧者には利益がもたらされないし、マイニングが実行されていることは閲覧

中の画面等には表示されず、閲覧者に、マイニングによって電子計算機の機能が提供されていることを知る機会やマイニングの実行を拒絶する機会も保障されていない。

このような本件プログラムコードは、プログラム使用者に利益をもたらさないものである上、プログラム使用者に無断で電子計算機の機能を提供させて利益を得ようとするものであり、このようなプログラムの使用を一般的なプログラム使用者として想定される者が許容しないことは明らかといえるから、反意図性を肯定した原判決の結論に誤りはない。

なお、原審において、弁護人は、本件プログラムコードがウェブ閲覧時に断りなく実行されることが普通に行われているジャバスクリプトのプログラムであり、この種のプログラムについては、閲覧者が承諾していると考えられる旨主張しているが、前記のとおり、プログラムの反意図性は、その機能を踏まえて認定すべきであるから、ジャバスクリプトのプログラムというだけで反意図性を否定することはできない。」

(2) 規範的構成要件要素としての反意図性要件

以上の東京高裁の法解釈は、規範的判断により、処罰範囲を限定しようとしている点で立法担当者の見解と軌を一にしている。そこで、立法担当者の見解を分析してみたい。2011年の刑法改正⁽⁶⁾で入った不正指令電磁的記録に関する罪についての立法担当者の解説(杉山=吉田)は、反意図性、すなわち「意図に沿うべき動作をさせず、又はその意図に反する動作をさせる(第168条の2第1項1号)」という文言について「①当該プログラムの機能の内容や機能に関する説明内容、想定される利用方法等を総合的に考慮して、その機能につき一般に認識すべきと考えられるところを基準として規範的に判断することとなる。」と述べる。そして、あてはめの具体例として、やや長く引用すれば「例えば、②市販されているソフトウェアの場合、電子計算機の利用者は、そのプログラムの指令によって電子計算機が行う基本的な動作については当然認識している

(6) 情報処理の高度化等に対処するための刑法等の一部を改正する法律 法律第七十四号(平二三・六・二四)

ものと考えられる上、それ以外の詳細な機能についても、使用説明書等に記載されるなどして、通常、使用者が認識し得るようになっているのであるから、そのような場合、仮に使用者がこのような機能を現実には認識していなくても、そのプログラムによる電子計算機の動作は、「使用者の意図に反する動作」には当たらないこととなる。③逆に、フリーソフトの中には、使用説明書が付されていないものもあり得るが、その場合であっても、当該ソフトウェアの機能は、その名称や公開されているサイト上での説明等により、通常、使用者が認識し得るようになっていることから、使用説明書が付されていないというだけで、「使用者の意図に反する動作」に当たることとなるわけではない。」と書かれている(①から③は筆者等による挿入)。

①においては、規範的判断をすべきであることが強調されている。規範的判断には結論に至る判断過程に多くの要素が入り込むことによる、結論に対する予見可能性の低下というデメリットもあるが、ある程度、「規範的」に判断することによる処罰範囲の限定という効用も期待されるところではある。本稿では、立法担当者のいう「規範的」判断を、その例示も含めて詳細に分析していきたい。

2. 規範的判断の具体化—杉山＝吉田を読み解く

1. で問題提起した規範的判断の内実を得るための重要な要素は、前記②、③の記述の解釈である。立法担当者の考える規範的判断の具体化⁽⁷⁾は具体例等を通して表現される部分の精読による必要がある。本稿では、反意図性を否定する方向で使われている「使用説明書等に記載されるなどして、通常、使用者が認識し得るようになっているのである」という表現部分に着目したい。この表現だけからもいくつかの重要な考慮要素が抽出されると思われるからである。

(7) 渡邊・前掲注(2)が主張する規範的判断の具体化論とも抽象的には同じ方向性を向いていると思われる。

(1) 反意図性を判断するために必要なプログラムの範囲

前提として、不正指令「電磁的『記録』」である以上、保存されたコンピュータ・データを指すのであり、電子計算機に指令を与えるデータとはプログラム⁽⁸⁾であるから、何らかのプログラムが問題となることは明らかである。しかし、その動作を問題とするときには被告人の保存したファイルのみをさすものとは限らない。つまり、まず問題となるのが、不正指令電磁的記録か否かが問題となる際の、その動作を判断するために必要となるプログラムの範囲なのである。

一般的に、プログラムを作成する際には、ライブラリ⁽⁹⁾と呼ばれる、他のプログラムに組み込まれることが想定されたプログラムを組み合わせることや、別のプログラムがプログラムから呼び出されることなどがほぼ必ず行われている。その一部はオペレーティングシステム⁽¹⁰⁾に付随していたり、別にインストールされたりする。また、本件で扱うようなJavaScriptで書かれたプログラムは、ウェブページ内でブラウザのJavaScriptエンジンに解釈されて呼び出される形で動作する。近年では、Web APIという技術で、プログラムAがインターネットを介してつながっている国外のサーバ上のプログラムBを呼び出して実行することもある。

結局のところ、プログラムの動作というときには、協調して動作するすべてのプログラムを含めた上で、組み合わせられた結果の動作を指すものと考えることになる。それゆえ、その挙動が問題となるプログラムの

(8) IT用語辞典バイナリ「プログラムとは、計算機に動作をさせるため用意された、処理手順を指示する一連の命令の集まりのことである。(略)」

(9) IT用語辞典バイナリ「ライブラリとは、プログラム言語において、ある特定の機能を持つプログラムを定型化して、他のプログラムが引用できる状態にしたものを、複数集めてまとめたファイルのことである。(中略)ライブラリ(原文ママ)そのものが単独で機能することはなく、あくまで他のプログラムへの部品となる。(略)」

(10) IT用語辞典バイナリ「オペレーティングシステムとは、コンピュータを制御し、アプリケーションソフトなどがコンピュータ資源を利用可能にするためのソフトウェアのことである。「OS」と略されることが多い。(略)」

範囲も、上記の協調動作する全てのプログラムにならなければならない。

(2) 反意図性の下位基準としての「使用説明書等に記載される」こと

ア 一般的な解釈論

ここで問題となっているのは、プログラムの機能が使用説明書等に記載されることによって、使用者がプログラムがどのような動作をするかを理解していることであるから(1. (2)参照)、プログラムの機能について理解している範囲の動作しかしないプログラムであれば反意図性はない。

そのためには、①プログラムの動作について正しく記載した説明の存在(プログラムに内在するか否かを問わない)、かつ②当該説明に対する一般の使用者からの到達可能性⁽¹¹⁾の2つが必要だと解される⁽¹²⁾。そうでないと当該プログラムの動作を正しく一般の使用者が理解することが困難になって、当該プログラムが一般の使用者の意図に反する動作をすると判断されることになり得るからである。ここで、①は、「使用説明書等」という概念の具体化であって、②については「記載」の概念を具体的に解釈しようとするものだと位置付けることが出来る。

イ プログラムの動作について正しく記載した説明(前記①)について

ここで問題となっている説明は使用者がプログラムの動作を理解するための要件であるから、その内容はもちろん、使用者にとって、それがプログラムの動作を正しく記載したものであることが分かるものであることが求められる。

ここでいう「正しく」とは、プログラムの客観的な動作と当該説明か

(11) ここでいう到達とは、意味内容への到達を意図しており、内容を知ることまで含めて到達と呼んでいる(西貝吉晃「コンピュータ・データへの無権限アクセスと刑事罰(六)」法協135巻7号1746頁以下(2018)で説明する「情報取得」の概念も参照)。その内容を知って使えるようになることが前提となっている議論だからである。

