

## 7 Ciclo de Vida das Aplicações NCL

Como discutido no Capítulo 5, os comandos de edição NCL permitem maior dinamismo na execução das aplicações através de edições realizadas sobre as especificações das aplicações em tempo de execução (ao vivo). Além de poder modificar as aplicações nos receptores, preservando a sintaxe de autoria definida junto aos provedores, por meio desses comandos as aplicações podem também ser iniciadas, finalizadas, pausadas e retomadas.

Durante uma transmissão, a especificação da aplicação e os conteúdos das mídias por ela referenciados podem ser recebidos em momentos distintos ao longo da programação da emissora. Mesmo após o recebimento da especificação da aplicação (contendo todas as relações de sincronismo entre seus objetos de mídia) e dos conteúdos que fazem parte da apresentação, o início de uma aplicação no Ginga-NCL somente ocorre após o recebimento de um comando para iniciar a apresentação (*startDocument*).

Na arquitetura apresentada na Seção 3.1 o módulo Formatador é o responsável por receber e controlar as aplicações NCL. Ao receber uma aplicação NCL, o Formatador solicita a tradução das especificações em estruturas de dados adequadas à apresentação das aplicações. Essa tradução é realizada pelo módulo Conversor.

As estruturas geradas a partir da conversão da NCL são voltadas para o controle da apresentação e incluem tanto entidades abstratas, como monitores de eventos das âncoras temporais existentes nas aplicações, quanto entidades conceituais do modelo hipermídia no qual a NCL é baseada (Soares, 2005). Os eventos associados às âncoras temporais de objetos com conteúdos recebidos pelo carrossel de objetos são facilmente dedutíveis pela duração e o tempo de início dos objetos, conhecidos a priori. Os eventos associados às âncoras temporais de um objeto cujo conteúdo é recebido através de um fluxo elementar só podem ter seus tempos deduzidos a partir de bases temporais, como mencionado anteriormente na Seção 2.4.3 e discutido no Capítulo 6.

As estruturas geradas pelo módulo Conversor somente estarão disponíveis ao módulo Formataador após o comando *addDocument* ser recebido pelo módulo Gerente de Bases Privadas. Uma base privada corresponde ao local onde as estruturas relativas às aplicações permanecem à disposição do módulo Formataador. Como mais de uma base privada pode ser controlada simultaneamente, essa tarefa cabe ao módulo Gerente de Bases Privadas. Esse módulo também recebe os comandos de edição ao vivo, como o *addDocument* mencionado, e garante suas execuções.

Os sistemas de TVD discutidos na Seção 2.5, no entanto, introduzem uma restrição no controle do ciclo de vida das aplicações. Para esses sistemas, e também para o SBTVD, o controle dinâmico do ciclo de vida das aplicações deve poder ser realizado por tabelas AIT, que possui as limitações apresentadas na Seção 2.5. Essas limitações são contornadas pelos comandos de edição NCL, sem, contudo, desobedecer ao controle exercido pelas tabelas AIT. Para todo comando existente na tabela AIT existe um equivalente funcional de um ou mais comandos de edição NCL, apresentados na Tabela 8 para o sistema brasileiro de TVD terrestre. A partir de uma sinalização (a recepção do primeiro comando) AIT, os comandos de edição NCL podem ser usados.

Comando AIT	Comportamento Gíng-NCL	Equivalência Comando de Edição NCL	Restrições
PREFETCH	O Formataador NCL é preparado. O Gerente de Bases Privadas aguarda pelo comando <i>addDocument</i> para adicionar a aplicação à base privada indicada no comando de edição; e pelo comando <i>startDocument</i> para iniciar a apresentação da aplicação.		(i) A base privada será identificada pelo valor do campo <i>network_id</i> da tabela NIT ou pelo valor “ <i>network_id.program_number</i> ”, considerando <i>program_number</i> o identificador do serviço que a AIT foi multiplexada.
AUTOSTART	A aplicação terá sua apresentação iniciada automaticamente quando o serviço for selecionado.	Comando <i>addDocument</i> seguido do comando <i>startDocument</i>	(i) A base privada será identificada pelo valor do campo <i>network_id</i> da tabela NIT ou pelo valor “ <i>network_id.program_number</i> ”, considerando <i>program_number</i> o identificador do serviço que a AIT foi

