



2011年の審査基準では技術的な評価の比重が高くなっているため、この傾向が顕著であった。そのため、チームに1人は開発力に優れているメンバが必要となるだろう。また、SDIは研究発表ではないので、技術の先進性だけでなく、有用性、現実性、ビジネス性、訴求性などがバランス良く盛り込まれたシステムであることが重要である。そのようなシステムを作るためには、何らかのシステム開発プロジェクトに取り組んだことがあるメンバがチーム内にいると心強いだろう。これは敷居が高いように聞こえるかもしれないが、最近はWebサーバや開発ツールのオープン化、無償化が進んでいるため、個人で何らかのシステムを構築した経験を持っている学生は少なくないだろう。

開発力以外で特に重要なスキルだと個人的に考えているのはプレゼンテーション力である。プレゼンテーションとは、ここでは発表の能力だけでなく、論理構成・デモンストレーション・デザインなど、広義の意味で情報を伝える能力を指す。プレゼンテーションは評価項目にも入っているが、スコア的にはそれほど比重は高くないため、軽視してしまうチームがある。だが、理由は後述するが、実際はスコア以上に非常に重要なポイントであるため、軽視できない。

また、プレゼンテーションの成否を決定する要素は英語力である。世界大会では当然英語でプレゼンをしなければならないが、英語でうまくコミュニケーションできなければ、作ったものの良さを伝えることができない。チーム内で役割分担をすれば、台詞をすべて暗記するのはさほど難しくはないだろうが、問題は質疑応答である。世界大会に出場したある国のチームは全員で手分けして台詞をすべて覚えたが、質疑応答で審査員が何を言っているかが分からなく、伝えたいことが伝わらなかったという事例もある。そういう意味で、最低1人は今までに十分なプレゼンの経験を持っているメンバが必要で、かつその人物は英語が堪能であると心強い。英語圏もしくは英語に類似した言語圏の帰国子女か留学生を1人チームに入れることができれば理想であるが、

そうでない場合は英語プレゼンテーションとディスカッションを十分に訓練して挑む必要がある。

そして、最後に強調したいのがチームワークの重要性である。すべてのメンバに共通の事項だが、コミュニケーション力のない人や自分勝手な人はチーム開発に向かない。いくら個人の力があっても4人で1つのものを作るということを忘れてはならない。自分から積極的に「何をしないとイケないか」を考え、自分から行動するようなメンバを集める必要があるだろう。また、いくら技術力が高くても、忙しすぎる人もチームメンバとしては不相当だ。最低限でも週に数回ミーティングができる程度ではないとお互いに何をやっているかが見えなくなり、プロジェクトをうまく進めることは難しいだろう。通常のシステム開発では、毎日進捗を把握することもあるが、それはソフトウェア開発企業での話で、学生では毎日ミーティングの時間を作ることは難しいかもしれない。それでも、週数回のミーティングや、プロジェクトを推進するための最低限の時間も取れないメンバは、チームワークを期待するのは難しいだろう。

ちなみに、「賞金を分けるのがもったいないから少人数で出場しよう」と考えている学生がいたら、参加しないことをお勧めする。数々のスキルを必要とするImagine Cupでは、4人でも競争力のあるチームを作ることはなかなか難しい。そのような考えを持っている学生は、賞金を獲得することなく終わることになるだろう。Imagine Cupの価値は、賞金ではなく名誉や出会い、成長など金銭では得られない貴重な経験にある。ちなみに、筆者の場合は今までに日本大会で獲得した賞金はすべて世界大会の準備費用に使っている。2006年のインド大会では、必要物品(開発用デバイス・デモ用機材)や、ヒアリング・実地調査に必要な移動費用などで赤字になってしまった。そこまではしなくてもいいと思うが、他の国の学生も予選で得た資金を使って世界大会の準備をするというところは多いので、賞金を自分の懐に入れるところは大きな差が生まれるだろう。賞金を目当てにしているようなメンバは最初からチー

ムに入れないほうが良いと思うが、気心の知れている友人や仲間とチームを作る場合でも、お金の絡むところはシビアなので最初にしっかり決めておいたほうが、後々の問題を回避できるだろう。