(12) この点については杉山=吉田が反意図性を欠くことの理由付けとして述べている「通常、使用者が認識し得るようになっている」という事情を踏まえれば、導くことが出来よう。

ら知りうる動作の間に齟齬がないということである。プログラムの開発が先行して、プログラムの開発者や開発した会社などプログラムの動作をよく知る者が、それを正しく説明する場合もあれば、プログラム開発のために作られた仕様書など、説明が先にある、プログラムがそれに違わないように作られる場合もある。ここで、プログラムが先に開発された場合でも、説明書を書く人物が開発者だけとは限らない。開発者以外の第三者がそのプログラムの動作を詳細に解析して、それについての報告書があったとすれば、それもまた、プログラムの動作を正しく記載した説明だといえる。

たとえば、医療機器に組み込まれているプログラムを考えてみよう。医療機器は、医薬品、医療機器等の品質、有効性及び安全性の確保等に関する法律(以下「薬機法」)などの法規制を受ける。また、一部の医療機器のプログラムは、許認可の仕組みがあり、PMDA(独立行政法人 医薬品医療機器総合機構)が承認する等の仕組みがある(薬機法第五章)。この際、機能、アルゴリズムの明確化、精度の評価などが審査項目となっており、必要な資料の提出が求められる⁽¹³⁾。この点、無事に許認可を受けた医療機器について、申請段階において提出された資料には、医療機器プログラムの動作について正しく記載された文章が含まれると考えられる。

その他、国際標準化機構(ISO)や国際電気標準会議(IEC)などの国際規格やFDA(アメリカ食品医薬品局)の医療機器認証、CEマーク(EU)など国外の制度も存在し、現にこれらの許認可を取得した場合には、同様の申請書、基準や指令などがプログラムの動作についてさまざまな説明をすることとなる。

なお、プログラムの動作を正しく記載した説明には、外国語で書かれた文献、数式を含む文献やプログラムのソースコード⁽¹⁴⁾など専門性の高

(13) 医療機器プログラムの承認申請に関するガイダンスの公表について(平成28年3月31日)(事務連絡)(各都道府県衛生主管部(局)薬務主管課あて厚生労働省医薬・生活衛生局医療機器・再生医療等製品担当参事官室通知)(URL: https://www.mhlw.go.jp/web/t_doc?dataId=00tc1785&dataType=1)

いものも存在する。

ウ 当該説明に対する一般の利用者からの到達可能性(上記②)について
イで述べた説明が仮にあったとしても、一般の利用者が探し出せなかったとすれば、その説明を知るための前提がないことになる。そうした説明まで使用説明書に入れて判断してはならない。つまり、立法担当者の見解もふまえると、その事案における、プログラムの名称、添付されているファイルや公開されているサイト上での説明などから見つけ出すことができる説明である必要がある。

ただし、(1)で論じたように、一般に、プログラムは、ライブラリに依存していたり別のプログラムを呼び出したりする。あるシステムの動作を理解する上で、1つの使用説明書等をみれば全てを理解できる場合もあれば、そうではなく、いわばモジュール毎に使用説明書が存在することによって、システム全体の動作を理解するためには、各モジュールの使用説明を全て読む必要がある場合もある。現実には、使用の過程において、様々なモジュールが使われていることを一般の利用者が認識することによって、各モジュールの説明を連鎖的に辿って行くこともあり得る。

たとえば、あるウェブサイトが第三者の決済サービスを使って決済をしている場合を考える。そして、そのウェブサイト上でその決済サービスを使っていることを表示していて、決済サービスのライブラリの動作として決済サービスに対して決済情報が送られているとする。また、この動作が決済サービスのプライバシーポリシー等書かれている状況を考える。そうした場合、プライバシーポリシー等にあるライブラリの開発元による説明が「プログラムの動作について正しく記載した説明」であった場合、決済サービスを使っている旨の表示からその正しく記載した説明まで到達できれば、反意図性が否定され得ることとなる。

反意図性は規範的に判断されると解されている以上、その下位基準も

(14) プログラムのソースコードは、人が機械に命令を伝えるものであると同時に、人同士の意思伝達の手段である。それゆえ、ソースコードの可読性が重視されて、背景事情など読み取りにくい内容についてはコメントが書かれる。

また規範的に判断されることになる。よって、ここで述べている到達が認められる範囲については、ある程度、総合考慮の結果として行われざるを得ないことになる。ただ、現在のプログラム開発では、大企業で数万人の技術者のフルタイム労働の共同による大規模な開発によってライブラリが作られることがある一方、アマチュアの個人が余暇を見つけてオペレーティングシステムの根幹やアプリケーションを開発していることもある。さらに、そういったプログラム群が協調動作をすることも一般的である。たとえば、現在、様々な企業が、複雑な機能を簡単なプログラムコードで組み込むことができるようなサービスを提供している。決済機能や本人認証といった機能は自力で作るにはとても複雑であるが、こういったサービスにおいては技術的に組み込みやすいことが求められる、数行のプログラムコードでこれらの機能が使えるようになっていくことも少なくない。しかしながら、その機能を呼び出す者にその動作を詳細にマニュアル化することを義務付ければ、本質的に複雑な動作を記述することになる上、提供されているサービスがアップデートされるたびにマニュアルを更新するなどの必要性が生まれることとなるが、それは非常にコストがかかる⁽¹⁵⁾。

すなわち、規範的判断において到達可能性を厳格にとれば、反意図性が認められやすくなり、それに対する対応が求められる。なぜなら、到達できない説明に記載された挙動については一般使用者の認識外になるからである。特に中小企業(SME、Small and Medium-sized Enterprise)の開発の場合に、これは大きな問題となる。というのは、SMEによる開発は、多くの場合、大企業や個人等の作成した多くのライブラリに依存しているので、SMEに他社の数万人の手による仕事や個人の仕事をいくつも詳細に解説することを義務付ける結果となる。反意図性の要件が過度な要求の作為を求めることになりかねず、開発者側にある種の説明責任として過大な負担を負わせることになりかねないからである⁽¹⁶⁾。また、ライブラリを開発しているのが海外の企業である場合や個人である

(15) ウェブサービスのアップデートの頻度は速いものでは週に複数回行われるものも存在する。

場合は、外国語で書かれた説明、数式やプログラムのソースコードなど専門性の要求される説明しか用意されていないことが想定される。技術者が日常的に読み書きしている、このような専門性の要求される説明を認めないという立場を取ると、本罪の存在により、SMEは、これらの内容を一般の利用者が理解できるようにするため、翻訳や数理的な知識の解説等に多大なコストを払い続けることになる。したがって、ウェブサイトが第三者の決済サービスを使って決済をしている場合に戻れば、サイト上での説明などから決済サービスのサービス名や運営会社等が分かった場合には、少なくともその決済サービスの公開されているサイトや運営会社のサイトまでは、到達可能と評価すべきである。ましてや、国際規格の認証を取得していることを公表している場合には、規格⁽¹⁷⁾が要求している動作を到達可能としてよいであろう。

一方、その逆に、立法担当者も具体的な被害者を想定しつつ、これを欺罔するような事案について類似の説明をしているが⁽¹⁸⁾、広く用いられているプログラムであったとしても、そのプログラムが動いている事自体を秘匿する、またはその名称を変更するなどの方法により、動作について正しく記載した説明へのアクセスが一切ない状況になったとすると、反意図性が肯定される。たとえば、Aがカフェでノートパソコンを開いて作業しており、Aがしばらく席を外したとする。その間にXが外部記憶装置をノートパソコンに接続して、外部記憶装置の内部に入っていたプログラムを実行したとする。この場合、具体的な使用者である

