

Informační a řídicí systémy I.

Úvod do OPC Unified Architecture

Pavel Balda
ZČU v Plzni, FAV, KKY

Osnova přednášky

- n Co je OPC UA?
- n Koncepce OPC UA
- n Integrované modely a služby
 - n Model zabezpečení (stručný úvod)
- n Model adresového prostoru

2

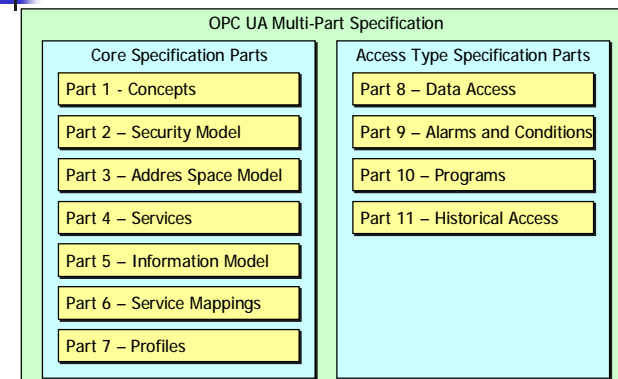
Doporučená literatura

- n [UA Part 1] OPC U. A. Spec., Part 1: **Concepts**, Version 1.00, July 28, 2006
- n [UA Part 2] OPC U. A. Spec., Part 2: **Security Model**, Version 1.00, July 28, 2006
- n [UA Part 3] OPC U. A. Spec., Part 3: **Address Space Model**, Version 1.00, July 28, 2006
- n [UA Part 4] OPC U. A. Spec., Part 4: **Services**, Version 1.01.05, February 1, 2007
- n [UA Part 5] OPC U. A. Spec., Part 5: **Information Model**, Version 1.00, July 28, 2006
- n [UA Part 6] OPC U. A. Release Candidate Spec., Part 6: **Mappings**, Version 0.93, June 1, 2006
- n [UA Part 7] OPC U. A. Draft Spec., Part 7: **Profiles**, Version 0.23, July 28, 2006
- n [UA Part 8] OPC U. A. Spec., Part 8: **Data Access**, Version 1.00, September 25, 2006
- n [UA Part 9] OPC U. A. Draft Spec., Part 9: **Alarms**, Version 0.62, March 26, 2006
- n [UA Part 10] OPC U. A. Spec., Part 10: **Programs**, Version 1.00, January 29, 2007
- n [UA Part 11] OPC U. A. Spec., Part 11: **Historical Access**, Version 1.00, January 5, 2007

- n Všechny specifikace jsou dostupné pro členy OPC foundation na: <http://www.opcfoundation.org>
- n Použité zkratky: U. A. = Unified Architecture, Spec. = Specification

3

Organizace Specifikace OPC UA



4

Co je OPC UA?

- OPC UA je platformově nezávislý standard pro komunikaci různých druhů systémů a zařízení prostřednictvím přenosu zpráv (messages) mezi klienty a servery po různých typech sítí
- OPC UA zajišťuje robustní a zabezpečenou komunikaci (proti útokům) založenou na ověření identity klientů a serverů
- OPC UA definuje standardní množiny služeb (services), které mohou být poskytovány servery a konkrétní server oznamuje klientům, jaké služby podporuje
- Informace je poskytována pomocí standardních a výrobcí definovaných datových typů
- Servery definují datový model, který mohou klienti za běhu zjišťovat (discover)
- Servery mohou poskytovat okamžitá i historická data a mohou klientům oznamovat (notify) alarmy a události
 - Událost (Event) je všeobecný termín označující výskyt „něčeho důležitého“ v systému nebo v jeho komponentě
 - Alarm je typem události, která má stav a v typickém případě vyžaduje potvrzení
- OPC UA může být mapován na různé komunikační protokoly, data mohou být různým způsobem kódována podle požadované přenositelnosti a efektivity

5

Cíle návrhu OPC UA

(1/3)

- OPC UA poskytuje konzistentní adresový prostor (AddressSpace) a model služeb (service model)
 - Adresový prostor je souhrn informací, které servery umožňují vidět klientům
 - OPC UA Server může ve svém adresovém prostoru obsahovat data, alarmy a události a historii
 - Přístup k těmto datům je pomocí integrované množiny služeb, které mj. zahrnují i model zabezpečení (security model)
- OPC UA umožňuje serverům poskytovat klientům definice typů objektů z adresového prostoru
 - Data mohou být poskytována mnoha způsoby, včetně binárních struktur a XML dokumentů
 - Formát dat může být definován OPC nebo jinými standardizačními organizacemi a výrobcí
 - Klienti se mohou dotazovat serverů prostřednictvím adresového prostoru na metadata, popisující formát konkrétních dat
 - Tímto způsobem mohou klienti, kteří neznají (nemají předprogramován) formát dat, být schopni zjišťovat formát dat za běhu a pak data správně používat

6

Cíle návrhu OPC UA

(2/3)

