

INDIGO

Zaawansowana niskokodowa platforma programistyczna [LCDP]
do integrowania maszyn i systemów IT oraz sterowania nimi
za pomocą graficznego modelowania procesów.

W jakim kierunku idziemy i dlaczego?

Poszukiwanie metody, która zapewni ludziom komfort porozumiewania się z maszynami w zrozumiały sposób, jest jednym z głównych kierunków rozwoju informatyki. Języki programowania stają się coraz bardziej zbliżone do języków naturalnych, co ułatwia i znacznie przyspiesza proces tworzenia niezawodnego oprogramowania.

Historia programowania rozpoczęła się od języka maszynowego, w którym komendy w postaci zer i jedynek (kod binarny) wydawane były wprost do procesora w urządzeniu. Oznaczało to, że dla każdego procesora trzeba było pisać osobne instrukcje. Następnie pojawiły się niskopoziomowe języki programowania, w których można było wykonać pewne operacje za pomocą oznaczeń symbolicznych. Ten trend umacniał się w kolejnych językach, takich jak C i C++, które coraz bardziej odchodziły od kodu maszynowego.

Były one dużo bardziej uniwersalne i umożliwiły rozwiązanie szerokiego zakresu problemów. Dodatkowo języki te – z których część jest używana i rozwijana do dziś – pozwoliły na znaczne skrócenie czasu tworzenia aplikacji i na zmniejszenie liczby błędów dzięki automatyzacji zadań wcześniej wykonywanych przez programistów. Przede wszystkim jednak pozwoliły na oddzielenie oprogramowania od warstwy sprzętowej, co umożliwiło jego przenośność, czyli możliwość wykorzystania tego samego kodu na różnych platformach sprzętowych lub w różnych systemach operacyjnych.

My zrobiliśmy kolejny krok naprzód i stworzyliśmy framework INDIGO – platformę do integracji maszyn i systemów IT oraz do sterowania nimi za pomocą grafów. INDIGO nie tylko może być przenoszone na różne platformy, ale co ważniejsze – pozwala łączyć ze sobą dowolne maszyny, systemy IT, sterowniki PLC czy różnego rodzaju czujniki w taki sposób, aby poszczególne węzły mogły nie tylko swobodnie wymieniać między sobą informacje, ale także kontrolować wzajemnie swoje zachowanie.

Co więcej, programowanie w INDIGO polega na modelowaniu poszczególnych procesów na grafach w notacji BPMN 2.0. Co z tego wynika? O ile w tradycyjnych językach programowania najmniejszy samodzielny fragment kodu jest w gruncie rzeczy listą prostych instrukcji, które tak naprawdę niewiele zmieniły się od czasów maszyny Turinga, o tyle w naszym przypadku jest to gotowy element jakiegoś procesu, który pełni konkretną funkcję, np. „wyświetl okno z komunikatem” czy też „wyślij komendę do urządzenia”.

Aby zaprogramować cały proces w INDIGO, wystarczy więc zamodelować jego przebieg na grafie. Poszczególne komponenty grafu mogą być poleceniami dla serwera do zapisu lub odczytu danych lub wykonania konkretnej funkcji, poleceniami dla urządzeń do wykonania zdefiniowanych wcześniej akcji, mogą też inicjować dwukierunkową komunikację z systemami IT.

Tworząc oprogramowanie w INDIGO, nie myślimy więc o samym kodzie źródłowym jak w przypadku programowania w tradycyjnych językach, ale o jego funkcjach, które są wywoływane przez poszczególne elementy grafu.

Grafy są naszym językiem programowania – zrozumieliśmy nie tylko dla programistów.

INDIGO – platforma do programowania wizualnego

Wykorzystanie grafów do programowania czyni z INDIGO platformę typu LCDP [low-code development platform]. Poszczególne elementy grafu są jak klocki lego, z których zwinnie i szybko budujemy nasze aplikacje.

Platforma INDIGO spełnia wszystkie założenia low-cod: znacząco ułatwia i przyspiesza tworzenie aplikacji, działa w oparciu o globalne wspierane standardy, jest otwarta na integracje z innymi systemami oraz pozwala na zwinne tworzenie aplikacji i umożliwia jej łatwe rozwijanie. Dodatkowo w INDIGO grafy zapewniają możliwość modelowania nawet najbardziej złożonych procesów i tym samym umożliwiają budowanie zaawansowanych systemów informatycznych w pełni dostosowanych do potrzeb.