(16) さらにいえば、開発チームが国境をまたぐことが多々あり、その場合、日本のみ、総合考慮の結果として決まる、不正指令電磁的記録に関する罪の条文の文言からは一義的に読み取り難い、高度な義務を課することとなる。その場合、他国のエンジニアが開発した内容も含めて、一般ユーザの到達可能性を担保する説明書類の整理を担当する負担を事実上、日本のエンジニアが負わされる可能性が高い。日本にエンジニアを置かなければ生じなかったであろうコストが発生したと考える企業も存在し得、特に、そう考える多国籍企業はそもそも開発拠点を日本にも置かなくなってしまう、これは日本の競争力を害することにもなりかねない。

(17) IT用語辞典バイナリ「規格とは、主に産業や技術の分野において、製品や材料、あるいは工程などに関して定義された基準のことである。(略)」

Aに対応する一般の利用者は、どのようなプログラムが実行されるかについて、名称やその説明を受けていないわけであるから、反意図性が肯定され得ると考えられる。

エ フリーソフトを用いた基準の普遍性の検証

既にウまでで述べたように、①プログラムの動作について正しく記載した説明の存在、かつ②当該説明に対する一般の利用者からの到達可能性がある場合には反意図性が欠ける。ここで、杉山=吉田は、「市販されているソフトウェア」を例示しつつ、反意図性が欠ける場合の例として「使用説明書等に記載されるなど」と書いている。対して、「フリーソフト」の場合について「使用説明書が付されていないものもあり得る」と述べていることからすると、使用説明書を製品に同梱されているものとしてイメージしている可能性がある。もっとも、杉山=吉田も、使用説明書「等」と述べており、同梱されていない使用の説明、つまりマニュアルについても同様の議論が妥当することを前提としているといえるから、使用説明書の概念についての厳格な対立に意味はないと思われる。

その上で、杉山=吉田は、「当該ソフトウェアの機能は、その名称や公開されているサイト上での説明等により、通常、利用者が認識し得るようになってきている」ことを理由として反意図性が欠け得るとしている。しかし、電磁的記録に係る記録媒体でパッケージソフトが販売されている常態はなく、むしろウェブストアで販売され、ウェブ上で使用方法の説明がなされることが多いことを考えると⁽¹⁹⁾、現時のペーパーレス社会

(18) 杉山=吉田73-74頁は次のように説明する。「例えば、ハードディスク内のファイルを全て消去するプログラムが、その機能を適切に説明した上で公開されるなどしており、ハードディスク内のファイルを全て消去するという動作が利用者の「意図に反する」ものでないのであれば、処罰対象とはならない。他方、そのプログラムを、行政機関からの通知文書であるかのように装って、その旨の虚偽の説明を付すとともに、アイコンも偽装するなどして、事情を知らない第三者に電子メールで送り付け、その旨を誤信させて実行させ、ハードディスク内のファイルを全て消去させたというような場合には、そのプログラムの動作は、利用者の「意図に反する」「不正な」ものに当たり、不正指令電磁的記録として処罰対象となり得ると考えられる。」

においては、使用説明書が同梱されているか否かといった問題は、フリーソフトか否かとは別に存在するものであり、単に使用方法の理解のための情報へのアクセスの容易性の問題に収斂されるものに過ぎない。

それゆえ、フリーソフトか否かで議論を分ける必要はなく、上記の①、②の基準を統一的なものと捉えてあてはめをすれば足りる。

3. コインハイブ事件へのあてはめ

(1) 不正指令電磁的記録性が問題となるプログラムとその動作の範囲

本件は①「ウェブブラウザ」、②「コインハイブのライブラリ」、③本件電磁的記録である「被告人の書いたスクリプト」(③のみが高裁の不正指令電磁的記録性を肯定した対象である「プログラムコード」)の3つの組み合わせが協調動作していた事案である。つまり、③は単体で動くものではない。それゆえ、被告人が書いたのは③のみであるにもかかわらず、プログラムの動作の適否を検証するためには、①から③の全てを統合したシステムを前提に考える必要がある。

(2) 到達可能性

ここまで論じてきたように、「プログラムの動作について正しく記載した説明」と「説明に対する一般の使用者からの到達可能性」についての要件があり、まずは、後者について論じる。

本件で問題となったコードの使用者(サイト訪問者)の視点から見ると、たしかに③について被告人がウェブページ上で説明した内容からは②が動作することは分らないが、もともと、ブラウザを立ち上げているのであるから、①が動作することは知っていたといえる。しかしながら、高裁判決は、③について被告人がウェブページ上で説明した内容のみをもって、ソフトウェアの動作を理解できるか否かを判断しており、これ

(19) 例えば、Amazonのウェブストアでソフトウェアのライセンスを購入可能である。また、任天堂SwitchやSony Playstation 5のアプリストアで、有料か無料かを問わず、アプリのライセンスを取得可能である。

は、①も並行的に動作していることを看過しているように見える。それでは、本来、使用者が知っていた①および③のプログラムについての、使用者の一般に認識すべき動作とは一体なんであるかが問題となる。

結論から述べれば、①および③は、異なる開発元によって作られたソフトウェアが協調動作しているのだから、協調動作ができるようにするために、その動作についての共通の前提となる説明が存在しなければならない。具体的には、ブラウザ各社の説明に加えて「RFC標準とW3C勧告」という2つの文書群がそれである。

これについては、歴史的な沿革からこれらの文書群がその地位を得たのであるから、その歴史的背景についての説明をする必要がある。

(3) インターネットとウェブとRFCとW3Cに関する事実

ア 規約、RFCとW3Cについての現状

インターネットは、世界中のネットワークを相互接続して作られた地球規模の情報通信網である。また、ウェブ(ワールドワイドウェブ、WWWともいう。)は、インターネット上の相互参照された文章からなるシステムである。インターネットとウェブは、日常用語では混同されることがあるが、要するに、ウェブは、ブラウザで開けるもののものであり、インターネットというのは、ウェブや電子メールなども含めて世界中で使えるネットワークサービスで使われているネットワークのことであると思っておいて大きな間違いはない。

インターネットとウェブでは、世界中にあるコンピュータが協調して動作している。協調動作のためには情報通信が必要であるが、それをするためには、通信のためのなんらかの約束事がなければ、お互いに意味を通じさせることができない。異なる約束事では、通信として意味がまったく変わってしまうので、双方がどのような約束として通信をしているかについて、なんらかの方法で合意を取ることが必要である。この約束事を規約(通信プロトコル、または単にプロトコル)と呼ぶ⁽²⁰⁾⁽²¹⁾。

インターネットでは、TCP/IPと呼ばれるプロトコル⁽²²⁾が使われてお

(20) 現在、高校の情報の学習指導要領の範囲内である。

平成30年告示 高等学校学習指導要領 第2章 各学科に共通する各教科
第10節 情報 第2款 各科目 第1情報I 2内容

〔4〕 情報通信ネットワークとデータの活用

情報通信ネットワークを介して流通するデータに着目し、情報通信ネットワークや情報システムにより提供されるサービスを活用し、問題を発見・解決する活動を通して、次の事項を身に付けることができるよう指導する。

