

Systém je souběžný když obsahuje více než jeden aktivní komponent.

Aktivní komponent – je asociován s vykonáváním systému (např. procesy, vlákna, klienti, aktivní objekty, uživatelé, atd.)

Pasivní komponent – nemá svoji vlastní aktivitu a může být použit aktivními komponenty (např. objekty, moduly, systémové prostředky, soubory, servery, atd.)

Tři kategorie souběžných systémů:

- nezávislé (nebo disjunktní, rozpojené);
- kooperační (spolupracující);
- konkurenční.

Konkurenční souběžnost existuje, když dva nebo více nezávisle navržené aktivní komponenty nevědí o existenci druhého, ale používají tytéž pasivní komponenty.

■ Aktivní komponenty musí soutěžit o pasivní komponenty a mohou je vlastnit (disponovat) dokud bude potřeba.

Kooperační souběžnost existuje když několik komponent spolupracují (tj. provádí nějakou práci společně a jsou si toho vědomy).

■ Komponenty mohou komunikovat prostřednictvím sdílených prostředků nebo explicitně.

■ Komponenty jsou navržené společně, aby mohly spolupracovat při dosažení společného cíle a používat pomoc a výsledky navzájem.

■ Komponenty synchronizují svoje provedení a mohou čekat na informaci vypočítanou jiným komponentem.

Kooperace zahrnuje všechny interakce, které jsou očekávané a žádoucí.

Konkurence zahrnuje interakce, které ačkoli jsou očekávané a přípustné, nejsou žádoucí.

Obnovení v souběžných systémech:

- Odrolování (odrolování všech komponentů závadné jednotky do předchozího správného stavu);
- Přerolování (obnovení účastníků jednotky do nového správného stavu).

Atomičnost: Jestliže závada vznikla uvnitř jednotky i není možné tolerovat ji transparentně, bude garantováno, že obalující jednotka bude o tom informována a bude se nacházet v stavu, ve kterém by se nacházela kdyby vykonání jednotky vůbec nezačalo.

■ Vykonávání vnořených jednotek jsou atomická pro kteroukoli obalující jednotku a pro kteroukoli externí komponentu.

Pravidla vnoření: [1], [2]

- každá vnořená jednotka musí být dokončená dříve než může být dokončená obalující jednotka ;
- vykonávání vnořené jednotky je nedělitelné a neviditelné (tj. Atomické) pro obalující jednotku a pro sourozenecké jednotky; (všechno nebo nic)!
- výsledky vnořené jednotky nemohou být viditelné mimo obalující jednotku až do té doby pokud tato obalující jednotka nebude dokončená;
- pouze komponenty, které se nachází v obalující jednotce mohou spoluúčinkovat (zúčastnit se) ve vnořené jednotce.

Sekvenční strukturování:

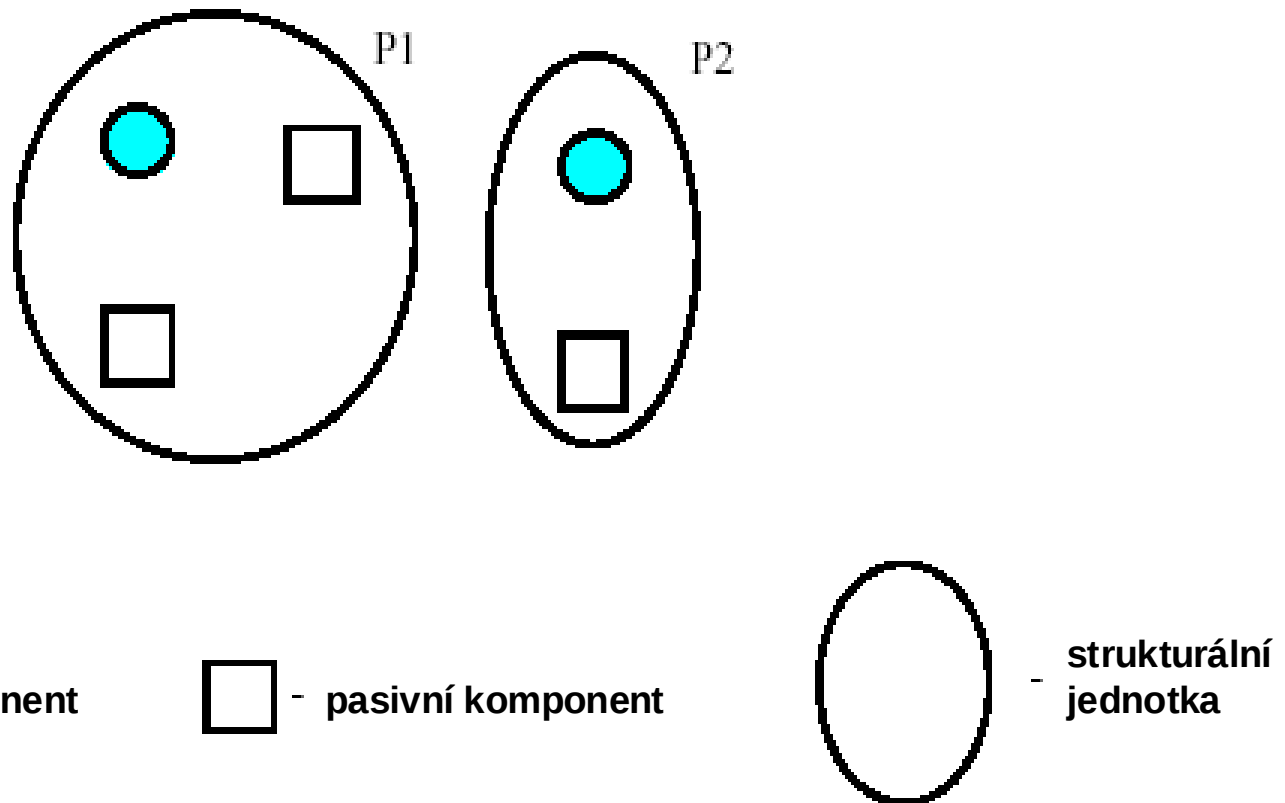
- Pro návrh disjunktních systémů se používá sekvenční strukturování.
 - Sekvenční programy mohou být statické nebo dynamické.
- Dynamický systém je reprezentován jako zásobník vnořených bloků kontextu.