Tymczasem jest jedna cecha, która wyróżnia INDIGO spośród wszystkich platform LCDP – tym wyróżnikiem jest obszar zastosowań i rodzaj aplikacji, który można przy jej pomocy wytworzyć. **INDIGO to jedyna platforma LCDP do tworzenia aplikacji dla przemysłu, która pozwala na tworzenie oprogramowania integrującego moduły sprzętowe oraz systemy IT.**

Architektura i technologie

Architektura INDIGO jest modułowa i przejrzysta. Została zaprojektowana w sposób, który przewiduje jej rozwijanie. Pozwala także na skalowanie poziome, czyli umożliwia zwiększenie wydajności oprogramowania przez dodawanie kolejnych serwerów (wirtualnych/fizycznych). Dodatkowo, system ma wbudowane mechanizmy agregowania informacji, więc jest przygotowany na przyjmowanie dużej ilości danych (z wielu urządzeń) z bardzo dużą częstotliwością.

Na wyższym poziomie szczegółowości można opisać architekturę INDIGO jako opartą o rozproszone węzły odpowiedzialne za realizację specjalistycznych zadań związanych ze sterowaniem, integracją oraz zbieraniem i przetwarzaniem danych od urządzeń i użytkowników. Każdy z węzłów jest samodzielny w działaniu i po odłączeniu od sieci, np. w przypadku awarii, nadal zbiera i przetwarza informacje, aby przesłać je dalej natychmiast po wznowieniu komunikacji.

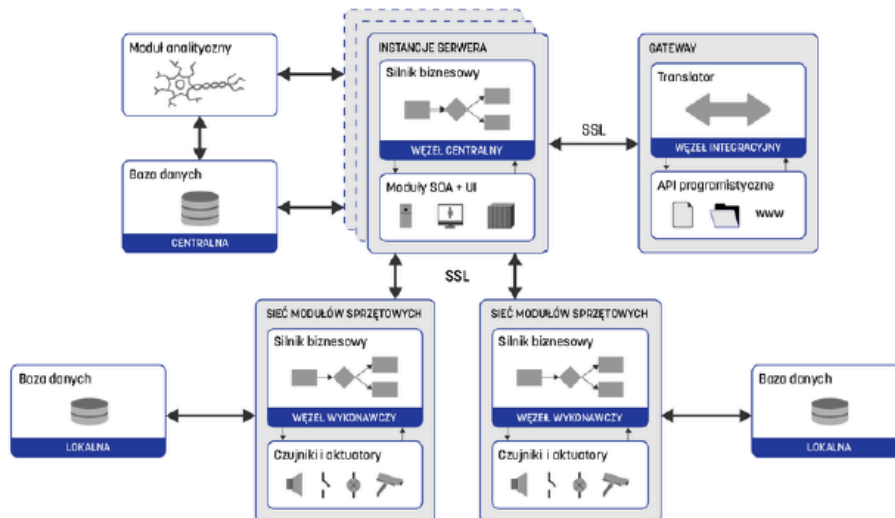
Szczególnym elementem systemu jest węzeł centralny, który posiada zdolność działania w wielu instancjach i umożliwia wysoką skalowalność rozwiązania.

Konstrukcja poszczególnych poziomów systemu oraz programowanie procesów za pomocą grafów umożliwiają zmianę w czasie rzeczywistym logiki działania każdego z elementów systemu oraz podmianę na inny posiadający taki sam zakres funkcjonalny.

Tak zaprojektowana architektura spełnia wszystkie postulaty Przemysłu 4.0, jak przezbrajanie fabryk w czasie rzeczywistym i krótkoseryjną produkcję przy jednoczesnym dbaniu o jakość oraz optymalizację na każdym poziomie działania produkcji,

przedsiębiorstwa oraz całej organizacji.

Rys. 1
Architektura



Aby umożliwić użytkownikom stosowanie optymalnych rozwiązań sprzętowych i programistycznych pochodzących od różnych producentów, a jednocześnie zapobiec przywiązaniu tylko do jednego dostawcy (vendor lock in), INDIGO w całości oparte jest o bezlicencyjne technologie OpenSource (MIT lub równoważne): framework działa na systemach operacyjnych GNU Linux i wykorzystuje sprawdzone, używane i rozwijane rozwiązania, takie jak serwer baz danych PostgreSQL, ORM SQLAlchemy, framework Vue.js, serwer HTTP nginx oraz języki programowania Python 3.7 i JavaScript, TypeScript.

Bezpieczeństwo danych przetwarzanych przez INDIGO jest zapewnione przez asymetryczne szyfrowanie komunikacji za pomocą protokołów SSL/TLS.