ア 次のような知識及び技能を身に付けること。

- (ア) 情報通信ネットワークの仕組みや構成要素、プロトコルの役割及び情報セキュリティを確保するための方法や技術について理解すること。〔強調は筆者による〕

プロトコルの役割については次のような説明が教科書(坂村健ほか「改訂版 高等学校 情報の科学」文部科学省検定済教科書、高等学校 情報科用(数研出版、2016年検定)においてなされている。58頁「情報を伝達(やりとり)するとき、伝達の手順をきちんと決めておかないと、情報を送っても相手が理解できないことがある。また、どのように表現して伝達するか決めておかないと、情報を正しく読み取ることができない。

通信するときの必要な手順や、情報の表現と形式などに関するとり決めを、通信プロトコル(プロトコル)という。〕

「プロトコルとは、もともと「外交儀礼、議定書」という意味であり、通信では「あらかじめ決めておく規約、手順」という意味で用いている。〕

- (21) もちろん、通信について規約が必要であるということは、電子的な通信に限られるものではない。たとえば、手旗信号を考えてみよう。手旗信号は、世界的に用いられている通信方法で、手に旗を持った状態で人が一連のポーズを取ることで、通信をするものである。和文手旗信号についての規約は、運輸省の告示で、「日本船舶信号法(昭和二十七年十月一日 運輸省告示第三百二号)」の第三節において定められている。ここの規約に従えば、「イロハ」と通信したいときに、ある人物が特定のポーズを取ることになっている。そして、ここの規約に従って、その特定のポーズを読み取ることによって、「イロハ」と復号することができるという仕組みである。また、どのような文字を使うかは、仮名については、明治33年「小学校令施行規則」、漢字については、内閣告示の常用漢字表によって、定められている。このように、日本国内で使われるものであれば、日本の法体系によって、規約、いってみれば通信の仕方の約束事、が定められており、それで十分である。

り、歴史的経緯から、事実上の標準(デファクトスタンダード)⁽²³⁾として、IETFの管理するRFC⁽²⁴⁾がインターネットの標準化⁽²⁵⁾をしており、W3C⁽²⁶⁾が管理するW3C勧告がウェブの標準化をしている。

これから見ていくように、現在、世界的規模で使われている、イン

(22) 坂村・前掲注2059頁「インターネットは、コンピュータのハードウェアやソフトウェアが異なっても通信できる。これは、インターネットの通信プロトコルが、きめ細かく統一されているためである。インターネットで通信するときには、数多くのプロトコルが必要で、それらの処理を個別に行うとたいへん複雑になる。そのため、それらのプロトコルを階層に分けて構成し、効率的に処理・管理している。

インターネットのプロトコルは、TCP/IPプロトコルとよばれ(中略)」

赤堀侃司ほか「情報の科学」文部科学省検定済教科書、高等学校情報科用(東京書籍、2016年検定)31頁「TCP/IP(Transmission Control Protocol/Internet Protocol)：アメリカ合衆国国防総省の資金支援によって開発されたネットワークプロトコル。インターネットの標準的なプロトコルとなっている。」

(23) 山下博通「ネットワークシステム」文部科学省検定済教科書、高等学校情報科用(実教出版、2014年検定)20頁「事実上の標準のことをデファクトスタンダード(de facto standard)という。」

(24) 山下・前掲注2320頁「また、インターネットやLANのプロトコルには、TCP/IPを使用し、これが事実上の標準になっている。この各種プロトコルなどの仕様は、インターネットに関する技術の標準化を策定している組織であるIETFがRFCという文書で公開している。」

(25) IT用語辞典バイナリ「標準化とは、利害関係者間における利便性や意思疎通を目的として、物事や事柄を統一したり、単純化、秩序化したりすることであり、無秩序化や多様化、複雑化を防止する規格を定めることである。(略)」

(26) 坂村・前掲注20110頁「W3C：HTML・XMLなどのWWWに関する技術やガイドラインの標準化を進める団体。」

赤堀ほか・前掲注2234頁「インターネットで現在、最も利用されているのはWWWとよばれる分散情報システムである。このシステムは、インターネット上の資料管理を目的として開発されたもので、W3Cにより、HTMLなどの仕様が定められている。」「W3C(World Wide Web Consortium)：WWWで利用される技術の標準化を行う団体。」

ターネットやウェブは、もともとは研究機関程度の小さな組織からはじまり、国を超えて使われるようになったものである。その拡大の過程では、規約は、さまざまな改良が加えられながらも、インターネットやウェブ全体で、共通の規約が使われていることで発展してきた。イでは、どのようにプロトコルが統一され、変更されてきたかについて見ていきたい。

イ インターネットとウェブの歴史と規約の合意方法

(ア) インターネットの歴史

1970年頃まで、さまざまな大学や研究機関においてコンピュータをつないだネットワークが多く作られてきた。なかでも特に重要なネットワークが、1969年に作られたARPANETで、UCLA、スタンフォード研究所、UCサンタバーバラとユタ大学の間をつないだものである。しかし、ネットワークによって通信の規約はばらばらで、ネットワーク同士を相互に接続することができなかった。

そのような中で、複数のネットワークをまたいだ通信についての研究が進み、1974年に、Vinton G. Cerf, Robert E. Kahn, “A Protocol for Packet Network Intercommunication”, IEEE Transactions on Communications, Vol. 22, No. 5, May 1974 pp. 637-648という論文がIEEEに出た。

複数のネットワークをまたいだ通信のアイデアは、GATEWAYと呼ばれるネットワーク同士の境界にいる計算機を用意して、ネットワークをまたいだ通信をするときは、GATEWAYがリレー方式で情報を伝えるというものである。

そして、同年12月にプロトコル(規約)がRFC 675, “Specification of Internet Transmission Control Protocol”という文書にまとめられ、1977年に、この規約を用いてARPANET, PRNET, SATNETという3つのネットワークがつながり、これがインターネットのはじまりとされている。

ここで、注目したいのは、規約を記述するための文書が、RFCで公開されたということである。このRFC 675というのはRFCと呼ばれる文書群の675番目の文書ということである。

RFCと呼ばれる文書群は、もともとはARPANETについてのものがあった。1968年の夏にARPANETのノードが置かれることが予定され

ていた4つの大学や研究機関の大学院生や教職員が集まり、Network Working Groupというグループができて会議が開催された。その集まりの中で合意を取るために作られた文書群がRFCである。このようにして、RFCによってインターネットに関する技術は標準化されていくという伝統が作られていった⁽²⁷⁾。

ARPANETは、Advanced Research Projects Agency NETwork、高等研究計画局ネットワークの略であり、アメリカ国防総省の高等研究計画局(ARPA)による資金が入っていたものの、大学と研究機関によるものであるから、その規約が法的にエンフォースされているものではない。

その後も、RFCは刊行を続け、現在は9,000を超える。その中には、当初の規約を置き換えるものや一部改定するものも含まれている。現在主流のTCP/IPプロトコルは、RFC 791, “Internet Protocol”のIPv 4と、RFC 793, “Transmission Control Protocol”等で定義されている。

その後、Network Working Groupは非営利組織インターネットソサエティーの下部組織IETF⁽²⁸⁾(Internet Engineering Task Force)に発展し、RFCの管理団体もIETFに変わった。また、インターネットが重要になるにつれて、インターネットの標準となるための手続きである、インターネット標準化過程がRFC 2026, “The Internet Standards Process—Revision 3”によって定められた。インターネット標準化過程は、proposed standard(標準化への提唱)、draft standard(標準化への草稿)、

(27) ここまでのRFCの歴史は、RFC 2555, “30 Years of RFCs”を参照。

(28) IETF自身によるIETFの説明は次のとおり。(https://www.ietf.org/about/) “The Internet Engineering Task Force (IETF) is the premier Internet standards body, developing open standards through open processes. The IETF is a large open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet. The technical work of the IETF is done in Working Groups, which are organized by topic into several Areas. Much of the work is handled via mailing lists. The IETF holds meetings three times per year”.