Metody pro zajištění OPZ: [3]

Obnovovací bloky;

NVP;

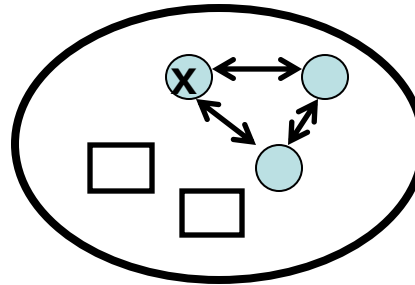
Ošetření výjimek.



Dva disjunktní (rozpojené) procesy P1 and P2 mají přístup k rozpojené množině pasivních komponentů

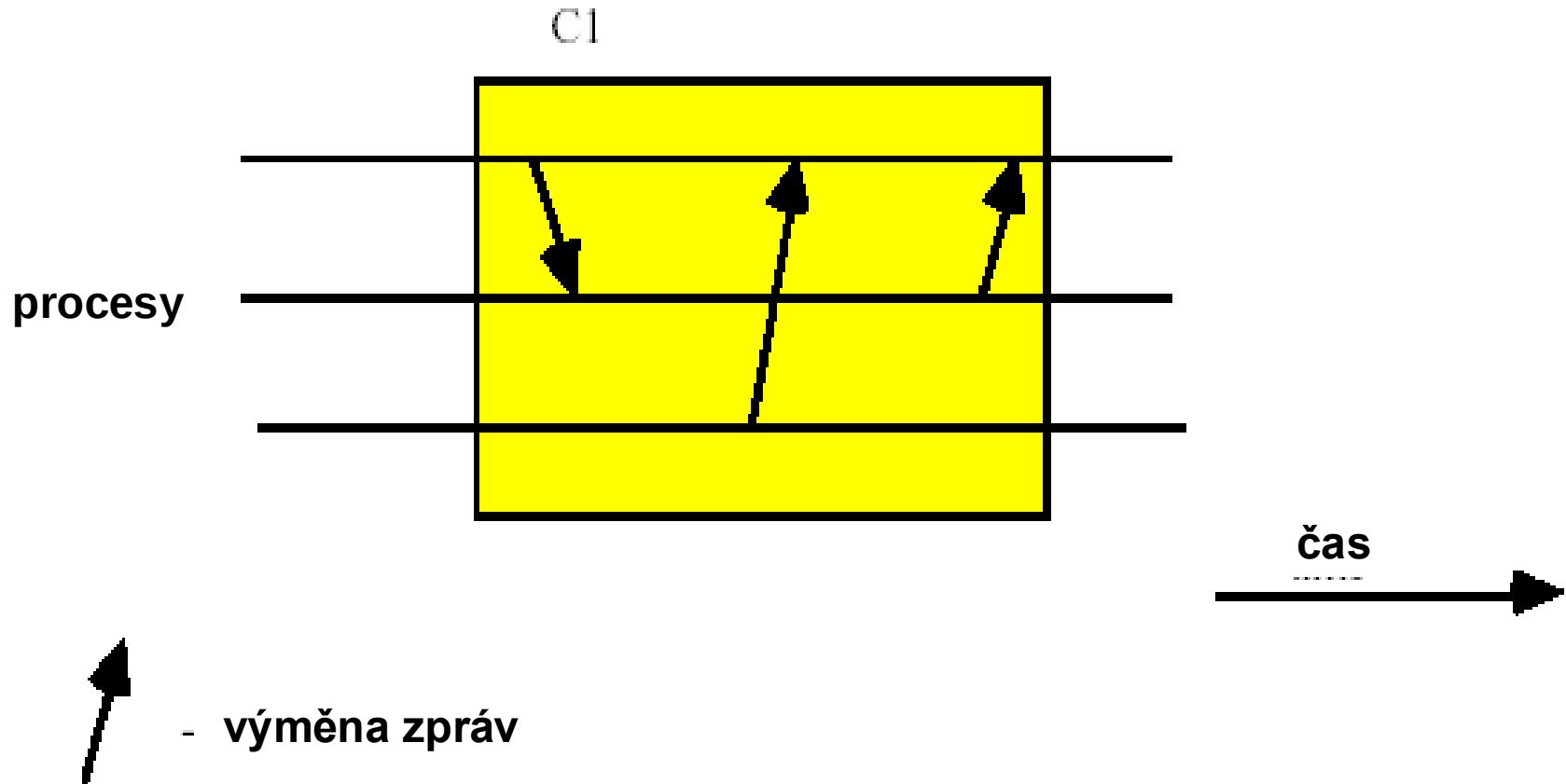
Dvě hlavní metody pro strukturování kooperačních systémů a zároveň pro včlenění prostředků zajišťujících jejich odolnost proti závadám:

- Konverzace;
- Atomické akce.



Konverzace: [1]

- Logické procesy vstupují do konverzace současně (v kontextu distribuovaných systémů to znamená, že vstupní události jsou souběžné).
- Každý proces zřizuje obnovovací bod.
- Procesy si volně vyměňují informace uvnitř konverzace, ale nemohou komunikovat s žádným procesem zvenku (mimo konverzace).
- Když se všechny procesy, které se zapojily do konverzace, dostanou do konce konverzace, bude vyvolán přijímací test.
- Jestli PT bude úspěšný, procesy opustí (odcházejí) konverzaci. Jinak, stavy všech procesů budou obnoveny (vrátí se do stavu zachovaného v obnovovacím bodu).
- Pro každý proces se používá několik alternativ. Po obnovení druhá alternativní varianta každého procesu bude vykonávána.
- Konverzace mohou být vnořené (tj. vkládané jedna do druhé).



Konverzace C1: tři účastníci

Atomické akce: [4], [5]

Byly vyvíjeny jako zevšeobecnění konverzace.