- OPC UA zavedlo množství vazeb mezi uzly
 - Uzel (Node) je základním stavebním kamenem adresového prostoru
 - Vazby už nejsou omezeny jen na jednu hierarchii
 - Server může data prezentovat v mnoha hierarchiích, přizpůsobených typickému nazírání klientů na daná data
 - Tato pružnost spolu s definicemi typů umožňuje OPC UA aplikovat v řadě oblastí
 - Cílem OPC UA není pouze vytvoření rozhraní pro přenos vzdálených dat (telemetry), ale poskytnutí lepší součinnosti (interoperability) funkcí ve vyšších hierarchických úrovních podniku, viz ilustrační obrázek s cílovými aplikacemi OPC UA
- OPC UA poskytuje požadovaná data robustně
 - Obsahuje mechanismy pro rychlou detekci a zotavení se z komunikačních chyb bez nutnosti čekání na vypršení dlouhých timeoutů protokolů v nižších vrstvách

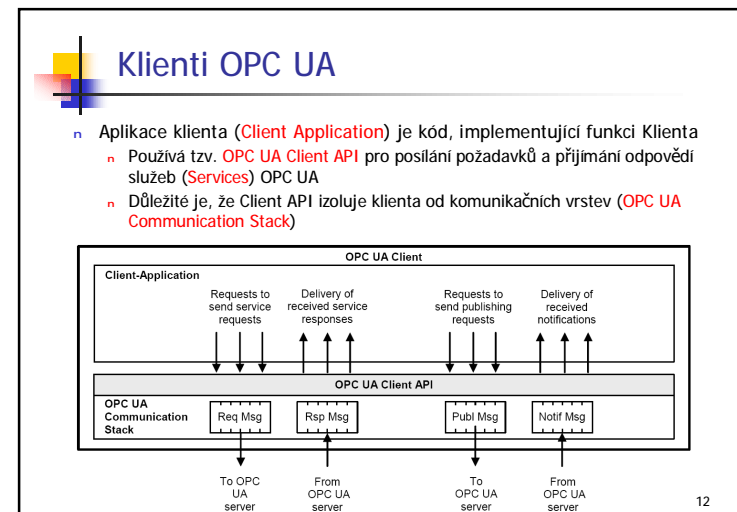
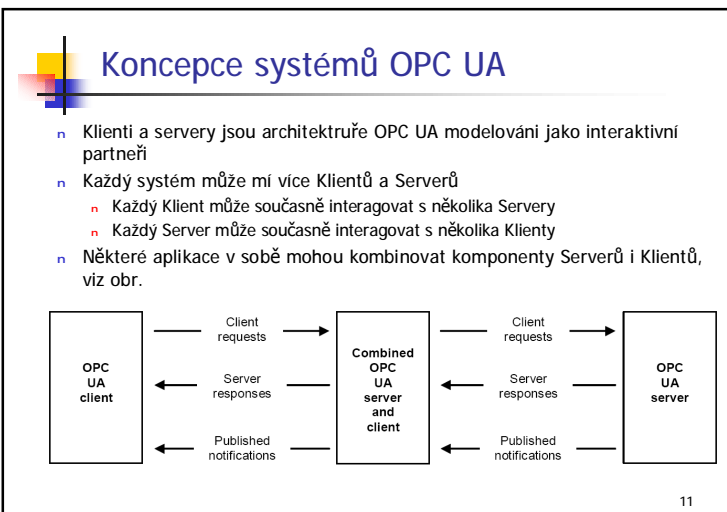
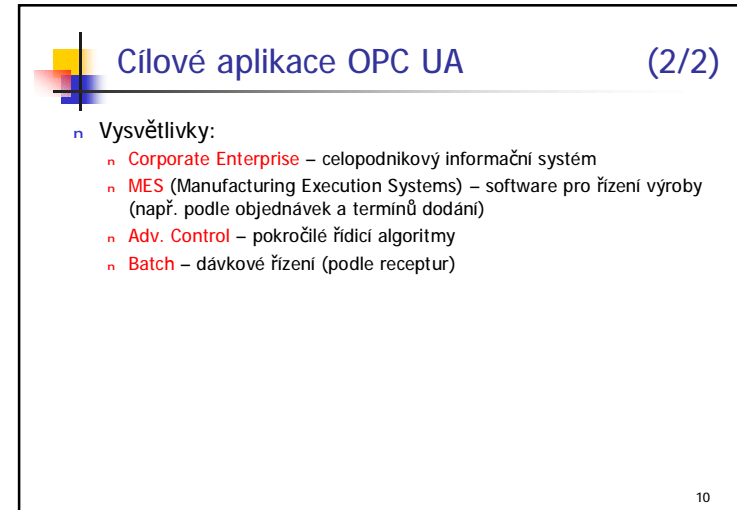
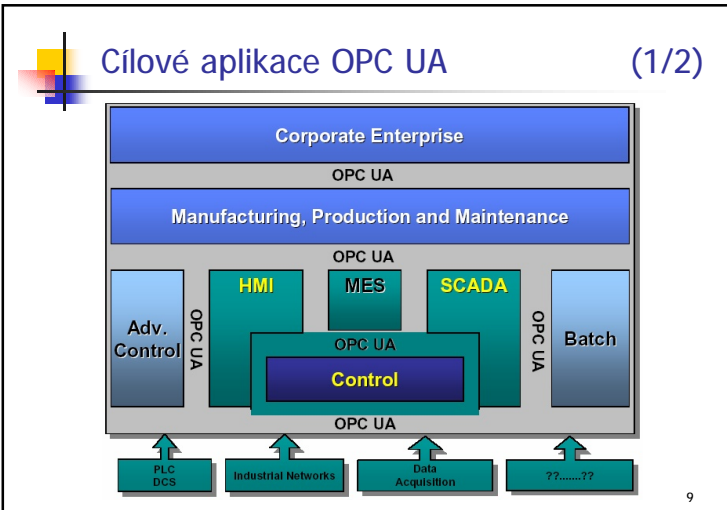
7