standard(標準)の段階を踏むというルールになっている。そして、RFCのうちstandardとして確立したものは、インターネットの標準⁽²⁹⁾であるとされている。ただし、いまでも、法的なエンフォースメントはなく、事実上の標準(デファクトスタンダード)である。

(イ) ウェブの歴史

1989年、欧州原子力機構(CERN)で働く技術者Tim Berners-LeeがCERN内の情報管理のための“Information Management: A Proposal”⁽³⁰⁾というウェブの始まりとなる提案書を書いた。提案書の内容は、文書に特別な記号を加えることで見出しや箇条書きなどが表現できるようにし、さらに、文書から参照している他の文書に飛ぶことができるリンクという機能を用意して、インターネット上に公開するというものであった。これによって、提案書は、まずは、CERN内部の情報を整理しようとしていたが、最終的には、世界中の情報がリンクの形でつながることを目指していた。そして、その後、提案書に従い、端末上で閲覧するためのソフトであるブラウザと文書を保管し送付するウェブサーバが開発された⁽³¹⁾。

そして、Tim Berners-Leeは、1994年にWorld Wide Web Consortium(W3C)を設立し、この団体がWebの標準を決めていくことになる。

ウェブに関してももちろん標準が必要である。たとえば、どのように文書に特別な記号を加えるかを定めたHTML(Hypertext Markup Language)の標準と、どのようにHTMLがインターネット上で送信されるかのHTTP(Hypertext Transfer Protocol)の規約は、それぞれRFCとなっており、1995年のRFC 1866, “Hypertext Markup Language - 2.0”と、1996年のRFC 1945, “Hypertext Transfer Protocol - HTTP/1.0”、1999年、RFC 2616, “Hypertext Transfer Protocol - HTTP/1.1”である。

(29) インターネットに関わるすべての規約(Protocols)・手順(Procedures)・取り決め(Conventions)についての標準である。

(30) <https://www.w3.org/History/1989/proposal.html>

(31) はじめのHTMLブラウザは、画像や動画をダウンロードすることはできたが、文章内部に埋め込んで表示することはできず、また、文字の色を変えることもできなかったようである。

時を前後して、1993年に、米国立スーパーコンピュータ応用研究所(NCSA)で、Marc Lowell Andreessenが、Mosaicブラウザを開発し、既存のHTMLを独自に拡張し、画像も扱えるブラウザを作った。その後、Marc Lowell Andreessenは、Netscape Communications Corporationという会社を立ち上げ、Netscape Navigatorという別のブラウザの開発をはじめた⁽³²⁾。1995年に登場したNetscape Navigator 2.0は、JavaScriptというスクリプト言語を加えるという独自の拡張がなされた⁽³³⁾。このときに、今までなかった「ウェブページにアクセスしただけで、ウェブページの設置者が書いたプログラムが動く」ということが起きるようになった。

(ウ) W3Cの現在のメンバー

W3Cには、Webにおいて重要なプレイヤーたちが参加している⁽³⁴⁾。主要なブラウザを作っている各社(Microsoft, Google, Apple, Mozilla Foundation, Alibaba)、ウェブサービス等のIT各社(Facebook, Amazon, Baidu, Tencent, Huawei, Xiaomi, AT&T, IBM, Intel, Cisco, Adobe, Wikimedia Foundation, Ethereum Foundation等)、メディア(BBC, NY times, 欧州放送連合)、クレジットカード会社(VISA, Master Card, American Express)、世界各国の政府機関、研究機関に加え、日本からもNTT、NHKや民放連、メーカー(日立、ソニー、NEC、パナソニック、富士通、三菱電機、東芝)、日本電子書籍出版社協会、出版社(講談社、集英社、角川)やIT企業(楽天、ソフトバンク、Yahoo Japan、Line)などが参加している。

(エ) W3C勧告とその位置づけ

W3Cは、ウェブの標準化団体であり、RFCとは別に独自のウェブの標準化のための技術文書を管理している。

W3CにもRFCのように標準化の段階があり、W3Cの技術文書は標準

(32) <https://www.w3.org/History.html>

(33) <https://developer.mozilla.org/ja/docs/Glossary/JavaScript> 「Brendan Eichによりサーバーサイド用言語として考案され、それからすぐの1995年9月に、JavaScriptはNetscape Navigator 2.0に搭載されました。」

(34) W3Cのメンバーのリスト。<https://www.w3.org/Consortium/Member/List>

化するまでに、作業草稿(Working Draft, WD)、勧告候補(Candidate Recommendation, CR)、勧告案(Proposed Recommendation, PR)、W3C勧告(W3C Recommendation)の段階を踏むことになっている⁽³⁵⁾。やはり事実上の標準であり、法的エンフォースメントがあるわけではない。もっとも、作業草稿段階であっても、すでにW3C内の作業部会(Working Group)において議論がなされ、公開することと今後の議論の基礎とすることについて部会内で採択され、ディレクター(W3Cの長)の承認を得ているものである。未解決の問題点や合意の取れていない箇所については言及することになっており、また、主要なブラウザを作っている開発元がすでに持っている機能を共通化するために提案していることが多く、草稿段階でもすでに一定の合意を前提とした内容である⁽³⁶⁾。

ウ 標準とブラウザ開発者の関係

ウェブに関係する開発者であるブラウザを作っている人たちおよびウェブサイトを作っている人たちは、RFC標準やW3C勧告といった標

(35) <https://www.w3.org/2020/Process-20200915/>

(36) W3Cが唯一のウェブ技術に関する団体というわけではない。たとえば、WHATWGという団体があり、絶えず変化していくHTMLの標準であるHTML Living Standardを策定している。この団体は徐々に影響力を強めた。2017年12月に、主要なブラウザ開発元であるApple, Google, Microsoft, Mozillaが、知的財産の指針と統治体制を整備した(<https://whatwg.org/faq>)。2019年5月以降は、HTMLとDOMについてはW3Cは独自の仕様策定をやめ、WHATWGの検討素案がW3C勧告となることが決まった。2021年1月に初めて実際に、W3C勧告として、検討素案であるLiving Standardのスナップショット(ある瞬間に切り取った複製)が承認された(<https://www.w3.org/blog/2021/01/whatwg-review-drafts-of-html-and-dom-endorsed-as-w3c-recommendations/>)。すなわちWHATWGの標準はW3Cの草稿に準じるものであると評価できよう。

以下、IT用語辞典バイナリから「WHATWGとは、Webの発展に興味を持つ人たちにより結成された、HTMLとWebアプリケーションに必要なAPIの開発に取り組んでいるコミュニティのことである。2004年、既存団体であるW3C(World Wide Web Consortium)が考えているXHTMLについての方向性、Webサイト構築現場のニーズへの対応に対して不満を表明した、Apple, Mozilla, Operaに所属するメンバーにより立ち上げられた。(略)」

準に従っているのであるが、RFC標準やW3C勧告といった標準は、一方向的にウェブの開発者を拘束するものではない。

むしろ、逆に、ウェブの開発者が、RFC標準やW3C勧告を事実上作っていくこともある。たとえば、Mosaicブラウザが、独自拡張として画像を表示する機能をつけた後に、RFC 1866が、その画像の表示の仕方をHTML 2.0として追認している。このように、一部のブラウザが独自の拡張として機能を採用した後に、その他のブラウザがそれを採用して、その後、標準になんらかの形で取り込まれるというのもきわめてよくある現象である。