## 企画編

何をテーマにして、どのような問題を解決する作品を開発するかは非常に重要で、企画の段階でほぼ勝負は決まると言っても過言ではない。いくら技術に新規性があっても、プレゼンのうまくても、企画のバランスが大事であり、何か1つ光るアイデアが入っている必要がある。ここでは、どんな作品がSDIに向いているのかについて考察する。これは毎年のテーマに依存するので、一概には言えないが、基本的に考慮しておくといいたいと考えている点がある。以下にポイントを列挙する。

- 分かりやすいもの
- デモンストレーション性の高いもの
- 現実性のあるもの
- ビジネス性の高いもの
- インパクトのあるもの
- 飛び道具(ハードウェア)を絡めるのが強い

以降、順に説明する。

### ● 分かりやすい作品

まず、分かりやすい作品が比較的利益であると言える。審査員はさまざまな背景(大学の教授であったり、技術者、管理職であったり)を持っているので、「こんなすごい技術を使っている」とアピールしても短い時間で伝わらない場合が多々ある。たとえば、複雑な背景から生まれた技術で、概念の説明に時間がかかってしまうようなものは、説明の時間が限られているオンライン競技では比較的不利になると言える。特に、世界大会においては英語が堪能でないチームであればさらに理解してもらえないのは難しいだろう。そのため、短い時間で理解しやすい作品は比較的利益になると思われる。

### ● デモンストレーション性の高い作品

基本的にはプレゼンテーション中にデモンストレーションをすることが望ましい。デモムービーだけでは、どこまで実際に実装できているか判断が難しいためである。そのため、リアルタイムにデモンストレーションのできる作品が比較的利益だ。このような理由から、デモンストレーションが可能で、作品の良さをデモによって力強く伝えることのできる、デモンストレーション性の高い作品が利益になると思われる。たとえば、視覚的な効果の高い作品などが該当する。このようなソフトでは、詳細な説明より、リアルタイムのデモンストレーションによって作品の良さを伝えることができるためである。たとえば、2011年で上位入賞したUSAチームは、視覚障害者用のノート作成支援システム(図-1)を開発した。これは、ズーム・視点移動などができるカメラとノートアプリケーションを組み合わせた作品だったが、競技中にOpenCVを利用した濃淡の切り替え機能や、タッチインタフェースで視点移動といったリアルタイムデモンストレーションを実施した。これらの視覚的に分かりやすいデモンストレーションは、数値を並べられるよりも説得力がある。一方、デモンストレーション性の低い作品の特徴とは何だろうか。たとえば、「塩基配列を超短時間で解析するアルゴリズム」を作ったとする。技術的新規性という面では素晴らしいことかもしれないが、「どこかの国際会議で受賞」「ダウンロードが1万件を超えた」などの客観的なデータがない限り、そのアルゴリズムの良さを審査員が短い時間の中で理解することは困難である。その良さが瞬時に視覚的に伝わらないためである。数値やグラフでその良さを説明することは可能だが、説明に時間がかかってしまう上に理解してもらえない危険性が高い。そのため、やはり基本は視覚的に訴えることのできる良さをを持った作品がSDIに向いていると言える。

### ● 現実性・ビジネス性の高い作品

研究成果を利用することができれば、一から開発



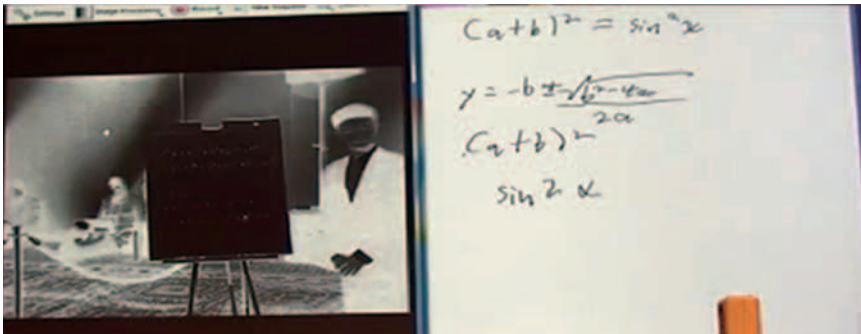
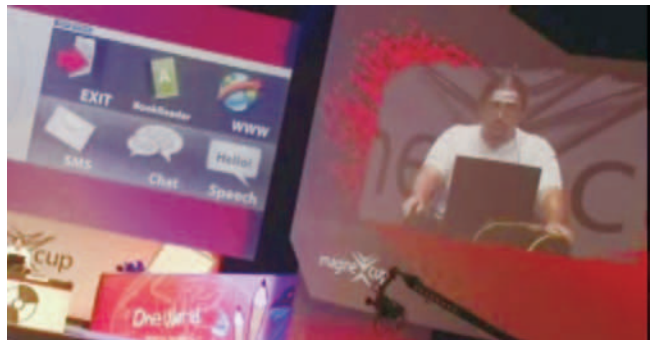