Cíle návrhu OPC UA

(3/3)

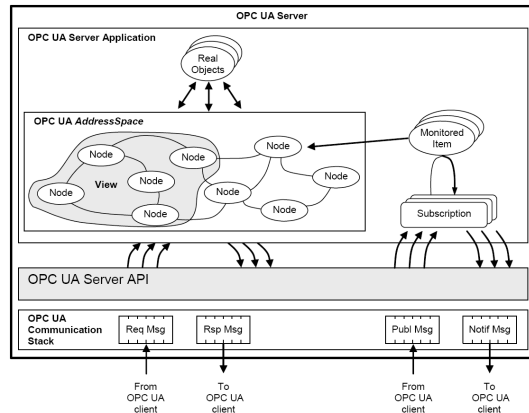
- OPC UA podporuje širokou škálu serverů, od PLC na výrobní hale až po podnikové servery
 - Velký rozsah velikosti, výkonu, výpočetních platform a funkčních schopností
 - OPC UA definuje rozsáhlou množinu schopností, servery mohou implementovat jen jejich podmnožinu
 - OPC UA definuje standardní podmnožiny – Profily [UA Part 7] – pro usnadnění vzájemné spolupráce (interoperability)
 - Servery mohou tvrdit, že vyhovují určitým profilům
 - Klienti mohou zjistit (discover) profily daného serveru a podle toho k serveru přistupovat
- OPC UA specifikace jsou „vrstevnaté“, aby izolovaly návrh (core design) od pod ním ležících výpočetních a síťových technologiemi
 - Tím může být OPC UA mapováno na budoucí technologie, aniž by musel být předělán základní návrh
 - V [UA Part 6] jsou definovány dva způsoby kódování: XML/text a UA Binary
 - A též dva způsoby přenosu: TCP a Webové služby založené na SOAP a HTTP
- OPC UA je navržen jako cesta pro migraci (migration path) pro OPC klienty a servery využívající technologii COM firmy Microsoft
 - Existující COM servery (DA, A&E, HDA) mohou být mapovány na OPC UA

8



Servery OPC UA (1/2)

(1/2)



13

Servery OPC UA (2/2)

(2/2)

- n Reálné objekty (**Real Objects**) – fyzické nebo sw objekty
 - n Mohou být dostupné z aplikace nebo jsou spravovány uvnitř
 - n Příkladem jsou fyzická zařízení
- n Serverová aplikace (**OPC UA Server Application**)
 - n Kód, implementující funkčnost serveru
 - n Používá OPC UA Server API pro posílání/přijímání zpráv klientům/od klientů.
 - n Opět izoluje server od nižší komunikační vrstvy
- n Adresový model (**OPC UA AddressSpace**) bude diskutován v další části
- n Publisher/Subscriber Entities
 - n **Monitored Items** – entity na Serveru vytvořené Klientem, které monitorují uživatelského adresového prostoru. Když dojde ke změně hodnoty nebo výskytu události/alarmu, generují „oznámení“ (**Notification**), které je předáno klientovi prostřednictvím „předplacení“ (**Subscription**)
 - n **Subscriptions** – předávají klientům notifikace (**Notifications**). Klienti řídí četnost posílání publikačních zpráv (**Publish Messages**)

14

Rozhraní služeb (Service Interface)

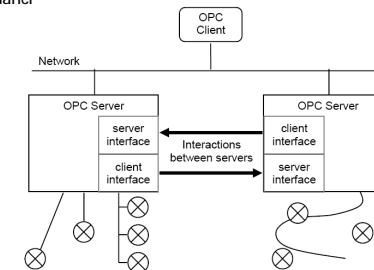
- n Služby (Services) definované OPC UA jsou popsány v [UA Part 4]
- n Služby požadavek/odpověď (**Request/Response Services**)
 - n Jsou volány Klienty pro vykonání určitého úkolu s jedním nebo více uzly (**Nodes**) v adresovém prostoru (**AddressSpace**) a vrácením odpovědi (**Response**)
 - n Volání probíhá pomocí tzv. rozhraní služeb (**OPC UA Service Interface**)
- n Publikační služby (**Publisher Services**)
 - n Služby volané pomocí rozhraní služeb za účelem periodického posílání notifikací (**Notifications**) klientům
 - n Notifikace zahrnují události (**Events**), alarmy (**Alarms**), změny dat (**data changes**) a výstupy z programů (**Program outputs**)

