Antykruchość: inżynieria systemów w najlepszym wydaniu

paź 6, 2025

W niniejszym artykule przedstawiamy nowe kryterium, zwane poziomem gwarantowanej niezawodności i odporności (ARRL), które definiuje QoS w sposób normatywny, głównie poprzez uwzględnienie sposobu, w jaki system radzi sobie z awariami. ARRL definiuje 7 poziomów, z których najwyższy można opisać jako poziom, na którym system staje się antykruchy.

Antifragility: systems engineering at its best

Wstęp

Normy bezpieczeństwa są często powiązane z systemami z programowalnymi komponentami elektronicznymi lub dotyczą ich głównie. Na przykład norma IEC 61508 [2] – tzw. matka wszystkich norm bezpieczeństwa – wyraźnie odnosi się do systemów z programowalnymi komponentami.

Powodem tego jest fakt, że wraz z pojawieniem się programowalnych komponentów w projektowaniu systemów, inżynieria systemów stała się dominującym problemem dziedziny dyskretnej, podczas gdy poprzednie technologie dominowały w dziedzinie ciągłej. W dziedzinie ciągłej komponenty mają wrodzoną właściwość łagodnej degradacji, podczas gdy w systemach dziedziny dyskretnej nie jest to prawdą.

Drugą szczególną cechą jest to, że w dziedzinie dyskretnej przestrzeń stanów jest zazwyczaj bardzo duża, a zmiany stanu mogą następować w ciągu nanosekund. Dlatego bardzo ważne jest upewnienie się, że żadna zmiana stanu nie może doprowadzić systemu do stanu niebezpiecznego. Pomimo identyfikowalnych słabości, różnych w zależności od dziedziny, normy inżynierii bezpieczeństwa narzucają kontrolowany proces inżynieryjny, co skutkuje dość przewidywalnym bezpieczeństwem, które może być certyfikowane przez podmioty zewnętrzne.

Proces ten jest jednak stosunkowo kosztowny i zasadniczo wymaga ponownej certyfikacji całego projektu i systemu za każdym razem, gdy wprowadzana jest zmiana. Podobnie, komponent, taki jak komputer ogólnego przeznaczenia, który jest certyfikowany jako bezpieczny w użyciu w jednej domenie, nie może być ponownie użyty jako taki w innej domenie. To ostatnie stwierdzenie jest jeszcze bardziej hojne.

Przy ścisłym przestrzeganiu norm, w obrębie tej samej domeny każdy nowy system wymaga ponownej certyfikacji lub co najmniej ponownej kwalifikacji,
tak że nawet w obrębie rodzin produktów ponowne użycie jest ograniczone względami bezpieczeństwa.

Pierwszym projektem był projekt ASIL [3]. Przeanalizowano w nim wiele norm, takich jak IEC-61508, IEC-62061, ISO-26262, ISO-13849, ISO-25119 i ISO-15998, a także CMMI i Automotive SPICE, w celu opracowania jednolitego przepływu procesu dla aplikacji o krytycznym znaczeniu dla bezpieczeństwa, ze szczególnym uwzględnieniem sektora motoryzacyjnego i maszynowego.

Ten ostatni opiera się na ogólnym metamodelu inżynierii systemów, pokazując, że możliwe jest wykorzystanie abstrakcyjnego modelu wyższego poziomu w inżynierii systemów (w tym przypadku inżynierii bezpieczeństwa). Jednocześnie zakłada on szereg ukrytych założeń. Na przykład, niespójne stosowanie terminologii i pojęć w różnych domenach stanowi poważną przeszkodę w ponownym wykorzystaniu.

W związku z tym oba projekty dostarczyły wglądu, że ściśle rzecz biorąc, międzydomenowe ponowne wykorzystanie artefaktów i komponentów związanych z bezpieczeństwem nie jest możliwe ze względu na ogromne różnice między standardami bezpieczeństwa. W dalszej części zobaczymy, że pojęcie bezpieczeństwa jako celu (często nazywanego poziomem integralności bezpieczeństwa, SIL) różni się w zależności od domeny.