図-1 視覚障害者用のノート作成支援システム

図-2 脳波を利用した全身麻痺障害者のための入力インターフェース



するチームに比べて優位になるかもしれない。実際に、世界大会に出場するチームの中には、自分たちの研究室でのプロジェクトを進化させた作品が少ない。卒業研究や大学院での研究の一部を利用している場合も多い。ただ、これは日本では難しい部分が多々ある。研究とは「使いやすさ」が第一優先事項ではなく、「新規性」が第一優先事項だからである。分かりやすい例で言うと、学会などで高い評価を受けたものを実際に一般の人に見せると使いにくそうなものに見えてしまうことが多々ある。それとは逆に、実用的で多くの人が使っているソフトウェアなどは、学会などでは新規性という面から評価されにくい場合がある。そのため、研究内容(の一部)を利用して応募する際には、実用面を考え直す必要があるが、その点がクリアできれば、完成度と実用面を兼ね備えた強みになる可能性がある。たとえば、2010年に上位入賞したニュージーランドのチームは、ラジオ波を利用した教材の配信システムを提案したが、ラジオ波でデータ送信をする部分やCRC(巡回冗長検査)などの基盤技術は研究室の成果を応用したものだった。この作品は、基盤技術部分に研究室の成果をうまく利用している例だと言える。

## ● インパクトのある作品

インパクトのある作品とはどのようなものだろう

か？ これは定義が難しいが、個人的には「狭く深く」型のプロジェクトが大事だと考えている。その逆のパターンとして、よくある失敗が、「広く浅く」方のプロジェクトで、「あれもできます、これもできます」と、機能の多さを強調する作品だ。たとえば、毎年以下のような作品を世界大会で見る。

- **リッチクライアントを使った高機能学習支援システム**
- **最新のモバイル技術を利用した高機能ボランティア支援システム**
- **最新の Web 技術を利用した高機能募金システム**

このような作品は、問題の本質を深掘りしない傾向がある。多機能で使い勝手が良いことをアピールするような作品は、実用的に見える一方で、聞き手(審査員)は話の要点が見えなくなる。言い方を変えると、「解決しようとしている問題が不明瞭」になる、ということである。多機能をアピールするのではなく、問題を深く掘り下げ、対象を絞ることが重要だ。個人的には、いろいろな機能が充実している必要はなく、狭い領域でも良いので、深くインパクトを与える作品が強いのではないかと考えている。インパクトのあるものの例としては、2010年に上位入賞したセルビアチームの「脳波を利用した全身麻痺障害者のための入力インターフェース」(図-2)

などが挙げられる。これはBMI (Brain Machine Interface) の技術を利用した時限式の選択インタフェースを主技術とする作品だ。このシステムで恩恵を受けるユーザ数は少ないかもしれないが、問題を深く掘り下げており、利用者にとっては深いインパクトを与える。

### ● 飛び道具(ハードウェア)を絡める作品

過去の傾向を見ていると、上位に入賞する作品の多くは、何らかのハードウェアと連動している。具体的には、2010年、2011年にトップ6に入った作品はほぼすべてが何らかのデバイスを利用していた。以下に2010年と2011年の上位作品で利用されていたハードウェアの例を示す。

- Webカメラ(+ OpenCVによる画像処理)
- 脳波検知デバイス
- ラジオ波の受信, 送信デバイス(トランスミッタ)
- Wii コントローラ
- 加速度・ジャイロセンサ
- Kinect
- モバイル端末