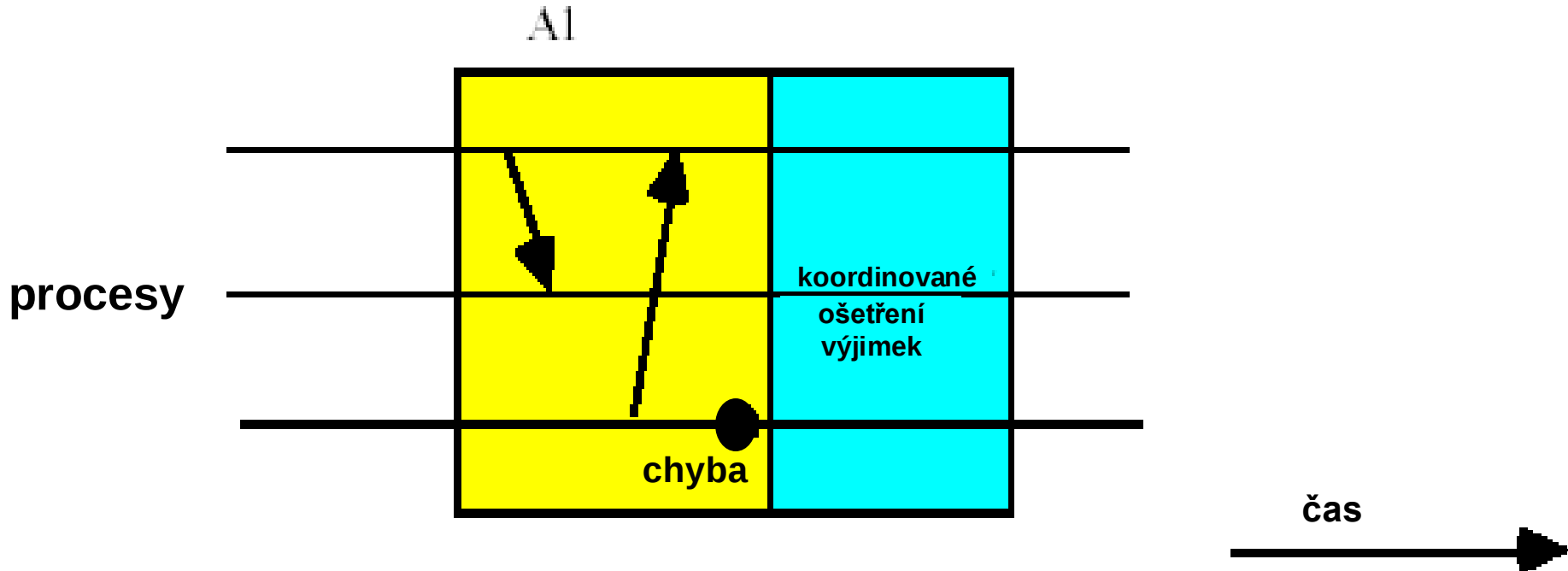
Atomické akce používají

- Odrolování
- Přerolování
- nebo jejich kombinace.

Odrolování je identické tomu, které se používá v konverzaci.

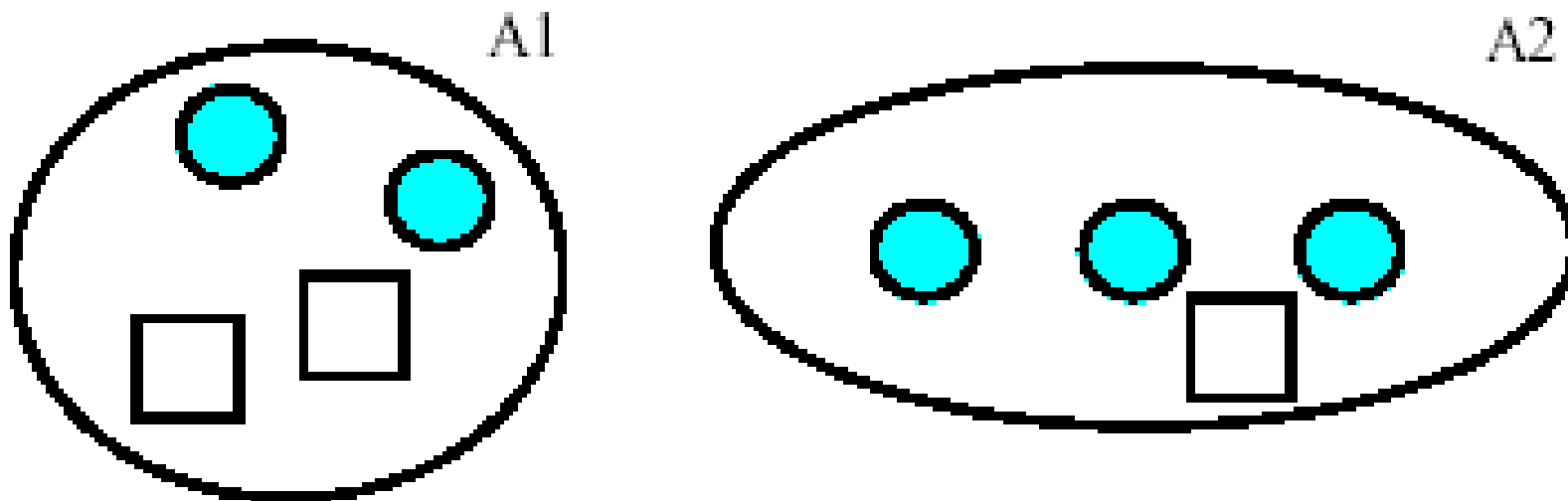
Přerolování se spoléhá na *Mechanismus Ošetření Výjimek* a může včlenit (začlenit) dodatečný mechanismus pro vyřešení četných výjimek, které vznikly souběžně v několika komponentech.

- Ke každému účastníku AA (procesu) je přiřazen svůj odpovídající Handler; Handlers, které určeny pro ošetření stejné výjimky, budou vyvolané ve všech účastnících.



Atomické akce A1: chyba byla odhalená, všichni tři účastníci jsou zapojeni do koordinovaného obnovení (ošetření výjimky)

Systém sestává ze dvou Atomických akcí



Dvě atomické akce A1 a A2 . Akce A1 má dva aktivní komponenty a akce A2 má tři aktivní komponenty

- Každá akce má několik procesů, které navzájem komunikují. Vykonání akce A1 a A2 je disjunktní (není souvislé).
- Pasivní komponenty uvnitř akce jsou jak prostředky pro kooperace tak i data reprezentující stav akce/systému nebo procesu.
- Všeobecně, za výsledky vykonání akce jsou považované jak stavy pasivních tak i aktivních komponentů.