			<p>multiplexada;</p> <p>(ii) Identificação de recursos do carrossel possui as mesmas restrições discutidas na Seção 2.4.1;</p> <p>(iii) O atributo <code>id</code> do elemento <code>&lt;ncl&gt;</code> (Soares, 2006) deve possuir o mesmo valor do campo <code>application_id</code> da estrutura <code>application_identifier</code> da AIT (ABNT, 2008b);</p> <p>(iv) A aplicação NCL será iniciada por todas as interfaces do elemento <code>&lt;body&gt;</code> (Soares, 2006);</p> <p>(v) A aplicação não pode ter seu início agendado para um determinado instante no tempo;</p> <p>(vi) A aplicação NCL será sempre iniciada a partir de seu início, uma vez que o parâmetro <code>offset</code> será 0.</p>
PRESENT	A aplicação não será iniciada automaticamente. Ela será disponibilizada em uma lista de aplicações que podem ser iniciadas pelo usuário.	Similar ao AUTOSTART. No entanto, os comandos serão processados após a aplicação ser selecionada pelo usuário.	Veja AUTOSTART
UNBOUND	A aplicação não será iniciada automaticamente. Ela será disponibilizada em uma lista de aplicações que podem ser iniciadas pelo usuário. A aplicação pode ser armazenada para ser iniciada em outro instante.	Similar ao PRESENT. No entanto, a aplicação pode ser armazenada para exibição em outro instante.	Veja PRESENT
DESTROY	A aplicação será parada.	Comando <code>stopDocument</code>	<p>(i) A base privada será identificada pelo valor do campo <code>network_id</code> da tabela NIT ou pelo valor <code>"network_id.program_number"</code>, considerando <code>program_number</code> o identificador do serviço que a AIT foi multiplexada;</p> <p>(ii) O atributo <code>id</code> do elemento <code>&lt;ncl&gt;</code> (Soares, 2006) deve possuir o mesmo valor do campo <code>application_id</code> da</p>

			<p>estrutura <i>application_identifier</i> da AIT (ABNT, 2008b);</p> <p>(iii) A aplicação não pode ter seu fim agendado para um determinado instante no tempo.</p>
KILL	A aplicação será parada e removida do receptor.	<p>Comando <i>stopDocument</i> seguido do comando <i>removeDocument</i></p>	<p>(i) A base privada será identificada pelo valor do campo <i>network_id</i> da tabela NIT ou pelo valor “<i>network_id.program_number</i>”, considerando <i>program_number</i> o identificador do serviço que a AIT foi multiplexada;</p> <p>(ii) O atributo <i>id</i> do elemento &lt;ncl&gt; (Soares, 2006) deve possuir o mesmo valor do campo <i>application_id</i> da estrutura <i>application_identifier</i> da AIT (ABNT, 2008b);</p> <p>(iii) A aplicação não pode ter seu fim agendado para um determinado instante no tempo;</p> <p>(iv) A aplicação não pode ser reiniciada após ser parada, sem que todo o processo de AUTOSTART seja refeito (exceto se iniciada por PREFETCH ou UNBOUND).</p>
REMOTE	<p>A aplicação remota ficará disponível para ter sua apresentação iniciada após a seleção do serviço. O Formatador NCL é preparado. O Gerente de Bases Privadas aguarda pelo comando <i>addDocument</i> para adicionar a aplicação à base privada indicada no comando de edição; e pelo comando <i>startDocument</i> para iniciar a apresentação da aplicação.</p>		<p>(i) A base privada será identificada pelo valor do campo <i>network_id</i> da tabela NIT ou pelo valor “<i>network_id.program_number</i>”, considerando <i>program_number</i> o identificador do serviço que a AIT foi multiplexada.</p>

Tabela 8 – Controle do Ciclo de Vida das Aplicações NCL por meio de tabelas AIT.

Note, na Tabela 8, que após receber os comandos PREFETCH, AUTOSTART, PRESENT, UNBOUND ou REMOTE, a flexibilidade dos comandos de edição NCL pode ser explorada. Note também, que o comando

UNBOUND surge como novidade, por ser um comando especificado apenas no SBTVD (ABNT, 2008b). Além disso, por meio dos comandos PREFETCH e REMOTE, é possível o agendamento de comandos de edição NCL para o controle de todo o ciclo de vida da aplicação.

Um comando de edição NCL pode ter seu processamento agendado através da especificação de valores temporais definidos em um campo denominado *eventNPT*, que faz parte da estrutura de dados que transporta o comando de edição (estrutura discutida no Capítulo 8). Quando o valor é atingido pela base temporal, que pode ser associada à estrutura através dos *taps* discutidos na Seção 2.4.3, o comando é entregue ao Gerente de Bases Privadas que, então, realiza o seu processamento.

Para o comando *startDocument*, existe ainda a possibilidade de definir o sincronismo do início da apresentação do documento com um valor NPT de uma base temporal. Nesse caso, o valor NPT e o identificador da base temporal são especificados nos parâmetros do comando, como apresentado na Tabela 5 (veja Capítulo 5).