Można to uzasadnić. Zapewnienie bezpieczeństwa zapewniane dla danego systemu jest specyficzne dla tego systemu w jego certyfikowanej konfiguracji
i certyfikowanym zastosowaniu. Często stoi to w sprzeczności z praktyką inżynierską. Inżynierowie stale budują systemy,
ponownie wykorzystując istniejące komponenty i łącząc je w większe podsystemy. Wynika to nie tylko z korzyści ekonomicznych, ale
często zwiększa zaufanie do systemu, ponieważ ryzyko błędów resztkowych będzie niższe, przynajmniej jeśli stosowany jest proces kwalifikacji
tych komponentów. Niemniej jednak normy inżynieryjne i bezpieczeństwa zawierają stosunkowo niewiele zasad i wytycznych dotyczących ponownego wykorzystania komponentów, co utrudnia rozwój bezpiecznych systemów poprzez ich kompozycję.

Innym pojawiającym się aspektem systemowym jest zarządzanie cyklem życia. Aspekt ten wysuwa się na pierwszy plan, ponieważ systemy mają coraz dłuższy czas życia, a komponenty są coraz bardziej połączone w większy system. Ponadto, czas życia jest taki, że poszczególne komponenty będą wymieniane lub modernizowane w miarę starzenia się lub mogą zostać zastąpione nowszymi, bardziej wydajnymi lub wydajniejszymi technologiami.

Typowymi przykładami są Internet Rzeczy, inteligentne sieci i infrastruktura, która staje się „inteligentna” dzięki wykorzystaniu elektroniki wbudowanej. Cechą wyróżniającą jest to, że rozwinięty system nie może już być rozwijany w izolacji, ponieważ staje się częścią systemu systemów.

Ta właściwość odpowiada pojęciu antykruchości, pierwotnie sformułowanemu jako koncepcja jakościowa przez Taleba.

Poziomy integralności bezpieczeństwa

Po zdefiniowaniu początkowej architektury, kolejnym ważnym działaniem jest przeprowadzenie analizy FMEA (analizy trybów i skutków awarii). Podczas gdy analiza HARA ma charakter odgórny i obejmuje stany środowiska i operatora, FMEA analizuje wpływ uszkodzonego komponentu na prawidłowe funkcjonowanie systemu (w szczególności pod kątem potencjalnych zagrożeń). Awarie można klasyfikować według ich źródła.

Awarie losowe są zazwyczaj wynikiem przyczyn fizycznych, podczas gdy awarie systematyczne są wynikiem błędów projektowych lub implementacyjnych. We wszystkich przypadkach, gdy stosowana jest elektronika programowalna, ich skutek jest często taki sam: system może natychmiast lub w późniejszym czasie przejść w stan niebezpieczny. Możliwe jest również, że pojedyncze lub nawet wiele usterek nagromadziło się, ale pozostaje uśpionych do momentu, aż błąd zostanie wywołany przez określone zdarzenie.

Wyniki analizy FMEA nie są przeznaczone do ponownego wykorzystania w innym systemie, nawet jeśli analiza jest na tyle ogólna, że umożliwia ponowne wykorzystanie w innych systemach. W związku z tym nie zdefiniowano kryterium, które pozwalałoby nam klasyfikować komponenty pod względem ich niezawodności, nawet jeśli można oszacować niektóre parametry, takie jak średni czas bezawaryjnej pracy (MTBF), aczkolwiek

w danym kontekście. W ostatniej części niniejszego
artykułu wprowadzamy kryterium uwzględniające zachowanie się usterek. Należy zauważyć, że choć ma ono charakter ogólny, należy jasno stwierdzić, że niniejszy artykuł koncentruje się na komponentach programowych działających na programowalnych komponentach elektronicznych.

Chociaż oba sektory mają swoje własne normy bezpieczeństwa, istnieje zasadnicza różnica. Podczas gdy w większości krajów systemy lotnicze i kolejowe podlegają ścisłym regulacjom i wymagają certyfikacji, w sektorze motoryzacyjnym normy prawne są znacznie słabsze, częściowo dlatego, że kierowca jest uważany za główną przyczynę wypadków. To ostatnie znacząco wpływa na czynnik „sterowalności" w określaniu wymaganego poziomu bezpieczeństwa (SIL).