Transakce

Davies jako první vyvinul koncepci **Atomická Transakce**.

[Davies C.T. Data Processing Spheres of Control. *IBM System Journal*, vol.17, No. 2, pp. 179-198, 1978]

Základní Transakční systém nedovoluje použít subtransakce nebo podporu uvnitř transakce.

Vnořené Transakce – Moss J.E.B. 1981[2].

Už mnoho let se Transakce používaly abychom zajistili konzistentní přístup k databázi. Transakce seskupuje libovolný počet jednoduchých akcí (operací) a reprezentuje je jako celek, který je nedělitelný z hlediska ostatních transakcí.

Použití transakce umožňuje provedení obnovení dat, které zapojuje několik objektů, takže není nutné si dělat starosti o souběžnosti.

Jestliže se stane něco neočekávaného, co zabraňuje operaci pokračovat v průběhu provedení transakce, může být tato Aborted (předčasně ukončena), což zruší všechny změny, které byly provedeny v zájmu transakce.

Schopnost transakce ukrýt efekt souběžnosti a zároveň působit jako firewalls pro selhání, je dělají vhodnými budujícími bloky pro strukturování spolehlivých distribuovaných systémů.

Je vyvinuto mnoho typů jednotek pro sestavení struktury konkurenčních systémů. Tyto jednotky mohou být tříděné, mohou odpovídat různé úrovni systémů a jsou určeny pro různé oblasti použití.

První schémata byla vyvinuta pro sdílené prostředky v multi-uživatelských OS a DB.

Jedna z prvních metod pro strukturování konkurenčních systémů byla metoda založená na Atomické Transakci.

Tato metoda popisuje obecná pravidla pro strukturování konkurenčních aplikací a pro uvedení atomických jednotek v programovacím rozhraní.

Atomické transakce: [6, 7, 8, 9]

Čtyři vlastnosti (**AKIT vlastnosti**) strukturálních jednotek:

- Atomičnost;
- Konzistentnost;
- Izolovanost;
- Trvanlivost.

Konzistentnost znamená, že vykonání kterékoli transakce představuje správnou transformaci stavů pasivních komponentů a nenarušuje jejich integritu.

Izolovanost (serializovanost) - vlastnost, která hraje hlavní roli v zajištění konkurenční schopnosti: při navrhování komponentu, který bude používat nějaké prostředky, návrhář nemusí znát jiné komponenty soupeřící o stejné prostředky. Tato konkurence je skrytá i neovlivňuje jakýmkoliv způsobem návrh komponentu.

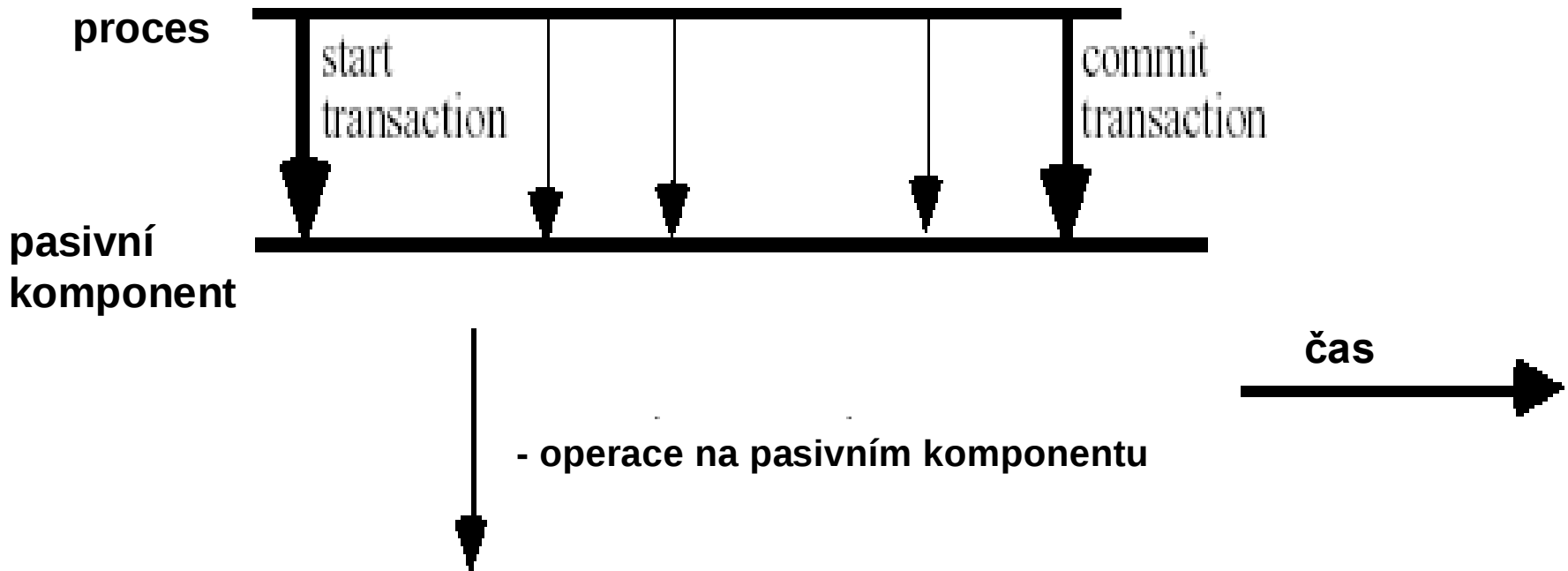
Je garantováno, že se Transakce nebudou ovlivňovat navzájem a obnovení kterékoli z nich bude oddělené od vykonávání ostatních Transakcí dokonce i když jsou tyto transakce vykonávány současně.

Trvanlivost- schopnost stavů pasivních komponentů přezít jakékoliv selhání HW (po úspěšném ukončení transakce).

Atomická transakce spoléhá na tři standardní operace:

- Start;
- Abort (předčasné ukončení);
- Commit (uložit do paměti).

Operace Start a Commit markují (označují) hranice Atomické Transakce.