Para que os objetos das aplicações, que referenciam fluxos elementares, sejam corretamente associados às bases temporais, é importante considerar os valores temporais no envio dos comandos *startDocument*. A Tabela 9, apresenta quatro ciclos de estruturas, formados por carrosséis de objetos e comandos de edição NCL, para transmissão nos segmentos definidos para o exemplo apresentado na Figura 9 (veja Seção 6.3). Os ciclos permitem que os receptores que sintonizam o canal em diferentes momentos da transmissão possam receber e executar as aplicações nos instantes especificados pelo provedor de conteúdo.

Na Tabela 9, quando o instante de *eventNPT* é especificado como 0, o comando é imediatamente entregue ao Gerente de Bases Privadas para ser processado. Caso contrário, espera-se que o valor NPT da base temporal atinja o valor especificado em *eventNPT* para que o comando seja entregue ao Gerente de Base Privadas. No ciclo definido para o segmento 1, note que o comando *startDocument* da aplicação “App1” referencia a base temporal “cid1” (parâmetro *nptBaseId* do comando) e o valor 5 para início da aplicação (parâmetro *nptTrigger* do comando). Assim, de acordo com a semântica do comando *startDocument* definida na Tabela 5 do Capítulo 5, se o receptor sintonizar o canal quando  $NPT_{cid1}$  for menor ou igual a 5, o Gerente de Bases Privadas aguarda que a base

temporal “cid1” chegue ao valor 5 para iniciar a apresentação do documento. Caso o receptor sintonize o canal quando  $NPT_{cid1}$  for maior que 5, o Gerente de Bases Privadas inicia imediatamente a apresentação da aplicação, considerando um deslocamento (Costa, 2006a) dessa apresentação de valor ( $NPT_{cid1} - 5$ ).

Outro ponto que deve ser observado nesse exemplo é a possibilidade do provedor de conteúdo enviar o carrossel antes do instante em que a aplicação será iniciada, como ocorre com a aplicação “App2”, enviada em S1 e S2; e a aplicação “App3”, enviada em S2 e S3. Observe ainda, que o agendamento do término da aplicação é realizado no exemplo através do campo *eventNPT*. Por exemplo, quando a base temporal “cid2” chegar ao valor  $NPT_{cid2}$  10, o Gerente de Bases Privadas receberá o comando que faz com que a apresentação da aplicação “App2” seja terminada. Um último ponto que vale ser ressaltado é a possibilidade de agendar o início da apresentação de uma aplicação, mesmo sem utilizar os parâmetros *nptBaseId* e *nptTrigger* do comando *startDocument*. No exemplo, isso ocorre para o início da aplicação “App3”.

Estrutura	Ciclos			
	S1	S2	S3	S4
carrossel(App1)	✓	✓	✓	✓
eventNPT = 0; addDocument(App1)	✓	✓	✓	✓
eventNPT = 0; startDocument(App1, cid1, 5)	✓			
eventNPT <sub>cid1</sub> = 30; startDocument(App1, cid1, 5)		✓	✓	✓
eventNPT <sub>cid1</sub> = 60; stopDocument(App1)				✓
carrossel(App2)	✓	✓		
eventNPT = 0; addDocument(App2)	✓	✓		
eventNPT = 0; startDocument(App2, cid2, 0.1)	✓	✓		
eventNPT <sub>cid2</sub> = 10; stopDocument(App2)		✓		
carrossel(App3)		✓	✓	
eventNPT = 0; addDocument(App3)		✓	✓	
eventNPT <sub>cid3</sub> = 1; startDocument(App3)		✓	✓	
eventNPT <sub>cid3</sub> = 15; stopDocument(App3)			✓	

Tabela 9 – Exemplo de Transmissão de Estruturas para Controle do Ciclo de Vida das Aplicações.

Ainda no comando *startDocument*, quando o parâmetro *nptTrigger* é especificado como “null”, o valor *eventNPT* identifica o instante de tempo em que a aplicação é iniciada. Se o parâmetro *nptBaseId* se referir a uma base temporal inexistente, o comando é ignorado.

Para que a Tabela 9 considere a restrição de sinalização discutida, antes de cada evento *addDocument*, deveria ser enviada, no exemplo, uma tabela AIT com o comando PREFETCH. Os comandos *stopDocument*, poderiam ser substituídos por tabelas AIT com comandos DESTROY. No entanto, nesse caso, os instantes dos comandos DESTROY são diferentes, por ser impossível o agendamento por tabelas AIT. Assim, cada DESTROY deveria ser enviado após o término do segmento ao qual a aplicação deve ser exibida.