Główną wadą SIL jest jednak to, że opiera się on na średnich wartościach statystycznych, często bez informacji o rozrzutach statystycznych. Nie tylko uzyskanie prawidłowych wartości jest bardzo trudne, a nawet niemożliwe, ale zależą one również od kilku czynników, takich jak sposób użytkowania, środowisko operacyjne oraz umiejętności i wyszkolenie operatora. Prawidłowe wartości statystyczne, takie jak wartość średnia, zakładają wystarczająco dużą bazę próbkowania, która często nie występuje. Co więcej, ignoruje to, że zdarzenia zakłócające, takie jak bardzo mało prawdopodobny wypadek, mogą całkowicie zmienić te wartości.

Jako przykład przytaczamy samolot Concorde, który był uważany za najbezpieczniejszy samolot na świecie, dopóki jeden z nich nie rozbił się śmiertelnie. Po katastrofalnym zdarzeniu „stał się" niemal natychmiast jednym z najbardziej niebezpiecznych samolotów na świecie, przynajmniej statystycznie rzecz biorąc, częściowo dlatego, że samolot był mniej intensywnie użytkowany niż większość samolotów komercyjnych.

To musi być punkt wyjścia do opracowywania bezpiecznych systemów z dyskretnymi komponentami, jeśli ktoś naprawdę poważnie myśli o bezpieczeństwie. Zasadniczo, łagodna degradacja nie dotyczy dyskretnych systemów przestrzeni stanów.

Brakujące ogniwo w inżynierii bezpieczeństwa: kryterium ARRL

1. Komponent musi być zaprojektowany tak, aby zapobiegać propagacji błędów. Dlatego interfejsy muszą być jednoznacznie identyfikowalne i zaprojektowane z uwzględnieniem „ochrony". Interfejsy te muszą być również jedynym sposobem interakcji komponentu z innymi komponentami. Stan wewnętrzny nie jest dostępny z innego komponentu, ale może być udostępniony jedynie za pośrednictwem dobrze zdefiniowanego protokołu (np. poprzez przesłanie kopii stanu).

2. Mechanizm interakcji, na przykład połączenie sieciowe, musi posiadać co najmniej te same dane uwierzytelniające ARRL, co komponenty, które łączy. W rzeczywistości w wielu przypadkach poziom ARRL musi być wyższy, aby utrzymać wystarczająco wysoki poziom ARRL na poziomie
(pod)systemu składającego się z komponentów.

3. Dlatego lepiej jest traktować interfejs jako komponent sam w sobie, niż na przykład zakładać niejawną komunikację między komponentami.

Należy pamiętać, że gdy komponent i podłączone do niego interfejsy spełniają wymagany poziom ARRL, jest to warunek konieczny, a nie wystarczający, aby system spełnił dany poziom ARRL i SIL. Sama aplikacja opracowana na podstawie zmontowanych komponentów i ich interfejsów również musi zostać opracowana tak, aby spełniała odpowiedni poziom ARRL.

Zilustrowano architekturę ARRL

Chociaż Tabela 3 omawia kilka poziomów technologicznych w systemie lub komponencie, uwaga skupia się na poziomach sprzętowym (elektronicznym) i programowym. Najniższy poziom to w dużej mierze dziedzina ciągła, w której obowiązują zasady i prawa nauki o materiałach. Ogólnie rzecz biorąc, dziedzina ta jest dobrze poznana, a stosowanie marginesów projektowych i bezpieczeństwa minimalizuje większość zagrożeń bezpieczeństwa. Ponadto komponenty w tej dziedzinie często wykazują łagodną degradację, cechę, która z natury przyczynia się do bezpieczeństwa. Dotyczy to nawet materiałów półprzewodnikowych używanych do opracowywania programowalnych układów scalonych.

Poziomy związane ze środowiskiem i użytkownikiem/operatorem systemu są w większości związane z czynnikami zewnętrznymi, które mogą stwarzać niebezpieczne sytuacje. Dlatego należy je uwzględnić podczas opracowywania systemu i odgrywają one ważną rolę w ocenie ryzyka (HARA). Jednakże, jako takie, są to czynniki zewnętrzne i często unikalne dla każdego systemu, współczynnik ponownego wykorzystania (poza na przykład identyfikacją wzorców i scenariuszy wielokrotnego użytku) jest ograniczony.