15

Interakce mezi Servery (1/2)

(1/2)

- n Interakce mezi servery (**Server to Server Interactions**) jsou takové interakce, kdy se jeden server chová jako klient jiného serveru.
- n Umožňují:
 - n A) Výměnu informací na stejné úrovni (peer-to-peer basis), např. při redundanci

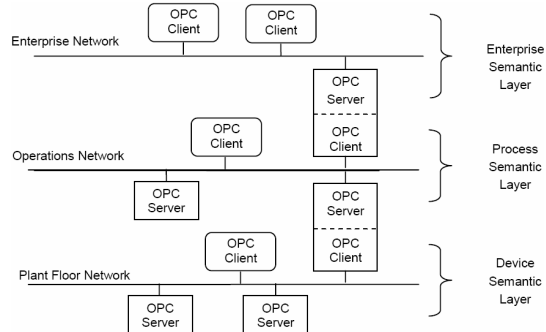


16

Interakce mezi Servery (2/2)

(2/2)

- n Druhou možností je:
 - n B) Řetězení serverů do vrstevnaté architektury



17

Integrované modely a služby

- n Model zabezpečení [UA Part 2]
- n Integrovaný model adresového prostoru [UA Part 3]
- n Integrovaný model objektu [UA Part 3]
- n Integrované služby [UA Part 4]

18

Model zabezpečení (Security model) (1/2)

(1/2)

- n Zabezpečení OPC UA zahrnuje činnosti
 - n **Autentizace** (authentication, ověření identity) klientů, serverů a uživatelů
 - n Zajištění integrity a důvěrnosti (confidentiality) komunikaci
 - n **Integrita** – proces zajišťující, že zprávy jsou přijaty tak, jak byly odeslány, tj. nebyly při přenosu zmodifikovány
 - n **Důvěrnost** – ochrana dat před jejich čtením „nepovolanými osobami“
 - n Průkaznost (verifiability) funkčnosti, která je tvrzena
- n [UA Part 2] zavádí **model zabezpečení**
 - n Lze volit a konfigurovat různá bezpečnostní opatření (security measures) pro zajištění požadavků dané instalace
 - n Model zahrnuje standardní mechanismy a parametry zabezpečení, které se používají v internetu
 - n V [UA Part 7] jsou definovány profily zabezpečení

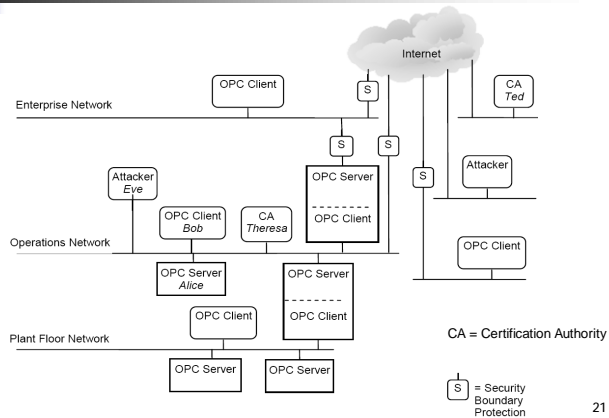
19

Model zabezpečení (Security model) (2/2)

(2/2)

- n **Navázání relace** (Session establishment)
 - n Zabezpečený komunikační kanál (secure communication channel) je na aplikační úrovni aktivní po celou dobu trvání relace
 - n Proto uživatelé mohou být autentizováni pouze jednou při navázání relace
 - n Při navázání relace se klient a server domluví na zabezpečeném komunikačním kanálu [UA Part 4, 6], vymění si své certifikáty a informace o tom, jaké mají schopnosti (capabilities)
 - n OPC Foundation generuje certifikáty pro různé Profily [UA Part 7]
- n **Auditing** (prověřování)
 - n Možnost procházení záznamů o prověřování (audit logs)
 - n V případě, že se vyskytne problém v zabezpečení detekovaný na serveru, je možné dohledat a prověřit „audit log“ příslušného klienta
 - n Servery mohou generovat „prověřovací“ události pro klienty, kteří je zpracovávají a zapisují (log).
- n **Zabezpečení přenosu** (Transport security)
 - n OPC UA vychází z infrastruktury zabezpečení webových služeb
 - n Přenášené zprávy mohou být zašifrovány (encrypt) a digitálně podepsány (sign) – zajištění důvěrnosti a integrity
 - n Pro šifrování se používají technologie z nižších komunikačních vrstev
 - n Algoritmy šifrování a podepisování jsou pro daný profil definovány v [UA Part 7]