Silnik biznesowy

Najważniejszym elementem technologii INDIGO jest lekki silnik procesowy programowany za pomocą grafów BPMN 2.0. Silnik może działać na urządzeniach klasy IoT sterując ich zachowaniem i komunikacją z innymi węzłami oraz serwerem.

Konstrukcja silnika nie ogranicza jego użycia do wybranych obszarów działania. Umożliwia wywoływanie dowolnych funkcjonalności, sterowanie interfejsami i realizację zadań synchronicznych oraz planowanych. Silnik realizuje swoje zadania na podstawie wgranych do niego grafów.

Dzięki oparciu technologii INDIGO o języki skryptowe oprogramowanie może być uruchamiane na dowolnych urządzeniach działających w architekturze x86. Nieserwerowe węzły INDIGO mogą być uruchamiane również na urządzeniach z systemem Android.

Każdy węzeł posiada własną bazę, dzięki czemu np. po restarcie proces może być kontynuowany. Aktualnie system oparty jest o bazę PostgreSQL, ale dzięki zastosowaniu warstwy pośredniej SQLAlchemy możliwe jest podłączenie każdej innej nowoczesnej relacyjnej bazy danych.

Silnik pozwala na wykorzystanie wielu procesów jednocześnie synchronicznie i asynchronicznie oraz ich planowanie w cron. Możliwe jest kolejkowanie komunikatów w przypadku offline i ich wysyłanie po przywróceniu łączności.

Grafi

Platforma INDIGO oparta jest o programowanie graficzne. Grafami programuje się każdą warstwę logiczną – od sterowania modułami sprzętowymi, poprzez opis logiki działania wielu modułów towarzyszących urządzeniu, obiegu informacji w grupie urządzeń aż do przesyłania informacji pomiędzy różnymi grupami urządzeń i systemami IT również w odległych lokalizacjach.

Grafami sterowane jest także działanie interfejsu, tj. pokazywanie i chowanie okien, ich wewnętrzna logika działania oraz wyświetlane treści. W systemie definiuje się tylko szablony określające strukturę widoku i komponentów, z których składa się np. tabela, tekst czy przycisk.

Grafi definiują reakcję na komunikaty interfejsu, sprzętu, innych procesów określając kolejne kroki do wykonania, warunki ich wykonania oraz komunikaty do wysłania. Za ich pomocą definiuje się także obsługę błędów i zachowania w trybie offline.

Realizacja działań na grafie polega na wywołaniu funkcjonalności z generycznych bibliotek – nisko funkcjonalnych jak np. obsługa zmiennych, obsługa komunikacji przez porty, etc. i wysoko funkcjonalnych, jak np. moduł rejestracji przestojów.

Grafy tworzymy za pomocą naszego autorskiego, dedykowanego narzędzia IDE [Integrated Development Environment].

Integracja z zewnętrznymi systemami IT

Framework INDIGO można integrować z zewnętrznymi systemami na dwa sposoby.

Framework INDIGO łączy się z zewnętrznymi systemami udostępniającymi API dowolnego rodzaju (SOAP, SQL, REST itp.) poprzez tworzenie węzłów integracyjnych REST. Mechanizmy wykorzystywane w INDIGO do tworzenia tych węzłów pozwalają na:

- walidacje danych wejściowych i wyjściowych,
- wyświetlanie interfejsu do deweloperskiego wywoływania punktów integracyjnych węzła – użyteczne zarówno dla programistów pracujących w INDIGO jak i zewnętrznych aplikacji,
- uruchamianie węzła w trybie danych testowych, bez połączenia z docelowym systemem.

Deployment

INDIGO w pełni wspiera *continuous deployment* umożliwiając np. zdalne wgrzywanie widoków i zasobów oraz podmianę grafów. System zbiera w sposób scentralizowany logi ze wszystkich węzłów w czasie rzeczywistym.

Framework jest w pełni zmodularyzowany tj. deployment zawiera tylko wymagane do działania moduły.

Interfejs, czyli jak komunikujemy się z użytkownikiem

Wysoka elastyczność i konfigurowalność INDIGO dotyczy także interfejsu, który zaprojektowaliśmy zgodnie z najnowszymi trendami w tworzeniu aplikacji webowych.