Rys. 2. Ogólny widok komponentu według ARRL

Rysunek 2 ilustruje ogólny obraz komponentu. Jest on postrzegany jako blok funkcjonalny, który przyjmuje wektory wejściowe, przetwarza je i generuje wektory wyjściowe. W ogólnym ujęciu przetwarzanie można postrzegać jako funkcję przejścia komponentu. Podczas gdy ta ostatnia terminologia jest najczęściej stosowana w dziedzinie ciągłej, w dziedzinie dyskretnej funkcja przejścia jest często maszyną stanu lub zbiorem współbieżnych maszyn stanu. Ważne dla widoku ARRL jest to, że funkcja przetwarzania nie jest bezpośrednio powiązana z wejściami i wyjściami, ale poprzez interfejsy komponentu, które działają jako strażnicy.

Rys. 3. Ogólny komponent ARRL-1

Ponieważ ARRL-0 nie daje żadnych gwarancji co do swojego zachowania, możemy z gracją pominąć ten poziom i zacząć od poziomu ARRL-1 (rys. 3). Takiemu komponentowi można jedynie częściowo „zaufać”, tj. na tyle, na ile został przetestowany. Niepewność związana jest z nieprzewidzianymi wartościami wejściowymi; wątpliwościami co do kompletności zabezpieczeń wejścia/wyjścia, pozostałymi błędami w funkcji przetwarzania, błędnymi założeniami (np. błędnymi wymaganiami [19]), a zatem mogą występować nieprzewidziane wartości wyjściowe. Innymi słowy, chociaż raport z testów dostarcza pewnych dowodów, brak błędów nie jest gwarantowany, a zatem komponent ARRL-1 nie może być używany jako taki w systemach o znaczeniu krytycznym dla bezpieczeństwa.

Poza ARRL-5: antykruchość

Antykruchość to termin cytowany przez Taleba [7], głównie w kontekście subiektywnego ludzkiego kontekstu społecznego. Cytuje on ten termin, aby wskazać na coś wykraczającego poza solidność i odporność, co reaguje na stresory (i podobne) poprzez faktyczną poprawę swojej odporności na nie.

Przyjmując ten pogląd w kontekście inżynierii systemów, dostrzegamy, że takie systemy już istnieją. Wyróżnia się je poprzez traktowanie systemu jako elementu większego systemu, obejmującego środowisko operacyjne,

jego ciągłe procesy i wszystkich interesariuszy. Dalsze różnice to kultura otwartości, ciągłe dążenie do doskonałości oraz istnienie licznych wielopoziomowych pętli sprzężenia zwrotnego, w których niezależne organy kierują i sterują systemem jako całością. Rezultatem jest system, który ewoluuje w kierunku wyższych stopni antykruchości.

Istotną różnicą w porównaniu z tradycyjną inżynierią jest to, że system jest stale redefiniowany i adaptowany w interaktywnym procesie, którego celem jest zwiększenie jego antykruchości. Jak widzieliśmy we wcześniejszych rozdziałach, w dziedzinie takiej jak lotnictwo wdrożono proces, który na przestrzeni lat doprowadził do wzrostu jakości usług (QoS). Samoloty są najbezpieczniejszym, ale także najbardziej ekonomicznym i energooszczędnym sposobem podróżowania, przekraczającym minimalną odległość. Jak zobaczymy, inżynieria systemów stosuje już pewne zasady antykruchości, ale nie w sposób ściśle normatywny, jak to określa ARRL. Zastosowanie koncepcji antykruchości do kryterium ARRL pozwala nam zdefiniować dwa nowe poziomy normatywnego kryterium ARRL. ARRL-6 wskazuje na system, który zapobiegawczo dąży do unikania awarii poprzez konserwację i naprawy. ARRL-7 wymaga szerszego procesu, który jest w stanie nie tylko naprawiać, ale także aktualizować system w sposób kontrolowany, bez zakłócania jego funkcji, chyba że jest to konieczne. Biorąc pod uwagę istnienie systemów o takich (częściowych) właściwościach, nie jest jasne, czy użycie neologizmu „antykruchość" jest uzasadnione w celu zastąpienia niezawodności i odporności, nawet jeśli wskazuje to na wyraźny poziom jakościowy i wyróżniający.