エ W3Cによる制限

ウェブのセキュリティは、技術者の間で非常に大事な問題と扱われている⁽³⁷⁾。このため、ウェブブラウザ内で走るプログラムについて、W3Cは様々な制限を推奨している。Firefoxを開発するMozilla Foundationは、W3C勧告等の内容を整理して、数十項目を、コンテンツセキュリティ⁽³⁸⁾、コネクションセキュリティ、データセキュリティ、情報漏洩、完全性、クリックジャックからの保護、ユーザー情報セキュリティに分類して整理している。JavaScriptは、ウェブサイトにはアクセスしただけで実行され、信用できる人物が書いているとは限らないので、どのような動作をしたとしてもセキュリティ上やプライバシー上問題がなくなるようにつくられた環境の中で動いているということである⁽³⁹⁾。

オ ブラウザの独自の制限と機能

W3Cの制限に加えて、さらに安全に使えるようにブラウザには独自の制限がある。たとえば、Mozillaでは、ポップアップ広告に使われるポップアップウィンドウの大きさは、縦横100ピクセル未満になってはいけないことになっている⁽⁴⁰⁾。

一方、JavaScriptの機能が非常に広く使われるようになると、既存の仕組みでは禁止されている処理も特定の場合にのみ許可したいことがあ

(37) ブラウザ開発元の一つであるMozilla Foundationのウェブページである <https://developer.mozilla.org/ja/docs/Web/Security>に詳しい。

(38) 現在CSPレベル2が勧告<https://www.w3.org/TR/CSP2/>、CSPレベル3は現在は草稿段階である。<https://www.w3.org/TR/CSP3/>

るようになる。まず問題となったのが位置情報である。

2009年頃から、ブラウザの独自実装として、ブラウザによってはJavaScriptでの処理の際に位置情報(Geolocation)を使えるようになった。しかし、使えるようになるといっても、アクセスしたどのサイトでもユーザの位置情報が分かってしまうことはプライバシーの問題となるので使用者から許可を取る仕組みも同時にできた⁽⁴¹⁾。そして、主要なブラウザがその機能を採用し、2013年からW3C勧告となっている⁽⁴²⁾。

その後、ブラウザが通知(Notification)を出す機能(Webプッシュ通知やブラウザ通知などとも呼ばれ、そのウェブサイトを開いていなかったとしても、新聞社などのウェブサイトの更新やSNSなどでメッセージが来た際などにその旨のお知らせをポップアップとして画面に表示する機能)が作り出さ

(39) 一例を挙げれば、あるウェブサイトからダウンロードされたプログラムが他のウェブサイトなどのデータにアクセスする制限の仕方を定めたsame origin policyに反した処理については、“If the origin of url is not same origin with the XMLHttpRequest origin the user agent SHOULD raise a SECURITY_ERR exception and terminate these steps.”とあり、処理を中断(terminate)することが求められている。なお、SHOULDとは、RFC 2119 “Key words for use in RFCs to Indicate Requirement Levels”に定義された用語である。IPA/ISEC(独立行政法人 情報処理推進機構 セキュリティセンター)による、RFC 2119非公式訳は「「する必要がある(SHOULD)」この語句もしくは「推奨される(RECOMMENDED)」という形容表現は、特定の状況下では、特定の項目を無視する正当な理由が存在するかもしれませんが、異なる選択をする前に、当該項目の示唆するところを十分に理解し、慎重に重要性を判断しなければならない、ということを意味します。」としている。

(40) ユーザが気が付かないようにとても小さいウィンドウを開くことを禁止する趣旨である。ブラウザの独自の制限が記述された開発者向けのドキュメントとして、次がある。

<https://developer.mozilla.org/ja/docs/Web/API/Window/open>

(41) <https://developer.mozilla.org/ja/docs/Web/API/Geolocation> <https://www.w3.org/TR/geolocation-API/>

(42) 位置情報についての一番はじめのW3C草稿は2008年12月に書かれている。
<https://www.w3.org/TR/2008/WD-geolocation-API-20081222/>

れ、それについて許可を取る仕組みができた。(ただし、いまでもIEとSafari on iOSなどでは通知を出す仕組み自体を採用していない。)⁽⁴³⁾

カ 許可

W3Cでは、こういった許可を得て実行されるべき処理が増えてくると、どのようなJavaScriptの処理に許可が必要なものかが議論をはじめ、どのような処理ならば、どのような許可を取るべきかも決められていくことになった。

W3C Permissions APIは、現状、Geolocationなどのさまざまな許可のとり方が統一になっていないために、JavaScriptが統一的な許可の取り方をするようにしようという技術提案である。

現在、草案段階ではあるが、これを読めば、ブラウザの各開発者が、どのような処理において、許可が必要と考えているかが分かる。

その内容は、位置情報やカメラやジャイロスコープなどのセンサ、コピー&ペーストの際のクリップボード、通知⁽⁴⁴⁾であり、画面上にポップアップしてくる通知のようなものも、許諾をする仕組みを予定している。

ポップアップウィンドウという技術で、閲覧中のブラウザのウィンドウとは異なるウィンドウを開きそこに広告を表示するポップアップ広告のようなものについて立法担当者は許諾がなくても反意図性が欠け得るとしており、ポップアップウィンドウについてはW3Cの規格上も許可はいらぬ⁽⁴⁵⁾が、このW3C Permissions APIでは、ポップアップしてくる通知のようなものも、許可を取るべきであるとしている。つまり、W3Cのほうが具体的にブラウザについて厳しい基準を要求している。その上で、W3Cはクリップボードやカメラなど、許諾を得るようにする

(43) <https://developer.mozilla.org/ja/docs/Web/API/Notification/permission>

(44) 許可が必要な機能についてはhttps://developer.mozilla.org/ja/docs/Web/API/Permissions_APIと<https://www.w3.org/TR/permissions/#permission-registry>に限定列挙されており、“geolocation”, “notifications”, “push”, “midi”, “camera”, “microphone”, “speaker”, “device-info”, “background-fetch”, “background-sync”, “bluetooth”, “persistent-storage”, “ambient-light-sensor”, “accelerometer”, “gyroscope”, “magnetometer”, “clipboard-read”, “clipboard-write”, “display-capture”, “nfc”である。

仕組みを用意しており、かなり厳格に許可を求めるようにできている。それを考えるとW3Cはかなり厳格に、セキュリティだけでなく鬱陶しさなど含めて、安全に使えるようにしている。なお、このように、不正な計算ができないように注意深く作られた環境をサンドボックスと呼ぶことがある⁽⁴⁶⁾。

キ ブラウザ上のJavaScriptの特殊性について

ブラウザ上のJavaScriptは、ブラウザのJavaScriptを解釈するエンジンの中で動いており、ブラウザとJavaScriptがそろってはじめて動くものである。

JavaScriptを設置した人は、どのようなブラウザで動くかが分からないのであり、JavaScriptの動作は、どのようなブラウザの中で実行されているかによって、微妙な差異があるのだから、例外とはいえないブラウザの中で実行された結果としてどのように動作するかを認識する必要がある。そして、その処理が認められるかは、ブラウザがどのように振る舞うかを、包括的に考えることが必要となり、そのときに参考とする資料が、RFC標準やW3C勧告である。

(45) 現在、ポップアップウィンドウはブラウザの独自の制限によって、さまざまな制約を受けており、禁止される場合も多い。

例えば、Mozillaブラウザは、2001年9月リリースのバージョン0.9.4からページのロード時アンロード時のポップアップウィンドウを禁止する設定ができるようになった。(https://www-archive.mozilla.org/releases/mozilla0.9.4/)