Bezpečnostní prostředí OPC UA (1/2)



21

Bezpečnostní prostředí OPC UA (2/2)

- n OPC UA pokrývá několik úrovní podniku
 - n Od horních úrovní řízení podniku až po spodní úrovně řízení procesů a strojů
 - n Může být atraktivním cílem průmyslové špiónáže a sabotáží
 - n Z těchto důvodů musí pružně poskytovat různé zabezpečovací mechanismy
- n Používané zabezpečovací mechanismy závisí na kontextu aplikací
 - n Klienty a servery na stejném počítači lze snadněji bránit proti externímu útoku
 - n Klienti a servery na různých počítačích ve stejné pracovní síti (Operations Network) mohou být chráněni zabezpečením na hranici sítě (security boundary protections) s externími připojeními
 - n Některé aplikace OPC UA pracují v relativně otevřeném prostředí (internet), kde je zabezpečení mnohem složitější
- n [UA Part 2] dále diskutuje:
 - n Možné způsoby ohrožení systémů OPC UA
 - n Vztah k zabezpečení počítačových sítí a webových aplikací
 - n Architekturu zabezpečení

22

Model adresového prostoru

- n [UA Part 3] – **Address Space Model**
 - n Popisuje adresový prostor a jeho objekty
- n **AddressSpace (Adresový prostor)** = souhrn (collection) informací, které **Server** OPC UA zviditelňuje (makes visible) **Klientům**

23

Základní pojmy adresového prostoru (1/3)

- n **Node** (uzel) = základní komponenta adresového prostoru
- n **NodeClass** = třída uzlu Node v adresovém prostoru AddressSpace
 - n Třídy NodeClass definují metadata komponent objektového modelu OPC UA
- n **Attribute** (atribut) = primitivní (základní) charakteristika uzlu
 - n Všechny atributy jsou definovány OPC UA a nesmějí být definovány Klienty ani Servery
 - n Atributy jsou jediné prvky adresového prostoru, kterým je dovoleno mít datové hodnoty (data values)
- n **Object** = uzel (Node), reprezentující fyzický nebo abstraktní prvek systému
 - n Objekty jsou modelovány užitím tzv. objektového modelu OPC UA
 - n Příklady objektů: systém, subsystém, zařízení, ...
 - n Objekt lze definovat jako instanci typu objektu (ObjectType)
 - n Objekt je též synonymum pro instanci objektu (ObjectInstance)
- n **ObjectType** = uzel (Node), reprezentující definici typu objektu (Object)

24

Základní pojmy adresového prostoru (2/3)

- n **Reference** (reference, odkaz) = explicitní vztah (pojmenovaný ukazatel) jednoho uzlu (Node) k druhému
 - n Uzel, obsahující referenci je zdrojový uzel (**SourceNode**)
 - n Uzel, na nějž je odkazováno je cílový uzel (**TargetNode**)
 - n Všechny odkazy jsou definovány pomocí typů odkazu (**ReferenceType**)
- n **ReferenceType** (typ reference, odkazu) = uzel, reprezentující definici typu odkazu (Reference)
 - n Typ odkazu (Reference Type) určuje význam (sémantiku) odkazu (Reference)
 - n Jméno tohoto typu vyjadřuje v jakém vztahu jsou zdrojové a cílové uzly a obecně odráží operaci mezi nimi, např. „A obsahuje B“
- n **Variable** (proměnná) = uzel (Node), který obsahuje nějakou hodnotu
- n **VariableType** = uzel, reprezentující definici typu proměnné (Variable)
- n **Complex Data** = data, složená z prvků více než jednoho primitivního datového typu, např. struktura
- n **Property** (vlastnost) = proměnná, která je cílovým uzlem (TargetNode) pro odkaz **HasProperty**
 - n Vlastnosti popisují charakteristiky uzlu (Node)

25

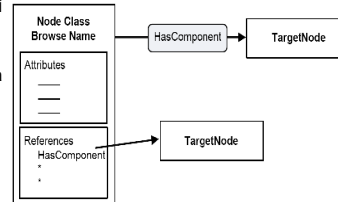
Základní pojmy adresového prostoru (3/3)

- n **Event** (událost) = všeobecný termín označující výskyt „něčeho důležitého“ v systému nebo v jeho komponentě
- n **EventType** = uzel typu **ObjectType**, reprezentující definici události (Event)
- n **EventNotifier** = speciální atribut (Attribute) uzlu, umožňující, aby se klient zaregistroval (subscribe to) k tomuto uzlu pro přijímání oznámení (**Notifications**) o výskytech události (**Event**)
- n **Notification** (notifikace) = všeobecný termín pro data, oznamující detekci události (Event) nebo změny hodnoty atributu (Attribute)
- n **Message** = datová jednotka přenášená mezi Klientem a Serverem, reprezentující požadavek (request) nebo odpověď (response) určité služby (Service)
- n **Service** = Operace v Serveru OPC UA, která může být volána Klientem [UA Part 4]
- n **Subscription** = koncový bod v Serveru definovaný Klientem, který se užívá pro vrácení notifikací serveru. Obecný termín popisující množinu uzlů zvolených klientem pro které:
 - n Server periodicky sleduje splnění určité podmínky
 - n Server posílá notifikace klientovi, když je daná podmínka splněna

