

Streszczenie

**wydzielonych fragmentów „Ubiquitous computing systems”  
oraz „Pervasive systems”**

z „Distributed Systems 3rd edition 2017”,  
strony 40-43,  
Maarten van Steen, Andrew S. Tanenbaum

streszczenie wykonał  
Cezary Troska

## Wyjaśnienie użytych tłumaczeń

Pojęcia „pervasive systems” oraz „ubiquitous computing systems” w języku polskim są sprowadzone do wspólnego określenia „przetwarzanie rozpowszechnione”. W tym streszczeniu będę tłumaczył je jako odpowiednio „systemy powszechne” i „systemy powszechne-ciągłe”, aby dochować wierności rozróżnieniu obecnemu w naszym podręczniku.

## Systemy powszechne

Systemami powszechnymi nazywamy systemy rozproszone wykonujące swoje zadania w sposób subtelny i mało zauważalny. Często nie mają jednego, wyróżniającego się interfejsu i zamiast tego polegają na dostępnych sensorach w celu zbierania informacji o zachowaniach użytkownika. W interakcję z użytkownikiem wchodzi poprzez wywoływanie bodźców przekazujących informacje, a nawet mające na celu sterowanie zachowaniem korzystającego.

Wiele urządzeń w systemach powszechnych charakteryzują się niewielkimi rozmiarami, zasilaniem na baterie oraz korzystaniem z sieci bezprzewodowych.

## Systemy powszechne-ciągłe

Pojęcie będące tytułem tej sekcji to jeden z trzech typów systemów powszechnych (pozostałe dwa to systemy mobilne oraz sieci czujników).

Cechą charakterystyczną systemów powszechnych-ciągłych jest ich stała aktywność. Oznacza to, że użytkownik cały czas wchodzi w interakcje z systemem, nawet gdy nie jest tego świadom. Główne wymogi, którymi powinny się kierować to rozdystrybuowanie, interakcja, świadomość kontekstu, autonomia i inteligencja.

## Rozdystrybuowanie

Elementy systemu powinny być połączone w sposób umożliwiający stwarzanie iluzji korzystania z jednego, scentralizowanego systemu.

## Interakcja

Inwazyjność systemu powinna być sprowadzona do minimum. Użytkownicy mogą być nieświadomi dostarczania informacji programowi. Oprogramowanie może wykorzystywać akcje użytkownika skierowane do innych programów, aby sterować swoim zachowaniem. Tak zebrane dane wejściowe nazywamy danymi dyskretnymi.

## Świadomość kontekstu

System musi być w stanie analizować otrzymane dane wejściowe w kontekście sytuacji, w jakiej aktualnie znajduje się użytkownik. Program musi wiedzieć jak odpowiedzi na pytania „gdzie?”, „kto?”, „kiedy?” i „co?” powinny przekładać się na jego funkcjonowanie. Zaproponowane tu pytania są najczęściej stosowanymi w praktyce, jednak inne informacje też mogą tworzyć kontekst, w zależności od specyfiki problemu.

## Autonomia

Elementy systemu powinny mieć duże możliwości niezależnego podejmowania akcji i reagowania na zmiany. Jest to istotne, ponieważ centralne podejmowanie decyzji w każdej sytuacji wiązałoby się z ogromnym obciążeniem „administratora” podejmującego takie decyzje. Przykładami sytuacji, w których lokalne podejmowanie decyzji sprawdza się lepiej od centralnego może być monitorowanie dostępnych aktualizacji i aplikowanie ich, czy uzyskiwanie adresów IP w celach komunikacji sieciowej.

## Inteligencja

Systemy powszechne-ciągłe powinny stosować sztuczną inteligencję w celu rozwiązywania szerokiego spektrum sytuacji nieprzewidywalnych, których obsługa zwykle wymagałoby zaawansowanych algorytmów. Dokładniejsze kierowanie się tym kryterium będzie możliwe dopiero w przyszłości, gdy dalszy rozwój SI dostarczy nam rozwiązań przystosowanych do operowania w systemach rozproszonych.