競技カテゴリーが「ソフトウェア」デザイン部門であるため、当然ソフトウェアが主体となるべきなのだが、正直な話ソフトウェアだけの作品ではできないことの幅が決まっており、それほど新しいことができないことに起因すると考えられる。また、すべてソフトウェアだけで構成される作品は、短い審査時間では中身がしっかり開発されているか判断できない場合がある。そのため、何らかのハードウェアが絡むものがあれば実現できることの幅を広げ、他の作品と差別化ができると思われる。しかし、ハードウェアと連携したソフトウェアを開発するにはそれなりの経験を要するため、短期間では質の低いものしかできない上に、無理矢理使うと作品のバランスが崩れてしまう。そのような場合は、無理にハードウェアを組み込む必要はなく、ソフトウェア開発に全力をつぎ込むべきだろう。また、ソフトウェアだけでも十分に革新的であれば無理にハードウェアを組み込む必要はないため、プロジェクトの方向性とバラ

ンスを取ることが大事だ。

### 開発編

さて、企画が決まればいよいよ開発である。開発の段階で一番重要なのはスケジューリングだろう。1人で開発する場合と異なり、チームで開発する場合、「いつまでになにをやる」と決めたことを守りながら進めることが重要だ。開発したものをベースにプレゼンテーションやユーザテスト、ヒアリングなどを行うので、開発が遅延した場合、予定していたヒアリングが行えない、プレゼンテーションの流れを変更しなければならないなど、スケジュール全体に影響する事態が発生する。このような事態を避けるためには、開発途中の部分でも仲間と常に共有しながら進めるのが大事だ。チーム開発に慣れてない開発者は往々にして開発途中のものを共有することを嫌うが、チームで開発をしている場合、どこまで開発できているのかがお互いに見えないと、スケジュール全体を統括することが難しい。逆に情報が共有されていると、他のメンバからのサポートを受けられるなど多くのメリットがある。そのため、バージョン管理システムなどを導入して常にソースを共有するなどの方法が有効だろう。

また、プロジェクトの本質ではない部分の開発には時間をかける必要はない。本質ではない部分とは、たとえばユーザインタフェースの作り込みなどである。前編ではユーザインタフェースの作り込みは短い時間で効果的に作品の良さを伝えるためには重要なポイントだと記述したが、それは多くの作品において問題の本質的な部分ではないだろう。そのため、少ない労力でそれなりの効果を得られるような方法を取るべきだ。具体的にはユーザインタフェース部分などでは積極的にライブラリを使うのが良いだろう。Microsoft系技術に関しては、Code4FunやCodePlex、The Code ProjectといったWebサイトで、豊富にSilverlightなどのライブラリが共有されている。また、SourceForgeやGoogle Codeなどでも役に立つライブラリが共有されていることだ

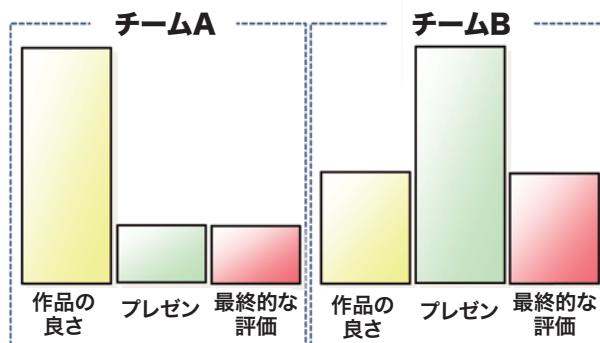


図-3 作品の良さを 100% 伝えるには

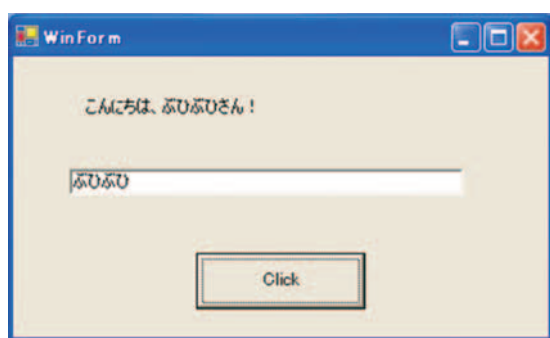


図-4 インタフェースデザインとユーザエクスペリエンス

ろう。これらの資産を有効に使い、自分たちは作品の本質的な部分の開発に注力するのが大事なポイントだ。

## プレゼンテーション編

### ● プレゼンテーションの重要性

プレゼンテーションは評価項目にも入っているが、全体の 10% 程度であり、スコア的にはそれほど比重は高くない。そのため、軽視してしまうチームがあるが、実際はスコア以上に重要なポイントであることは、競技を経験した者が口をそろえて言うことだ。図-3 を見てほしい。