Oparty jest o pulpity i widżety, z których konfigurujemy panele zarządzania. Budowa naszego interfejsu umożliwiła prezentację różnego rodzaju informacji w różnych układach graficznych i funkcjonalnych, które wynikają z potrzeb danej implementacji. Tworzenie dedykowanych ekranów jest proste i szybkie, ponieważ komponuje się je z poziomu samego interfejsu bez konfiguracji w plikach i dodatkowego programowania (WYSIWYG). Dzięki temu interfejs może być zindywidualizowany dla każdego użytkownika, ale możliwe jest także stworzenie jednolitego interfejsu dla niehomogenicznego parku maszynowego.

Za pomocą grafów sterujemy także interfejsem

INDIGO w pełni wspiera mechanizmy uprawnień na łączonych rolach, internacjonalizacja, zmianę wielkości elementów interfejsu oraz rozdzielczości. Możliwe jest także dostosowanie wyglądu interfejsu (w tym w trakcie działania) w ramach schematu kolorów, wyglądu kontrolki, etc.

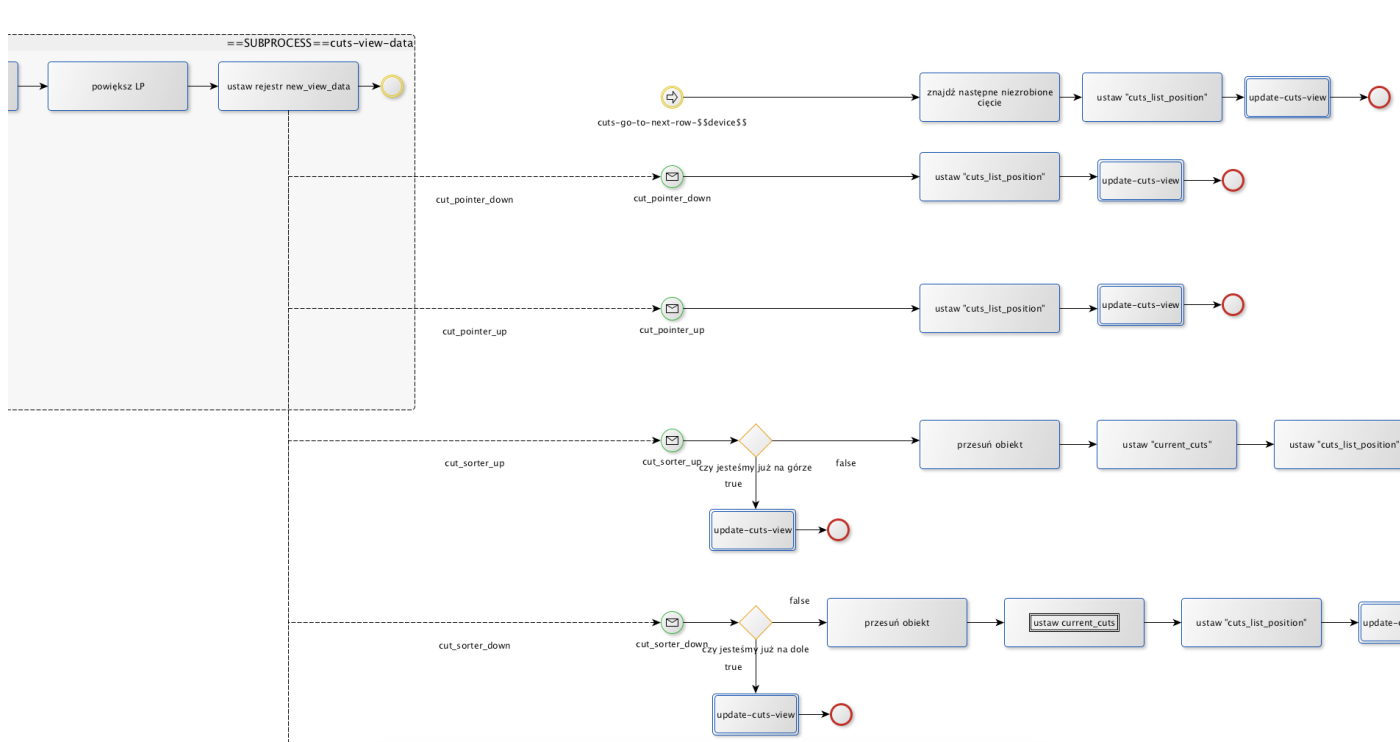
Jak to robimy?