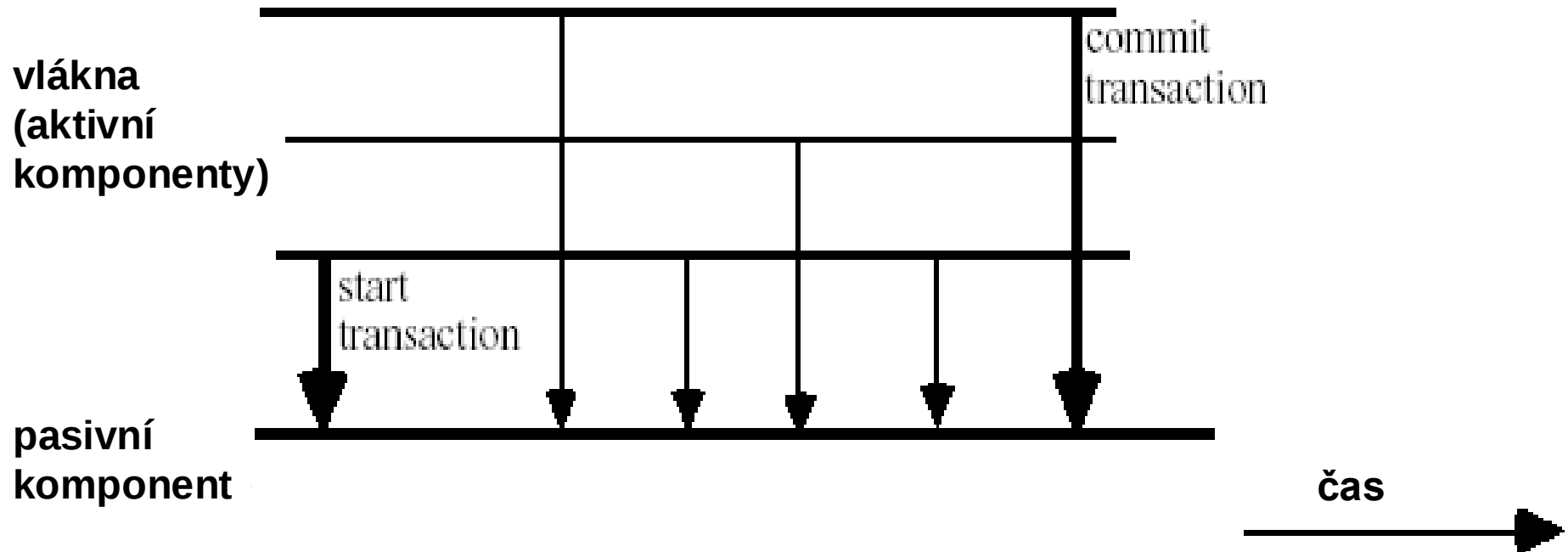
■ **Atomická transakce** začíná aktivním komponentem, který pak provádí několik operací na pasivním komponentu a commit transakce (ukončuje transakce uložení stavu pasivního komponentu).

■ Všechny pasivní komponenty mohou být přístupné několika transakcemi současně ale AKIT vlastnosti těchto transakcí jsou garantovány Podporou Transakce.

Multi-vláknové transakce [10]

MvT dovolují několika aktivním komponentům (vláknům) zúčastnit se stejné (téže) transakce a provádět operace na stejných objektech (pasivních komponentech) společně.

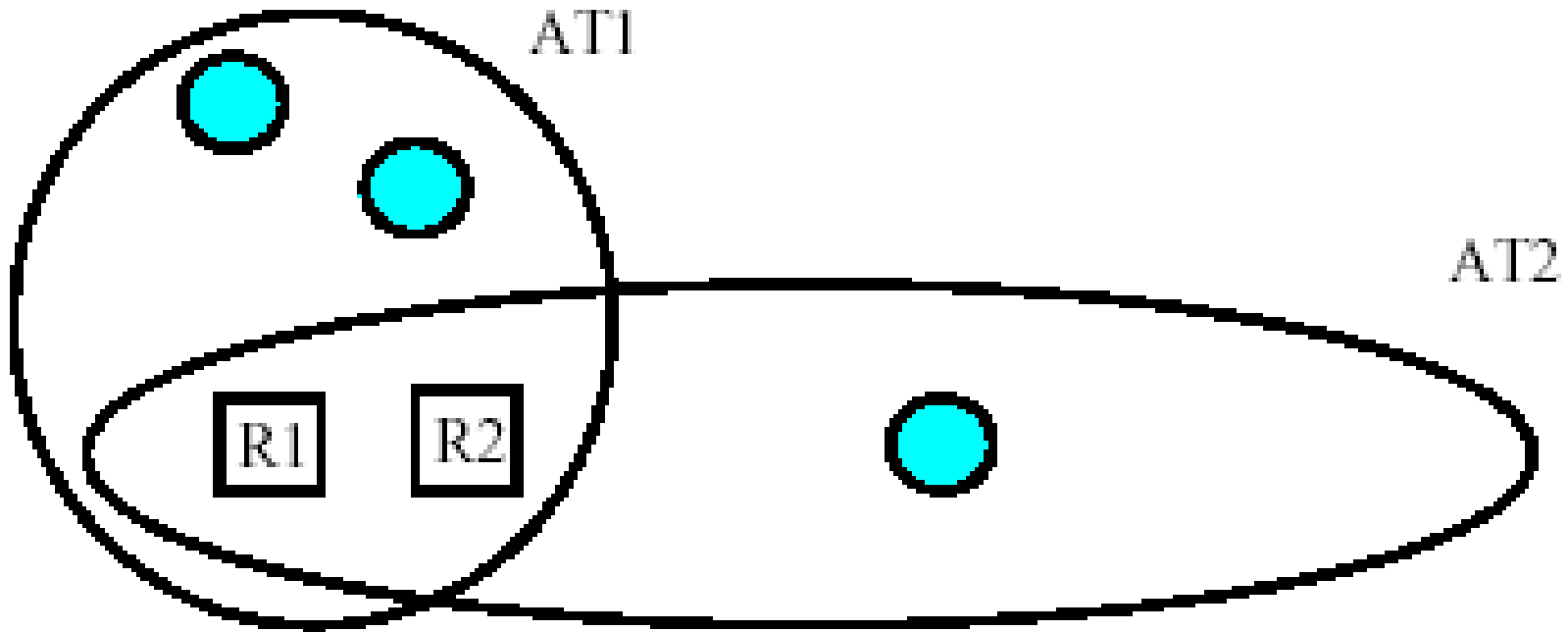
- Účastníci stejné transakce nespolupracují jakýmkoliv způsobem !
- Vykonávání vláken není synchronizované v rámci (v mezích) MvT.
(Pro toto schéma obnovení znamená Abort transakci).