26

Konvence pro schémata adresového prostoru

- n Uzly (**Nodes**) jsou reprezentovány **obdélníky**. Mohou mít záhlaví složené z:
 - n 1 řádku (**BrowseName**) nebo
 - n 2 řádků (1. řádek – NodeClass, 2. řádek – BrowseName)
- n V těchto obdélnících mohou být vloženy další dva obdélníky definující atributy (**Attributes**) a odkazy (**References**)
 - n Konkrétní jména atributů a referencí se uvádějí přímo do těchto obdélníků
- n Odkazy (**References**) se znázorňují
 - n Buď obdélníky se zakulacenými rohy a šipkami jimi procházejícími
 - n Nebo šipkami vedoucími od jména odkazu
 - n V obou případech vede šipka od zdrojového uzlu (**SourceNode**) k cílovému (**TargetNode**)



27

Konvence pro definování tříd uzlů

- n Pro definici tříd uzlů (NodeClasses) v kap. 5 [UA Part 3] se používají tabulky:

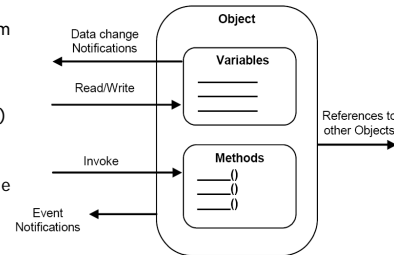
Jméno	Užití	Datový typ	Popis
Attributes			
"Jméno atributu"	"M" nebo "O"	Typ dat atributu	Definuje atribut (Attribute)
References			
"Jméno odkazu"	"1", "0..1" nebo "0..*"	Není použit	Definuje použití odkazu třídou uzlu (NodeClass)
Standard Properties			
"Jméno vlastnosti"	"M" nebo "O"	Typ dat vlastnosti	Definuje vlastnost (Property)

- n Sloupec „Užití“
 - n "M" značí **Mandatory** (povinné), "O" je **Optional** (volitelné)
 - n "0..*" – nejsou žádná omezení na počet, reference nemusí uvedena; "0..1" – reference je uvedena nejvýše jednou; "1" – reference musí být uvedena právě jednou

28

Objektový model (Object model)

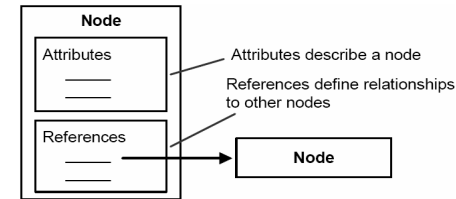
- Objektový model OPC UA slouží pro zpřístupnění definic objektů adresového prostoru serveru klientům
- Objekty (Objects) jsou definovány pomocí proměnných (Variables) a metod (Methods)
- Objektový model umožňuje vyjadřovat též vztahy k jiným objektům
- Prvky objektového modelu jsou v adresovém prostoru reprezentovány uzly (Nodes)
- Každý uzel odpovídá nějaké třídě uzlu (NodeClass)
- Každá třída uzlu reprezentuje samostatný prvek objektového modelu



29

Uzlový model (Node Model)

- Objekty a jejich komponenty jsou v adresovém prostoru (AddressSpace) reprezentovány množinou uzlů (Nodes) propojených odkazy (References)
- Atributy (Attributes) popisují uzel
- Odkazy (References) definují vazby na ostatní uzly



30

Třídy uzlů (NodeClasses)

- Jsou definovány pomocí atributů (Attributes) a odkazů (References), kterým musí být nastavena hodnota (musí být instancovány) při definici uzlu v adresovém prostoru (AddressSpace)
- Standardní třídy uzlů pro adresový prostor OPC UA jsou definovány v kap. 5 [UA Part 3]
- Společně se těmito třídám říká **metadata** adresového prostoru
- Každý uzel (Node) adresového prostoru je instancí jedné z těchto tříd uzlů (NodeClasses)
- K definici uzlů nesmí být užitá žádná jiná třída.
- Klientům ani Serverům není dovoleno definovat třídy uzlů (NodeClasses) ani rozšiřovat definice těchto tříd.