仮に A チームと B チームが存在したとしよう。A チームは作品の良さがきわめて高いがプレゼンテーションが悪く、内容が 100% 伝わらなかった。その結果、最終的な評価を大きく落としている。一方、B チームは作品の良さはチーム A より低いにもかかわらず、プレゼンテーションがすばらしく、作品の良さが 100% 伝わり、その結果、チーム B のほ

うが最終的な評価で勝っている。つまり、プレゼンテーションが悪いと自分たちの作品の技術的な新規性や有用性などがうまく伝えられない可能性があり、プレゼンテーション以外の評価項目のスコアも低くなることもある。

インタフェースのデザインも同様である。極端な例だが、図-4 左を見てほしい。

「このシステムは高度な AI を備えており、数億の専門的な質問に対して瞬時に回答するアルゴリズムを実装している」と言われた場合、どう思うだろうか。本稿の読者の中には、「中身で勝負だ。中身が良ければ評価されるはず」と硬派な考えを持つ方もいるかもしれないが、人間は情報の大部分を視覚から得るため、このような要因で作品に対する印象が決まってしまうのは避けられない。特に、Imagine Cup のようにオンサイトでリアルタイムに行われるコンテストでは、その傾向が強い。図-4 右は過去にエントリーされた作品の例だが、多くのチームはインタフェースのデザインでも妥協していない。無用にインタフェースにこだわる必要はないが、リッ



クライアント技術を駆使して開発してくる作品が多い中で、極端に質素すぎるインタフェースでエンタリすると不利になる。

ここで伝えたいのは、自分たちの作品の良さを100% 伝えるためには、プレゼンテーションやインタフェースなどの点で損をしてはいけない、という点である。プレゼンテーションの良さで作品の悪さをカバーできると言っているわけではないので、見栄えを取り繕うことばかりに力を入れるような本末転倒な結果にならないように気をつけてほしい。

また、審査員が非常に優秀で多くのことを知っていると仮定して説明することや、「言わなくても伝わるだろう」と安易に考えることは危険である。これは、審査員の専門はバラバラで、限られた時間の中で作品を評価しないとイケないため、作品の深い部分まで掘り下げて評価することが難しいことに起因する。そのため、伝えたいことがあったとしても、プレゼンテーションや伝え方の問題によって伝わらなかった場合は、「なかったもの」として扱われるので、短い時間で作品の良さを効果的に伝えるには、プレゼンテーションや伝え方の工夫の点で妥協をしてはいけない。2011年のニューヨーク大会で印象的だったのは、上位に残った6チームはすべてプレゼンテーションやデザインの質が非常に高かったことである。プレゼンテーションやデモンストレーションの流れは緻密に設計されており、インタフェースのデザインなどの点でも凝っているものが多かった。

## ● プレゼンテーションのトレーニング

欧米人に比べて日本人はプレゼンテーションが下手だと言われている。欧州のある国では、幼稚園の子どもたちに自分の好きな玩具を持ってこさせて、「じゃあ、今からみんなの前で、そのおもちゃの良いところはどこなのか説明してください」と説明させる教育をしている。これは、小さい頃から論理的に説明する癖をつけるという訓練である。比べて日本はどうだろうか。小中高と受験のための勉強がメインでプレゼンテーション能力を鍛える訓練はほぼ

皆無だと言って良い。また、授業中に積極的に意見を言うのはどことなく気恥ずかしい雰囲気の中で育ってしまっている。これは日本の慎ましさ、という文化的側面を持っているかもしれないが、このような態度や感覚は世界規模のコンテストではまったく通用しない。アピールする部分はしっかり自信を持って論理的に力強くアピールすることが重要だ。