Będzie to wymagało dalszych badań. Normatywne poziomy ARRL opisują, jak sama nazwa wskazuje, poziomy niezawodności i odporności. Podchodzą do koncepcji łagodnej degradacji poprzez redundancję, zakładając jednak, że w przypadku braku błędów komponenty systemu można uznać za wolne od błędów. Dodatkową funkcjonalność i redundancję (która jest również wolna od błędów) należy postrzegać jako ulepszenie na poziomie architektury lub procesu.

Jednak we wszystkich przypadkach, wbrew koncepcji antykruchości, system nie zyska na odporności ani niezawodności. Może jedynie opóźnić katastrofalne awarie, jednocześnie tymczasowo utrzymując zamierzone usługi. Osiąga to, zakładając, że wszystkie rodzaje błędów można przewidzieć, co jest najnowocześniejszym stanem wiedzy inżynierskiej. Oczywiście w praktyce nie da się przewidzieć wszystkich błędów i dlatego potrzebna jest dodatkowa warstwa, aby sobie z nimi poradzić. Proponowany schemat wprowadza już dwie koncepcje, które są niezbędne, aby pójść o krok dalej. Po pierwsze, istnieje redundancja w architekturze i procesie, a po drugie, istnieje funkcja monitorująca, która działa poprzez rekonfigurację systemu po wykryciu błędu.

Założenia antykruchości

Jak więc system może stać się „lepszy" w obliczu błędów? Wprowadzając metrykę jako cel, musimy w jakiś sposób mierzyć i wprowadzać pętle sprzężenia zwrotnego. Jeśli ekstrapolujemy i skalujemy, zakładamy, że system posiada pewien rodzaj automodelu, którego może użyć do porównania swojego obecnego stanu z celem odniesienia.

W związku z tym albo projektant musi zawrzeć ten model w systemie, albo model jest zewnętrzny i staje się częścią systemu. Jeśli weźmiemy pod uwagę systemy, które od samego początku zawierają swój automodel, to oczywiste jest, że stawanie się „lepszym" systemem ma swoje granice, a granicą tą jest pomysł projektantów w momencie powstawania koncepcji. Chociaż istnieją systemy, które ewoluują, aby osiągnąć lepsze optimum (pomyślmy o sieciach neuronowych lub algorytmach genetycznych), systemy te ewoluują w kierunku wartości granicznej.

Innymi słowy, nie ewoluują, lecz zbiegają się. Z drugiej strony, jeśli rozszerzymy system jak na rys. 1, to system może ewoluować. Może ewoluować i ulepszać się, ponieważ bierzemy pod uwagę jego otoczenie i wszystkich

interesariuszy, w tym użytkowników, jako część systemu. Stale dostarczają oni informacji o wydajności systemu i podejmują działania w celu jego poprawy. Oznacza to również, że proces inżynieryjny nie kończy się w momencie pierwszego uruchomienia systemu. W rzeczywistości nigdy się nie kończy, ponieważ doświadczenie jest przenoszone na nowsze projekty.

Istnieje wiele przykładów działających już systemów antykruchych, być może nie zawsze idealnych, choć w większości przypadków. Doskonałym przykładem jest branża lotnicza, która dzięki spadającej co roku liczbie ofiar śmiertelnych i rosnącej jakości usług dowodzi, że spełnia kryterium antykruchości. Co więcej, odnosi sukces komercyjny. Przyjrzyjmy się zatem niektórym jej właściwościom i wyciągnijmy ogólne zasady, odzwierciedlone w normach i praktyce lotniczej.

Wnioski i przyszłe prace

W niniejszym artykule przeanalizowano koncepcję poziomu integralności bezpieczeństwa (SIL) i umieszczono ją w szerszej perspektywie, obejmującej jakość usług i wiarygodność. Koncepcje te są bardziej ogólne i wyrażają najwyższe wymagania systemu z perspektywy potencjalnego użytkownika. Omówiono pewne słabości koncepcji SIL, głównie jej probabilistyczne ujęcie systemu, podczas gdy inżynieria często opiera się na kompozycji z wykorzystaniem komponentów lub podsystemów.