31

Atributy (Attributes)

- Datové prvky popisující uzly
- Klienti mohou přistupovat k hodnotám atributů pomocí služeb (Services) **Read**, **Write**, **Query** a **Subscription/MonitoredItem**, viz [UA Part 4]
- Atributy jsou elementární komponenty tříd uzlů, jejich definice je zahrnuta do definice třídy uzlu (viz kap. 5), a proto nejsou samostatně začleněny do adresového prostoru
- Definice každého atributu se skládá z:
 - Celočíselného identifikačního kódu (**id**), viz [UA Part 6],
 - Jména (**name**),
 - Popisu (**description**),
 - Datového typu (**data type**),
 - Povinného/volitelného indikátoru (**indicator**)
- Klienti ani Servery nesmí rozšiřovat množinu atributů definovaných pro danou třídu uzlu (NodeClass)
- ??? Indikátor ???

32

Odkazy (References)

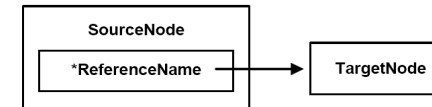
(1/2)

- n Používají se k vyjádření vztahu mezi jednotlivými uzly (Nodes)
- n Mohou být získány pomocí služeb prohlížení (browsing Services) a dotazování (querying Services) z [UA Part 4]
- n Podobně jako atributy jsou odkazy základními komponentami uzlů (Nodes)
- n Jsou však definovány jako instance uzlů odkazového typu (**ReferenceType Nodes**)
- n Uzly typu **ReferenceType** jsou viditelné v adresovém prostoru (**AddressSpace**) a jsou definovány pomocí tříd uzlů odkazového typu (**ReferenceType NodeClass**), viz kap. 5.3
- n Uzel (**Node**) obsahující odkaz (**Reference**) se nazývá zdrojový uzel (**SourceNode**)
- n Uzel (**Node**), na který je odkazováno se nazývá cílový uzel (**TargetNode**)
- n Kombinace zdrojového uzlu (**SourceNode**), typu odkazu (**ReferenceType**) a cílového uzlu (**TargetNode**) se používá ve službách (Services) OPC UA k jednoznačné identifikaci odkazů (References). Proto smí každý uzel odkazovat na jiný uzel pomocí stejného typu odkazu nejvýše jednou, viz obr. na další straně

33

Odkazy (References)

(2/2)



* Name of the Reference's ReferenceType

- n Cílový uzel (**TargetNode**) může být ve stejném adresovém prostoru (**AddressSpace**) jako zdrojový uzel (**SourceNode**) nebo v adresovém prostoru jiného serveru OPC UA
- n Cílové uzly v jiných serverech jsou identifikovány pro služby (Services) pomocí kombinace jména vzdáleného serveru a identifikátoru uzlu v tomto vzdáleném serveru
- n Jméno serveru je URI (Uniform Resource Identifier) identifikující daný server v síti. Jméno serveru je jednoznačné v dané síti, je-li síť Internet, je jednoznačné globálně
- n OPC UA nevyžaduje, aby cílový uzel existoval, takže **odkazy mohou ukazovat na neexistující uzly**

34

Proměnné (Variables)

- n Proměnné se používají pro reprezentaci hodnot (values). Existují 2 typy
- n Vlastnosti (**Properties**)
 - n Serverem definované charakteristiky objektů (Objects), datových proměnných (**DataVariables**) a jiných uzlů (Nodes)
 - n Liší se od atributů (Attributes) – charakterizují „co“ uzel reprezentuje (např. nějaké zařízení nebo objednávku) na rozdíl od metadat, užívaných pro vytváření instancí uzlů (Nodes) z tříd uzlů (**NodeClasses**)
 - n Vlastnosti (Properties) nemohou mít další vlastnosti (Properties) – zabrání rekurze
- n Datové proměnné (**DataVariables**)
 - n Reprezentují obsah objektu (Object)
 - n Např. funkční bloky v RS mohou být reprezentovány objekty. Parametry funkčního bloku, např. požadované hodnoty mohou být reprezentovány datovými proměnnými. Objekt funkčního bloku může mít i vlastnosti (Properties), např. dobu provádění nebo svůj typ
 - n Datové proměnné (**DataVariables**) mohou mít další datové proměnné – jen pokud jsou komplexní. Pak jsou prvky definic odpovídajícího komplexního typu.

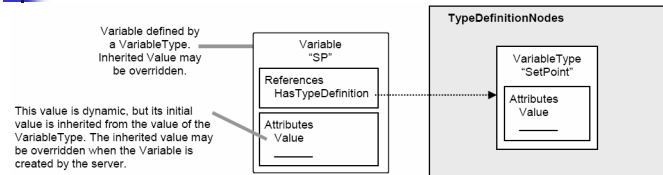
35

Uzly definující typ (TypeDefinitionNodes)