Rys. 2

Szablon opisujący zachowanie widoku.

```
<template-section id='body_section' type='vertical' scheme='fullscreen'>
  <template-section id='header_section' scheme='header'>
    <info-picture id='stabilis_logo' icon='images/stabilis-logo.png' position='start' left='--separation-big' right='--separation-big' />
    <info-line id='console_info' position='start' left='--separation-big' right='--separation-big' interspace='--separation-big' />
    <date-and-time id='clock' position='end' right='--separation-big' stick='end' />
  </template-section>
  <template-section id='main_section' type='vertical' >
    <template-section id='cuts_section' align='start' >
      <template-section id='table_section' type='vertical' size='stretch'>
        <info-table id='cuts_table' grow='expand' />
        <info-line id='comments' size='--font-big' />
      </template-section>
      <template-section id='input_section' type='vertical' align='end' grow='none' size='stretch'>
        <input-number id='input_1kat' />
        <input-number id='input_2kat' stick='end' />
        <input-number id='input_waste' bottom='--separation-small' />
      </template-section>
    </template-section>
    <template-section id='machine_section' size='stretch' grow='none' >
      <info-box id='machine_info' />
    </template-section>
    <template-section id='actions_section' scheme='commands'>
      <execute-command id='actions_navigate' />
      <execute-command id='actions_cuts' />
      <execute-command id='actions_machine' />
    </template-section>
  </template-section>
</template-section>
```

Rys. 3
 Fragment programu w postaci grafu opisującego zachowanie widoku



Rys. 4
 Przykładowy ekran systemu odpowiedzialny za sterowanie maszyną i prezentację parametrów. Wygląd oraz sposób zachowania zostały w całości opisane za pomocą szablonu oraz grafu procesowego.

STABILIS TERMINAL: 150P_1058 UZYTEKOWNIK: operator Maszyny STAN: PRACA [STOP] 12:44:09 PM

DOSTĘPNE ZLECENIA PRODUKCYJNE

ZLECENIA DO REALIZACJI				
NUMER ZLECENIA	KLIENT	METRAŻ [MB]	DATA ZLECENIA	STATUS
▶ 2018/ZPS/SOK/561284	LOREM	200.12	2018-03-29	
2018/ZPS/SOK/556390	IPSUM	201.12	2018-03-29	
2018/ZPS/SOK/556391	DOLOR	201.12		W TOKU

UWAGI: dostawa 26/05

LISTA PAKÓW W ZLECENIU - 2018/ZPS/SOK/561284				
NUMER PAKA	DŁUGOŚĆ [MB]	CIĘŻAR [KG]	ARKUSZY	PACZKA [MB]
▶ 2018/PAK_PROD/SOK/070685	55.12	150.12	5	20.12
2018/PAK_PROD/SOK/070686	56.12	151.12	6	21.12
2018/PAK_PROD/SOK/070687	57.12	152.12	7	22.12

OGÓLNE

WYLOGUJ

MASZYNA

USTAWIENIA I KALIBRACJA	UTRZYMANIE RUCHU	ALARMY
-------------------------	------------------	--------

NAWIGACJA

RĘCZNE ZLECENIE	PRODUKCJA WEWNĘTRZNA
-----------------	----------------------

Testy end-to-end

Aplikacje napisane we INDIGO są testowane za pomocą scenariuszy testowych zapisanych w grafach. W ramach scenariusza testowego, możliwe jest:

- symulowanie działań użytkowników na interfejsie,
- symulowanie sygnałów z urządzeń peryferyjnych (skanery kodów kreskowych itp.),
- symulowanie sygnałów z PLC,
- korzystanie z testowych danych w węzłach integracyjnych,
- sprawdzanie zmian na interfejsie użytkownika,
- sprawdzanie zapytań do węzłów integracyjnych,
- sprawdzanie komunikatów do peryferiów i PLC.

Co nas wyróżnia?

Programowanie za pomocą INDIGO daje unikalne możliwości:

- umożliwia tworzenie stacjonarnych aplikacji z funkcją offline,
- pozwala tworzyć złożone aplikacje, ponieważ modeluje procesy a nie przepływ danych,
- pozwala na dowolność wyboru systemu kontroli wersji,
- posiada wbudowany moduł raportowy,
- umożliwia zdalny monitoring stanowiska pracy oraz odtworzenie działań użytkownika,
- posiada wbudowany moduł do integracji z zewnętrznymi systemami,
- posiada zintegrowane mechanizmy do testów oraz lokalnych danych developerskich.

Co z tego wynika?