プレゼンテーションの研究も忘れてはならない。良いプレゼンテーションとは何か、どうすればもっと良いものになるのか、ということ常々気をつけて訓練する必要がある。プレゼンテーションのテクニックを解説していくと専門書数冊分の量になってしまうので、詳細は省くが、1つだけ重要な要点を挙げるとすると、優秀なプレゼンテーションを何度も聞くことではないかと思う。今はブロードバンドが普及したおかげで世界中の優秀な発表を無料で聞くことができる。たとえば、米Apple社のスティーブ・ジョブズ(Steven Jobs)やMicrosoft社のスティーブ・バルマー(Steven Ballmer)、TechCrunchのプレゼンテーションなどは参考になるだろう。特にTechCrunchは発表の長さ、流れ、内容などImagine Cupに通じる部分が多いため、どうやって観客を盛り上げているのか、彼らのプレゼンにあって自分のプレゼンにないものは何かなどを分析すると得られるもの多いだろう。筆者もImagine Cupに出場するにあたってプレゼンを作成するときは、スライドの構成やデザイン、話し方、言い回しなど、著名人のプレゼンテーションを何度も分析して得たノウハウを詰め込んだ。当然それらの動画はすべて英語だが、「リスニングの勉強もできて一石二鳥」と考えるとやる気も出るのではないだろうか。

そして最後に勝負を決めるのは練習量だ。時間が許す限り仲間相手でも先生相手でも誰を相手でもいいので、数多く練習するべきだ。練習した分だけ流暢にプレゼンテーションができるようになる上に、たくさん話すことで、プロジェクトに対して多くのフィードバックをもらえる。すべてのフィードバックに対応する必要はないが、よく指摘されるところなどをヒントにストーリーの構成を変更したり、デ

ータを追加したりといったことが可能になるだろう。この結果、さらに自分のプロジェクトに自信が持てるようになり、説得力が上がっていく。これは学会における研究発表にも通じる点ではないだろうか。

## まとめ

Imagine Cup は世界中で大きな盛り上がりを見せており、IT 系学生が参加する代表的なコンテストの1つに成長した。Imagine Cup はいくつかのカテゴリから構成されるが、その中でもソフトウェア部門は最も注目を集めるカテゴリであり、その分熾烈な競争が繰り広げられる。そこでは、世界の社会問題を解決する「アイディア」、作品の根底にある「技術」、実現するための「ビジネスモデル」、伝えるた

めの「プレゼンテーション」が総合的に評価される。それぞれの点で注意すべきポイントやノウハウが存在し、それらをしっかり押さえているかで勝負が分かれる場合もあるだろう。本稿は筆者の視点でまとめたものであるため、すべての情報がすべてのチームに適用できるとは思わないが、将来の日本チームが世界大会で上位を目指す上で少しでも役に立つ情報が本稿を通じて共有できれば幸いである。

(2011年10月4日受付)

中山 浩太郎 (正会員) nakayama@cks.u-tokyo.ac.jp

2007年大阪大学大学院情報科学研究科博士号取得。同年同研究科研究員を経て、2008年より東京大学知の構造化センター特任助教。2004～06年 Imagine Cup 日本代表。2009～11年 Imagine Cup 世界大会審査委員。

## リンク集

### ■ Imagine Cup 全般

- 公式レポート：世界大会レポート(2011)  
<http://www.microsoft.com/japan/academic/imaginecup/2011/wf/report.mspx>
- 公式レポート：日本大会レポート(2011)  
<http://www.microsoft.com/japan/academic/imaginecup/2011/local/report.mspx>
- 公式ビデオ：Message for Students – Imagine Cup 2011 World Final  
<http://msdn.microsoft.com/ja-jp/academic/video/hh529909/>
- 公式ビデオ：コンピュータの未来“2019年の世界”  
<http://msdn.microsoft.com/ja-jp/academic/video/hh546809/>
- 公式ビデオ：Imagine Cup 2011 ニューヨーク世界大会レポート  
<http://msdn.microsoft.com/ja-jp/academic/video/hh546746/>
- 公式：マイクロソフトアカデミックチームブログ  
[http://blogs.msdn.com/b/microsoft\\_japan\\_academic/](http://blogs.msdn.com/b/microsoft_japan_academic/)

### ■ 開発

- CodePlex  
<http://www.codeplex.com/>
- CodeProject  
<http://www.codeproject.com/>
- Coding4Fun  
<http://channel9.msdn.com/coding4fun/>
- GIZMODO  
<http://us.gizmodo.com>

### ■ プレゼンテーション・デザイン

- TechCrunch  
<http://techcrunch.com/>
- note&point  
<http://noteandpoint.com/>
- Quince  
<http://quince.infragistics.com/>
- Sociomedia | UI デザインパターン  
<https://www.sociomedia.co.jp/category/uidesignpatterns/>