MvT začíná aktivní komponentem; pak několik aktivních komponentů provádí operace na pasivních komponentech a jeden z nich commit (ukončuje) transakci

Atomické transakce jsou určeny pro zajištění odolnosti pouze proti HW závadám.

- OPZ je zajištěná prostřednictvím odrolování stavů objektů (pasivních komponentů).
- Atomická transakce nezačleňuje žádné rysy
 - pro zajištění odolnosti proti SW závadám nebo závadám prostředí;
 - pro zajištění obnovení aktivních komponentů.



Dvě atomické transakce, AT1 a AT2. AT1 má dva aktivní komponenty a AT2 má jeden komponent

- Výsledky provedení AT jsou reprezentované stavy pouze pasivních komponentů.
- AT garantuje, že soupeřící komponenty (tj. komponenty z různých transakcí) budou vykonávané tak jako by byly disjunktní (rozpojené).

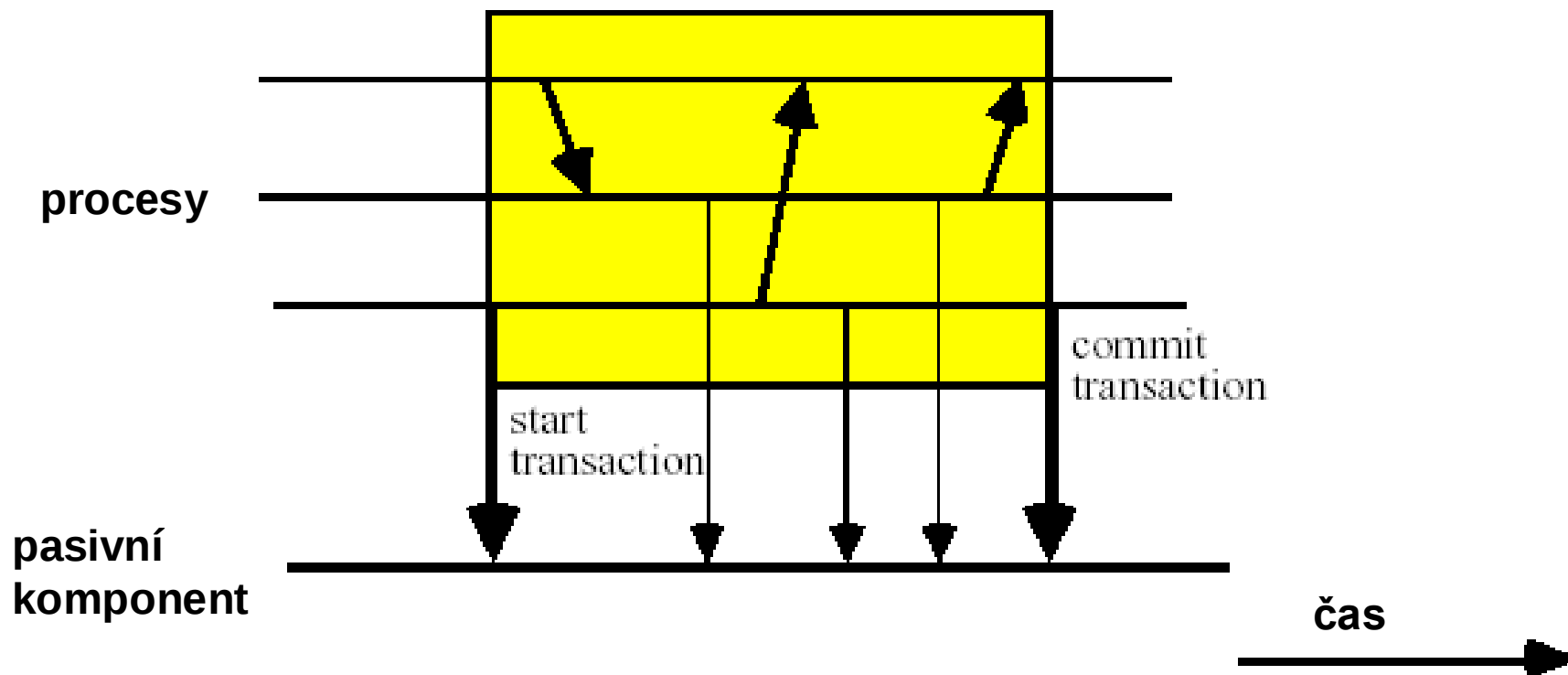
Metody slučující kooperační a konkurenční souběžnost 16

Rozšířené konverzace [13], [14], [15]

Systém sestává z:

- (takzvaných “konverzačních”) kooperačních komponentů, které synchronizují své obnovení (tj. vrácení ke kontrolním bodům) pomocí mechanismu konverzačního typu.
- serverových komponentů, které jsou podpořené mechanismem Atomické Transakce.

EC1



Rozšířená konverzace EC1 s třemi účastníky

Metody slučující kooperační a konkurenční souběžnost

17

Rozšířené konverzace:

Serverové komponenty přijímají požadavky (ve formě zpráv nebo dálkového volání procedury) a provádějí transakční operace START, ABORT a COMMIT.

■ Tyto systémy jsou strukturované rekurzivně použitím vnořených konverzací. Vnořené konverzace mohou mít přístup k transakčním objektům (pasivním komponentům).

■ Podpora rozšířené konverzace:

- podpora garantuje, že konverzace budou správně manipulovat pasivními objekty: operací starts, abort a commit (provedené transparentně pro programátory kooperačních systémů) dodržují striktní pravidla, která garantují správné chování kombinovaného systému;
- podpora zajišťuje koordinované obnovení (Odrolování) všech komponentů zapojených do rozšířené konverzace.

■ RK zajišťuje odolnost proti SW závadám spolupracujících komponentů.

■ RK garantuje, že konverzace správně manipulují transakčními objekty.