また、アンロード時にポップアップウィンドウを開くことを禁止することは、WHATWGと最新のW3C勧告では標準とされている(https://github.com/whatwg/html/pull/3960、https://www.chromestatus.com/feature/5989473649164288#details)。

(46) IT用語辞典バイナリ「サンドボックスとは、もともとは砂場・砂箱といった意味であるが、IT用語としては、プログラムがシステムの他の部分に悪影響を及ぼすことのないように設計された環境のことである。

(略)セキュリティの分野では、外部で作られたソフトウェアが万が一不正なプログラムを含んでいても、システムに侵入したり不正操作したりされないように、プログラムを置く制限された場所をサンドボックスという。」

まず、標準化されているものは、標準に従って一般にブラウザが作られているのであるから、この文書を読めば、ブラウザの動作をよりよく理解することができるのであり、プログラムの動作について正しく記載した説明といえる。

また、草稿段階のものも、現在の各社が作っているブラウザの機能の共通化など現状の整理や追認のためのものが多く、また、主要ブラウザが実際に、草稿段階のものをどの程度採用しているかは、主要ブラウザの一つであるFirefoxを作っているMozilla Foundationのウェブページなどで一覧として閲覧することができる。

さらには、ウェブのエコシステムが、どのような考え方に基づいているかを窺うことができるものであるから、草稿段階のものに書かれていたとしても、プログラムの動作について正しく記載した説明といえることができる。ただし、主要ブラウザにおける採用が十分でないものは、その分、認識しうる範囲を割り引いて考えるべきものであろう。

そのことを前提として、ここまでの歴史を整理し直すと、次のようになる。JavaScriptは、①セキュリティやプライバシー上問題となるような処理を禁じられていて、それは、コンテンツセキュリティや完全性などの様々な制約によって実現されている。②使用者の許可を取るべき処理があるとされている。特定のセンサや位置情報などの使用の際には、使用者の許可を取るべきであり、その許可の取り方にW3C勧告の方法がある。通知を出すときには、使用者の許可を取るべきであり、その許可の取り方にW3C勧告の方法がある。その他、クリップボードを使うなど、いくつかの処理が限定列挙されていて、それらを使いたい場合は、使用者の許可を取るべきであるとされていて、主要ブラウザ(Google Chrome、Mozilla Firefox、Microsoft Edge等)では実装されているが、W3Cとしては草案の段階である。③それ以外の禁じられていない処理とそれらを組み合わせた処理は許されている。なお、バグの利用などで禁止されていない処理を組み合わせる場合、ないしは本来は必要でないのに必要であるかのよう装って、一定の許可をするように促すメッセージをエンドユーザに送り、それによって同人から得た許可を利用する場合等、「技術的に」禁止された処理と「法的に」同一と評価される

処理を行った場合もあり得ることは注意を要する⁽⁴⁷⁾。

(4) 上記の事柄の総合考慮

ア プログラムの動作について正しく記載した説明

RFCは、インターネットが開発される過程で、インターネットを含む技術を開発していた人々が、それらの技術と規約について合意を取るために作られた文書群であり、インターネットの規約を説明しているものである。そうであるから、インターネットに関係する開発をする場合RFCに従った開発がなされている。

また、W3Cは、ウェブの発明者であるTim Berners-Leeが、ウェブの標準化のために立ち上げた団体であり、Mozilla Foundation(Firefox)、Microsoft(Internet Explorer、Edge)、Google(Chrome)、Apple(Safari)といった主要なブラウザを作っている各団体はもちろん、各国の政府機関や研究機関、日本を含む各国のメーカーやメディアも参加している。ウェブに関する開発は、そのような経緯からW3Cによる標準化の内容に従って行われている。

ここまで論じてきたように、RFCとW3Cの標準は、ウェブに関するすべてのプログラム、すなわち、①「ウェブブラウザ」、②「コインハイブのライブラリ」、③本件電磁的記録である「被告人の書いたスクリプト」の各々、及びその協調動作を規律している事実上の標準である。それゆえ、これらの標準は「プログラムの動作について正確に記載した」と評価できる。

そうすると、まず、本件プログラムコードで実施されるマイニングという行為が、標準上、どのような取り扱いがなされているかが問題となる。ウェブの標準の内容から、(3)で論じてきたように、ブラウザ内での処理は「許可されない処理」「許可を取るべき処理」「許可なしにしてよい処理」に分けられている。ところで、②「コインハイブのライブラリ」、③本件電磁的記録である「被告人の書いたスクリプト」について考えて

(47) 本文で述べている「同一性の評価」については使用説明書の解釈方法論に踏み込むことになるので、別稿でさらに検討していきたい。

みると、マイニングのための処理は、完全性の侵害などの要素がないため許可されない処理ではなく、また限定列挙されている許可を取るべき処理を行っていないため、許可なしにしてよい処理に分類される。

イ 当該説明に対する一般の利用者からの到達

2. (2)イで論じてきたように到達可能性は、ある程度、総合考慮の結果として行われざるを得ない。そこでいくつかの要素を考えてみたい。

まず、RFC標準やW3C勧告は、ともに英語で書かれているが、その重要性から、IPA等が非公式の日本語訳を出している⁽⁴⁸⁾。

また、文章自体も簡単な図や少量の短いプログラムソースコードを例示のために含む場合がある以外は、きわめて平易な英語で書かれていてウェブで公開されている。

国際規格の認証は一般の利用者から到達可能と評価するべきところであるが、RFCやW3Cの標準は、高校の教科書にインターネットやウェブの標準である旨の説明がある。

よって、一般の利用者からの到達可能性についてもあるとあってよい。

ウ 結論

以上からすると、本件で問題となったプログラムコードの動作は、到達可能性のある文章に正確に記載されていた動作の範疇内の動作だったといえるから、当該動作について反意図性があったとは認められない⁽⁴⁹⁾⁽⁵⁰⁾。

(48) <https://www.ipa.go.jp/security/rfc/RFC.html> <https://www.w3.org/Consortium/Translation/Japanese等>

(49) なお、この議論はブラウザの特殊性からくるものであるから、その射程はRFCやW3Cの議論の及ぶ範囲内に限られる。カフェでノートパソコンで作業中に席を立ったときに、他人が勝手にUSBメモリ(外部記憶装置の例示)を挿入してソフトウェアを実行する事例で、USBメモリ内にあったのがバックグラウンドで動くような仮想通貨のマイニングツールであった場合、使用者に断りなく実行すれば、反意図性が肯定され得ると考える。また、ブラウザの外で動くプログラム、たとえばダウンロードしたゲームなどにマイニングツールが入っていたときにもここまでの議論は適用できず反意図性が肯定され得るであろう。

4. まとめ

本稿では、解説文言から読み取れる立法担当者の意図に沿いつつ、技術的な知見を基にして丁寧にこれを読み解くと、コインハイブ事件で使われたスクリプトは反意図性さえ否定されるのではないか、という試論を展開した⁽⁵¹⁾。すなわち、反意図性要件をプログラムの動作をいかにして一般の使用者に対して伝えるべきかについてのものだと考えたわけである。そうすると、本稿で論じてきたように一定の意味があることは確かである。しかしながら、一般の使用者についてプログラムの動作について理解すべきところが何であるかを真剣に論じるためには、コンピュータの歴史的発展や現在の仕組み全体について理解することが必要となる。それは、本稿自体があてはめで示したように、あまりにも解釈に当たっての調査のコストが大きすぎるように思われ、あらゆるソフトウェアで仕様書レベルの吟味を行うのは至難の業である。