Wprowadzono nową koncepcję o nazwie ARRL, definiującą normatywne kryterium dla komponentów i ich interakcji. Wykazano jednak, że SIL i ARRL są komplementarne. Zdefiniowano przepływ procesu z obsługą ARRL. Ma on tę zaletę, że lepiej oddziela dodatkowe funkcje bezpieczeństwa od standardowego wsparcia przypadków użycia niż tradycyjne, bardziej monolityczne podejście. W przyszłych pracach koncepcja zostanie dodatkowo zweryfikowana i zastosowana w kontekście aplikacji krytycznych dla bezpieczeństwa.

Pomoże to w pogłębieniu kryterium i umożliwi jego wykorzystanie do definiowania komponentów objętych umową. Zagadnienia wymagające dalszego doprecyzowania to na przykład:
– W jaki sposób poziom ARRL jako cel projektowy można przekształcić w cele podrzędne?
– Kiedy kontrakt jest kompletny i wystarczający, aby potwierdzić osiągnięcie określonego poziomu ARRL?
– W jaki sposób kontrakt i dowody komponentu mogą być dostarczone w sposób niezależny od domeny aplikacji?
– Jaki jest wpływ na proces inżynierii bezpieczeństwa/systemów?
– Jaki jest wpływ na architekturę systemu?

Kolejnym ważnym zagadnieniem jest analiza, w jaki sposób kompozycja systemu z wykorzystaniem komponentów kwalifikowanych przez ARRL prowadzi do pojawiania się właściwości, które mogą prowadzić do stanu krytycznego dla bezpieczeństwa. Podstawowym założeniem jest to, że system może już znajdować się w stanie krytycznym, postrzeganym jako zbiór błędnych stanów obecnych w jego komponentach, zanim zdarzenie wywoła stan katastrofalny dla całego systemu.

Chociaż ten aspekt został pokrótce poruszony poprzez wymóg obsługi partycjonowania i odpowiednich poziomów ARRL dla komponentów interakcji, wymaga on dalszej uwagi. Interesującym pytaniem jest na przykład, czy taki stan krytyczny można wykryć, zanim doprowadzi do zdarzenia katastrofalnego.

W Altreonic trwają obecnie prace nad zastosowaniem kryterium ARRL do wewnętrznie opracowanego systemu operacyjnego czasu rzeczywistego OpenComRTOS [20]. Chociaż formalnie system został opracowany i dostępnych jest wiele dowodów potwierdzających, nadal brakuje dowodów potwierdzających.

Znacznie ułatwiło to wykorzystanie portalu GoedelWorks, który umożliwia importowanie repozytorium oprogramowania i jego dokumentacji uzupełniającej. Większość zidentyfikowanych problemów jest związana z zastosowanym procesem. Wskazuje to, podobnie jak w przypadku większości projektów inżynierii bezpieczeństwa, że rozwój oparty na ARRL musi uwzględniać kryteria normatywne od samego początku. W takim przypadku dowody potwierdzające, wygenerowane i przechowywane w repozytorium GoedelWorks, zapewnią pakiet „kwalifikacji" dla opracowanego produktu. Jest to podobne do wymagań kwalifikacyjnych dla komponentów i podsystemów nabywanych zewnętrznie, które znajdują się w większości norm bezpieczeństwa.

Różnica polega na tym, że komponent zakwalifikowany przez ARRL będzie znacznie bardziej niezależny od domeny, co jest również celem projektowym, który spełnia ogólny metamodel GoedelWorks. Niemniej jednak uważamy, że kryterium ARRL, ze względu na swój charakter normatywny, stanowi obiecujące podejście do osiągania bezpieczeństwa w różnych domenach i systemach w rodzinie produktów poprzez komponowanie kwalifikowanych, godnych zaufania komponentów. Jednocześnie sugeruje ono, że specyfikacja komponentu wraz z jego umową i dowodami potwierdzającymi jest złożonym przedsięwzięciem, ale zgodnym z niekiedy niewypowiedzianymi założeniami, które można znaleźć w podręcznikach dotyczących bezpieczeństwa i inżynierii systemów.