Závěr:

RK = Konverzace + Atomická Transakce

Metody slučující kooperační a konkurenční souběžnost

18

Koordinované Atomické akce (KA akce): [12], [16]

KAA = Atomická akce + Atomická transakce

KA akce je jednotný obecný přístup k strukturování komplexní souběžné aktivity. Schéma umožňuje také obnovení četných interaktivních objektů v distribuovaných OO systémech.

■ KA akce poskytuje konceptuální rámec pro zacházení jak s různými typy souběžnosti (kooperační a konkurenční) tak i s prostředky zajišťujícími odolnost proti závadám. KAA rozšiřuje a integruje dva komplementární koncepty - Atomická akce a Transakce.

KA akce mají vlastnosti obou:

Atomické akce se používají pro řízení kooperační souběžnosti a pro implementace koordinovaného obnovení komponentů.

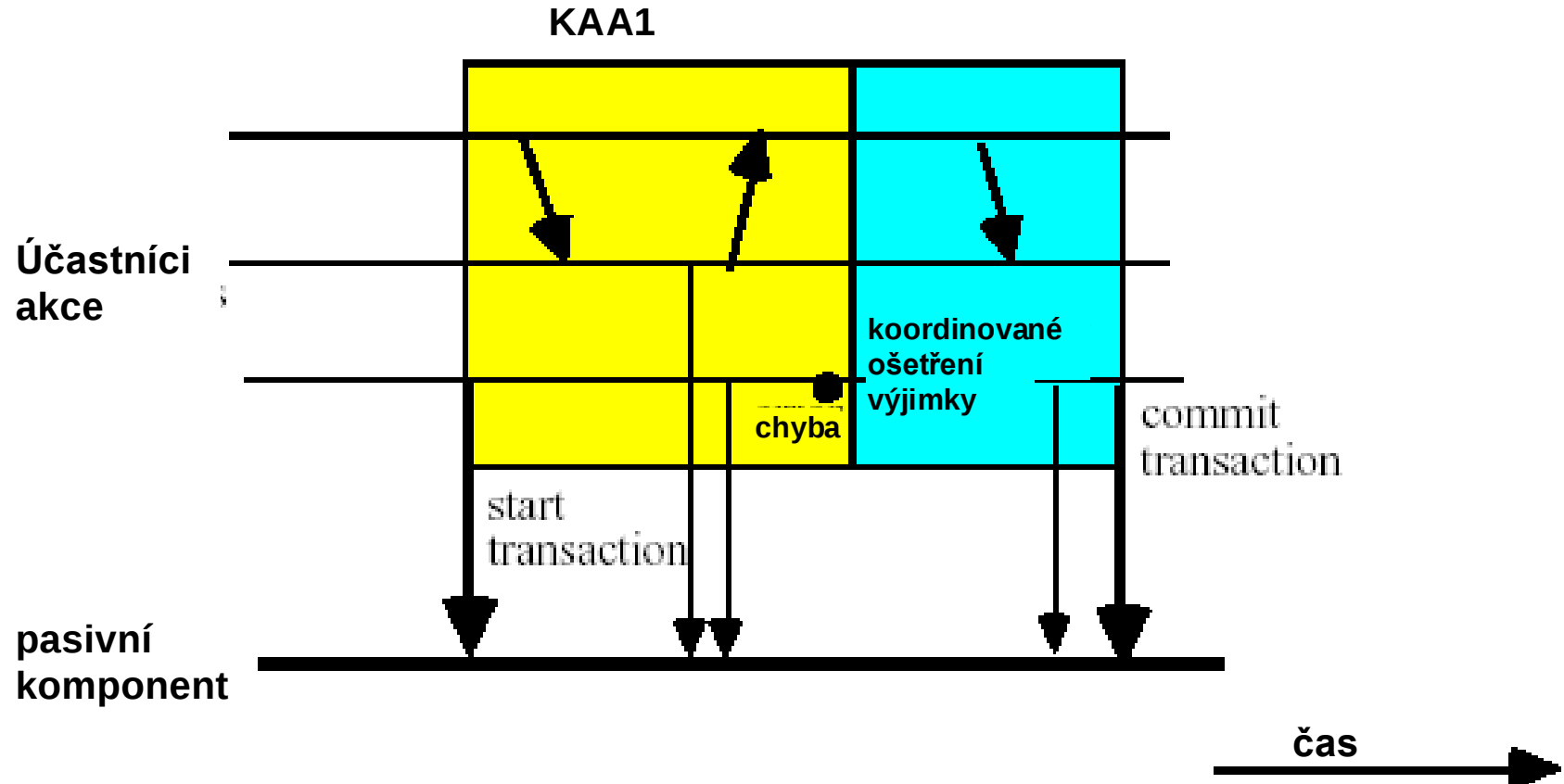
Transakce se používají pro udržování konzistence sdílených prostředků v přítomnosti selhání a konkurenční souběžnosti.

■ Každá KAA je navržena jako stylizovaná Multi-vstupní procedura s rolemi, které mohou být aktivované účastníky KAA (aktivními komponenty) a které mohou kooperovat uvnitř KAA.

Metody slučující kooperální a konkurenční souběžnost

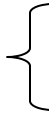
19

Koordinovaná Atomická akce



Koordinovaná Atomická akce CAA1 se třemi účastníky, kteří provádějí koordinované obnovení (přerolování)

KA akce může být dokončená:

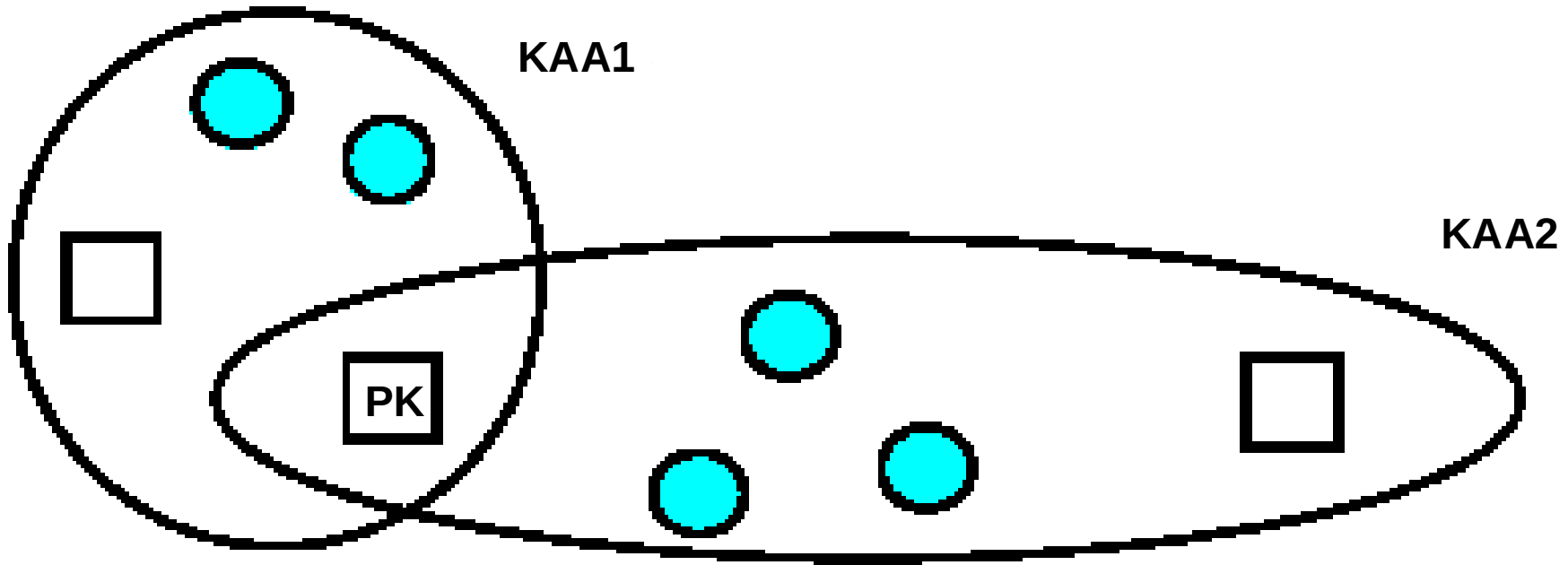
- **Normálně** 
 - buď když žádná chyba nevznikla (chyba nebyla odhalena)
 - nebo po úspěšném obnovení
- **Abnormálně (výjimečně)** – když výjimka „selhání“ byla šířená do obalující akce

■ **Externí (transakční) objekty (tj. pasivní komponenty) mohou být použité souběžně několika KAA takovým způsobem, že:**

- informace nemůže být pašovaná mezi nimi (tj. mezi KA akcemi)
- kterákoli posloupnost operací na těchto objektech, která je umístěná uvnitř závorek KAA (start – commit), má AKIT vlastnosti – vzhledem k ostatním posloupnostem.

■ **KA akce vypadá jako Atomická transakce pro ostatní prostředí.**

■ **Stav KA akce je reprezentován množinou stavů lokálních a externích objektů.**



Dvě KA akcí, KAA1 a KAA2, jsou sestavené z kooperačních komponentů a soupeří o pasivní komponent PK

- Účastníci těchto jednotek spolupracují přes pasivní komponenty, které patří výhradně jedné z těchto jednotek.
- V KAA existují dva typy pasivních komponentů:
 - Lokální (používá se pro spolupráci účastníků);
 - Sdílené několika jednotkami (mají AKIT vlastnosti).
- Výsledky vykonávání těchto jednotek jsou reprezentované stavy jak aktivních tak i pasivních komponentů.

metoda	souběžnost	závady	obnovení
konverzace (1976)	kooperační	závady návrhu	odrolování (rozmanitost SW)
atomické akce (1980)	kooperační	závady prostředí, závady návrhu	přerolování (výjimky) a odrolování (rozmanitost SW)
atomické transakce (mid-60ies)	konkurenční	závady HW	odrolování (opakování)
rozšířené konverzace (1991)	kooperační a konkurenční	závady návrhu	odrolování (rozmanitost SW)
koordinované atomické akce (1995)	kooperační a konkurenční	závady prostředí, závady návrhu, závady HW	přerolování (výjimky) a odrolování (rozmanitost SW, opakování)

Porovnání různých dynamických metod strukturování SW systémů

Atomické akce mohou být aplikované pro strukturování: [6]

- systémů pracujících v reálném čase;
 - řízení procesů;
 - telefonických spínacích systémů;
 - elektronického vybavení letadla
- a pro zajištění jejich odolnosti proti závadám.

■ Bylo navrženo několik úspěšných aplikací pro **Konverzace** a **Atomické akce**, které byly implementované v praktických systémech, v ukázkových příkladech a reálných příkladech:

- námořní příkazový a řídicí systém; [18]
- zjednodušený dopravní řídicí systém; [19]
- model antiraketového obranného C3 systému; [20]

Vlastnosti **Atomické transakce** byly vnesené do mnohých programovacích jazyků, do distribuovaných a OS, atd.[7,8]

Například, všechny systémy řízení DB se hodně spoléhají na Atomické Transakce.

- ❑ V dnešní době Atomická transakce (ATr) se používá více než Atomická akce:
- programátoři si zvykli, že je lepší pracovat s programem když mají pocit, že všechny prostředky jsou v jejich vlastní dispozici;
 - podporující SW (včetně OS a ATr) garantuje, že programátor při vyvinutí SW bude pracovat ve vlastním prostředí a nemusí se starat o koordinaci a kooperaci s ostatními komponenty souběžného systému;
 - hodně úsilí bylo věnováno vývoji Transakčních systémů. Je těžké změnit tento trend (směr vývoje), tj. stereotyp chování.

Nícméně použití ATr všude by mohlo vést ke zvýšení nákladů a je zavádějící

Proč potřebujeme kooperace:

- souběžné aktivity může potřebovat výměnu informace nejenom přes pasivní AKIT komponenty, ale může potřebovat synchronizaci jejich provedení a může mít společný cíl;
- je hodně důkazů, že závady SW se stávají dominujícím faktorem. Proto potřebujeme metody strukturování SW, které umožňují zajistit odolnost proti všem druhům závad.

Nové složité distribuované a souběžné systémy (Internetové aplikace, rozšířené pracovní toky, řídicí systémy, moderní telefonické spínací systémy, systémy řízení elektrických sítí , multi-robot dílny, komplexní middlewarové služby, atd.) vyžadují kooperace a metody pro strukturování, které kombinují kooperace a konkurence.