それゆえ、別稿⁽⁵²⁾で議論しているが、一般論としては、処罰範囲の限

(50) それではごく一般的なソフトウェアである、WindowsやLinuxといったオペレーティングシステムであっても反意図性は肯定されうるのである。たとえば、USBメモリにインストールされたオペレーティングシステムは、次のような方法で、アクセス制限のかかったハードディスクのデータを盗み出すことに用いられることがある。コンピュータの起動時には、オペレーティングシステムが起動する前に、BIOSと呼ばれるプログラムが起動し、それが自己改変によってオペレーティングシステムを起動する。また、BIOSの設定は、ハードウェアへの物理的なアクセスによって書き換えることが可能である場合がある。これにより、BIOSの設定を書き換えてUSBメモリに入ったオペレーティングシステムを起動するようにすることができ、結果として、ハードディスクなどへのアクセス制限を回避できることになる。このため、ハードディスクのデータを窃取する目的で、USBメモリに一般的なオペレーティングシステムをインストールする行為は、不正指令電磁的記録に関する罪の客観的構成要件を満たしていると考えられる。よって、オペレーティングシステムといった一般的なソフトウェアも反意図性が肯定される。つまり、オペレーティングシステムの開発者について同罪の作成罪が適用される余地がある。

- (5) 実質論としても、ここまでの議論は肯定できるのではないだろうか。50年間のインターネットとウェブの歴史の中で、ブラウザの中でどのような処理が許されるのかということが、世界規模の議論の中で決められてきており、それがRFCやW3Cの標準として結実しているのであった。そして、その標準は、頻繁に更新されている。

この標準に不満があれば、ほとんどのブラウザでは、ブラウザがJavaScriptを実行しないようにブラウザの使用者が設定できる。

そして、ブラウザ開発者たちが、仮にコインハイブを禁じるべきであると考えれば、ブラウザの独自の制限としてそのような処理を禁じるように変えて、ブラウザをアップデートすることができる。実際にセキュリティ上重大な問題がある処理が予期せずにできたと考えられる場合には発見から2週間かからずに修正されることもある(<https://www.zdnet.com/article/google-patches-an-actively-exploited-chrome-zero-day/>)。逆にいえば、コインハイブのようなブラウザ内での仮想通貨の採掘は、標準を読み込んで日々開発をしているブラウザの開発者の間では、即座に禁止すべきようなものではなかったということである。

もっとも、すべてのブラウザ開発元がブラウザ内での仮想通貨の採掘を手放しで歓迎していたわけではない。GoogleのChromeブラウザでは、2017年9月からの議論(<https://bugs.chromium.org/p/chromium/issues/detail?id=766068>)がなされ、2020年9月にHeavy Ad Interventionsという機能が追加された。この機能は、CPUやネットワークを多く使用する広告を検出して停止するという目的のもので、広告の配信でよく使われるウェブページに別のウェブページを埋め込む技術であるiframeの内部での計算の場合、合計60秒以上相当のCPU使用、任意の30秒間でのCPU利用率50%以上、4Mバイト以上のネットワーク使用のどれかでその計算を停止させるというものである。これは、Chromeの独自の制限で、2021年6月時点のChromeチームの調査では、Firefox、Edge、Safariにこの機能を追隨して実装する予定はないとされている(Chrome Platform Status: Feature: Heavy Ad Intervention: <https://www.chromestatus.com/feature/4800491902992384>)。また、この機能を追加している理由について、2020年5月14日にChrome開発チームがBlog記事(<https://blog.chromium.org/2020/05/resource-heavy-ads-in-chrome.html>)を公開して説明している。要約すると「害を受けたり不愉快であったりするような経験をユーザがしないようにChromeを開発しており、優良広告基準(Better Ads Standards <https://www.betterads.org/standards/>) (筆者注：Googleも参加する業界団体である優良

広告連合の定めたインターネット広告の基準)を守り、大半の人が嫌がる広告に対処しようとしている。これまでにユーザを誤解させることを意図した広告からユーザを守るような変更を加えた。そして、今回、大量の計算資源を使う広告に対処することにした。調査の結果、0.30%の広告が広告によるCPUやネットワーク資源全体の消費のうちの30%近くを消費しており、そのような広告の一例として、仮想通貨の採掘をする広告やプログラムの書き方やネットワークの使い方が下手な広告が挙がる。この機能について、これから数ヶ月かけて実験をして、8月末頃に安定版の機能に加えるつもりである。時間をかけて導入をするのは、広告やプログラムの制作者に対策のために十分な時間を与えるためである。また、広告主は広告が止まった際に報告を受けることができるようにした。」ということである。

付言しておく、コインハイブ事件で問題となったプログラムは、おそらくiframe内で動いていなかったと思われるので、Heavy Ad Interventionsの制限を直接に受けるものではない。コインハイブのライブラリの公開が2017年7月であったことを考えると、3年以上経ってから、広告配信と仮想通貨の採掘を組み合わせた場合のChromeブラウザ独自の制限としてCPUやネットワークの使用量の制限が導入されたということとなる。しかも、その方法はあらかじめ変更を予告することで改善を促すものである。立法担当者が反意図性が欠けるとしたポップアップ広告が現在ほとんどブラウザの独自の制限として制限されていることとの均衡を考えると、ブラウザ内での仮想通貨の採掘の方がより穏当な扱いを受けている。

将来的に、ブラウザ中での仮想通貨の採掘の分類が、W3CのPermissions APIの中に加わり、許可がなくてはしてはいけない計算に変わる可能性がないとはいえない。しかしながら、現在のところはそのような合意や提案はない。この点、世界規模の合意によって作られてきた標準に反していないものに対して、日本法においては独自の判断から刑事的に違法の判断を与えること自体が直ちに好ましくないとまでは思わない。しかしながら、国際的な標準の変更は、当該標準の既存の体系を熟知している者たちの合議によってよりよい方向に修正が繰り返されてきている。そこに新たな軸を導入しようとする際には、既存の体系に不用意なメスをいれないように慎重さをもって、産業分野への弊害だけでなく、他の分野も含めて、総合的に違法判断の与える影響範囲を検討して行っていくべきものだと考える。

- 52) 西貝吉晃「不正指令電磁的記録に対する罪の解釈論」罪と罰58巻3号20頁(2021)。

定機能の観点から反意図性要件の意義及び機能に対しては、そもそも論として筆者等は疑問を感じてはいる。最終的には法改正も視野に入れた方が良いかもしれない。

本研究はJSPS科研費20K13356の助成を受けたものです。

謝辞

本稿については、技術を専門にされている下記のお二人の先生方から貴重なご意見を賜った。ご意見の内容を簡潔に記しつつ、感謝の意を申し上げます。

まず、平木敬東京大学名誉教授には、WEBの歴史等について一定のご教示を賜ったほか、本件については「反意図性の定義を厳密化することの意義は実は大きくないと考えています。他人の財産を無断で使うという意図は反意図性の有無とは独立だからです。」というご感想を頂いた。反意図性の限界点及び他罪との棲分けの在り方について重要な示唆を賜ったものと考えている。

また、登大遊先生(ソフティーサ株式会社代表取締役会長、筑波大学国際産学連携本部准教授)からは、本件は実際には利益窃盗の話であり、それを不正指令電磁的記録に関する罪で捕捉するのは濫用的な法適用なのではないか、この問題はひいては同罪の立法の在り方についての重大な疑問につながる、という重要な示唆を頂いた。