- n OPC UA vyžaduje, aby servery poskytovaly definice typů pro objekty (Objects) a proměnné (Variables)
- n **HasTypeDefinition Reference**
 - n Používá se pro spojení instance s její definicí typu reprezentovanou uzlem **TypeDefinitionNode**
 - n Tento odkaz je **povinný** – OPC UA vyžaduje definice typů
 - n [UA Part 5] (Information Model) definuje základní typy **BaseObjectType**, **PropertyType** a **BaseDataVariableType**, které může server využít
- n Všeobecně známé (well-known) **NodeId** (používá se v odkazu **HasTypeDefinition**)
 - n Některé organizace mohou definovat **TypeDefinitionNodes**, které se stanou všeobecně známými
 - n V těchto případech mohou někteří klienti interpretovat **TypeDefinitionNodes** aniz by ho museli přičíst ze serveru. Server by je tedy nemusel zveřejňovat
 - n Avšak **TypeDefinitionNodes** musí být poskytnuty pro obecné klienty. To lze řešit tím, že **TypeDefinitionNodes** budou existovat na jiném serveru

36

Příklad proměnné definované svým typem



- Parametr setpoint „SP“ je reprezentován jako datová proměnná (DataVariable) v adresovém prostoru (AddressSpace).
 - Tato proměnná je součástí objektu, který není na obrázku znázorněn
- Pro obecnou definici použitelnou jinými objekty (Objects) se používá speciální typ proměnné (VariableType)
 - Každá datová proměnná (DataVariable) vyjadřující požadovanou hodnotu a používající tuto definici bude mít odkaz **HasTypeDefinition** nastaven na obecný typ VariableType se jménem „SetPoint“

37

Model událostí (Event Model)

- Model událostí (**Event Model**) definuje událostní systém pro obecné účely, který může být použit v mnoha oblastech aplikací
- Události (**Events**) reprezentují pomíjivé „výskyty“ (occurrences)
 - Příklady: změny konfigurace systému nebo výskyt systémové chyby
 - Event Notifications** informují o výskytu události
 - Události definované v OPC UA nejsou přímo viditelné v adresovém prostoru
 - K „předplacením“ (Subscription) mohou být použity objekty (**Objects**) a **Views**. O tom, zda je předplacení na daném uzlu (Node) možné rozhoduje atribut **EventNotifier**
- K předplacení událostí se užívají standardní služby **Monitoring and Subscription Services** definované v [UA Part 4]
- Jakýkoliv UA server, který podporuje události musí vystavit alespoň jeden uzel jako **EventNotifier**
 - K tomuto účelu se používá objekt serveru (server Object) definovaný v [UA Part 5]
 - Pomocí tohoto standardního serveru jsou k dispozici všechny události jím generované

38

Standardní třídy uzlů (NodeClasses)

- Uzly (**Nodes**) tvořící adresový prostor (**AddressSpace**) jsou definovány pomocí tříd uzlů (**NodeClasses**), které jsou standardizovány v kap. 5 [UA Part 3]
- NodeClasses** jsou odvozeny od obecné třídy **Base NodeClass**
- Definované třídy **NodeClasses** spadají do tří kategorií:
 - Třídy užívané pro definice instancí
 - Třídy užívané pro definice typů instancí
 - Třídy užívané pro definice datových typů
- My se omezíme jen na představení třídy **Base NodeClass**

39

Třída Base NodeClass

Jméno	Užití	Datový typ	Popis
Attributes			
NodeId	M	NodeId	Jednoznačný identifikátor třídy uzlu, kap. 7.2
NodeClass	M	NodeClass	Třída uzlu
BrowseName	M	QualifiedName	Nelokalizované lidsky čitelné jméno, více viz kap. 7.3
DisplayName	M	LocalizedText	Lokalizované jméno uzlu, používané klienty pro zobrazení jména uzlu uživateli. Servery mohou udržovat jedno nebo více jmen pro různá lokální nastavení (locale)
Description	O	LocalizedText	Nepovinný lokalizovaný popis
References			Pro tuto třídu uzlu nejsou specifikovány žádné odkazy

40



Datový typ NodeId (1/2)

(1/2)

Jméno	Typ	Popis
NodeId	structure	
namespaceIndex	UInt32	Index URI jmenného prostoru
identifierType	Enum	Formát a datový typ identifikátoru
identifier	*	Identifikátor uzlu (Node) v adresovém prostoru (AddressSpace) serveru UA

n namespaceIndex

- n Jmenný prostor je URI, identifikující autoritu (naming authority), zodpovědnou za přiřazení prvku identifier struktury NodeId.
- n Tato URI jsou „očíslována“ aby služby (Services) mohly efektivně přenášet a zpracovávat data
 - n Hodnota 0 je rezervována pro <http://opcfoundation.org/ua/>
 - n Hodnota 1 je rezervována pro lokální server

41



Datový typ NodeId (2/2)

(2/2)

n identifierType

Hodnota	Popis
NUMERIC	Číselná hodnota
STRING	Řetězcová hodnota
URI	Formát URI. Podpora tohoto typu je vyžadována specifikací
GUID	Globally Unique Identifier
OPAQUE	Specifický (nejasný) formát jmenného prostoru

n identifier

- n Hodnota tohoto prvku se používá v kontextu s hodnotami předchozích prvků identifikujících uzly
- n Typ je určen prvkem identifierType

42