Platforma INDIGO daje wiele korzyści, istotnych z punktu widzenia trendów i potrzeb związanych z tworzeniem aplikacji dla przemysłu. Do najważniejszych należą:

- unikalna elastyczność pozwalająca na tworzenie dedykowanych rozwiązań i realizacji bardzo złożonych aplikacji,
- znaczące przyspieszenie tworzenia aplikacji,
- łatwość i bezpieczeństwo wprowadzania zmian do działającej aplikacji, które nie były przewidziane w momencie projektowania i wdrażania oprogramowania,
- stabilność,
- zmniejszenie ilości błędów logiki, regresji i integracji,
- znaczące obniżenie kosztów zmiany wymagań w trakcie projektu,
- możliwość ewolucyjnego rozwijania aplikacji zgodnie ze zwinnymi metodykami produkcji oprogramowania,
- możliwość realizacji projektu przez personel o niższych kompetencjach (wdrożeńowcy zamiast programistów),
- ułatwienie i przyspieszenie procesu analizy z klientem.

Gdzie można zastosować platformę INDIGO?

Elastyczność INDIGO daje możliwości zastosowania platformy w przeróżnych obszarach. Nasza technologia sprawdzi się wszędzie tam, gdzie jest potrzeba zbierania dużej ilości danych z wielu urządzeń, ich analizowania oraz reagowania na zebrane dane. Niezależnie od tego, czy mowa o integrowaniu maszyn w fabryce, systemów automatyki budynkowej czy też różnych modułów sprzętowych w ramach jednego urzędnia (np. vendomat) zawsze dopasujemy się do indywidualnych potrzeb.

Przykładowe obszary zastosowań:

- zbieranie danych, analiza i sterowanie automatyką przemysłową,
- tworzenie oprogramowania dla prototypowych linii produkcyjnych,
- tworzenie zaawansowanych usługomatów i ich sieciowanie (maszyny vendingowe, car/bike sharing etc.),
- tworzenie systemów zarządzania budynkami (integracja systemów budynkowych, sterowanie automatyką)

STABILIS®MES

Globalna cyfrowa transformacja coraz śmielej wkracza do sektora przemysłowego. Rosnące zapotrzebowanie na narzędzia umożliwiające optymalizację produkcji i procesów biznesowych zainspirowało nas do stworzenia pierwszego produktu w oparciu o naszą platformę – oprogramowania umożliwiającego szybkie tworzenie dedykowanych systemów pozwalających na integrację maszyn i systemów IT, czyli opartego o informacje nowoczesnego zarządzania przedsiębiorstwem produkcyjnym. Nazwaliśmy je STABILIS®.

Podsumowanie

Tym, co wyróżnia INDIGO jest unikalna elastyczność, która umożliwia szybkie tworzenie dedykowanych aplikacji w pełni dostosowanych do potrzeb oraz zapewnia łatwość wprowadzania zmian do działającej aplikacji zgodnie z pojawiającymi się potrzebami.

Aplikacje tworzone w INDIGO umożliwiają dwukierunkową komunikację z maszynami lub systemami IT zaprogramowaną za pomocą grafów. To oznacza, że możemy nie tylko zbierać dane, ale także na nie reagować, wysyłając polecenia zwrotne dla maszyn czy informacje zwrotne dla systemów IT (np. ERP).

Dzięki temu, że każdy z procesów przebiegających w dowolnym systemie lub sieci maszyn jest zaprogramowany grafem możliwe jest modyfikowanie logiki działania systemu działającego na naszym oprogramowaniu niemalże w czasie rzeczywistym – wystarczy podmienić graf bez konieczności pisania kolejnych setek linijek kodu.

Grafy pozwalają nam zwiększyć dopasowanie i niezawodność naszego oprogramowania, ponieważ projektowanie funkcjonalności jest zrozumiałe dla zamawiającego i możliwe do modyfikacji nawet na etapie wdrożenia, kiedy to czasami okazuje się, że w praktyce pewne procesy należy zaprojektować inaczej, niż było to zakładane wcześniej.

Nasza podejście sprawia, że oprogramowanie INDIGO wyróżnia wyjątkowa elastyczność rozumiana jako możliwość szybkiego wprowadzania zmian, otwartość na integrację z nowymi technologiami oraz możliwość dodawania nowych funkcjonalności poprzez konfigurację bez konieczności tworzenia dodatkowego oprogramowania. Słowem w pełni realizujemy postulaty metodologii *rapid application development* (RAD).

Aplikacje, które tworzymy, służą przede wszystkim do aktywnej optymalizacji procesów biznesowych stąd naturalną ścieżką wykorzystania naszej technologii jest implementacja mechanizmów AI, machine learning oraz data science.