

# Segmentação Automática de Fluxos Musicais: uma Abordagem via Multi-Agentes

Ernesto Trajano de Lima Neto

Dissertação submetida à Coordenação do Curso de Pós-Graduação em Informática da Universidade Federal da Paraíba - Campus II como parte dos requisitos necessários para obtenção do grau de Mestre em Informática.

Área de Concentração: Ciência da Computação

Linha de Pesquisa: Teoria da Computação e Inteligência Artificial

Evandro de Barros Costa e Didier Guigue  
(Orientadores)

Campina Grande, Paraíba, Brasil

©Ernesto Trajano de Lima Neto, Agosto de 2001

UFFB - BIBLIOTECA - CAMPUS II	
407	02.04-2002

Ficha Catalográfica

LIMA NETO, Ernesto Trajano de

**L732S**

Segmentação Automática de Fluxos Musicais: uma Abordagem via Multi-Agentes.

Universidade Federal da Paraíba, Departamento de Sistemas e Computação,  
Coordenação de Pós-Graduação em Informática, Campina Grande-PB, Agosto de 2001.

89 p. Il.

Orientadores: Evandro de Barros Costa  
Didier Guigue

Palavras Chaves:

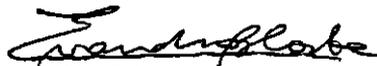
1. Inteligência Artificial
2. Computação Musical
3. Análise Musical
4. Sistemas Multi-Agentes
5. Segmentação

**CDU: 007.52**

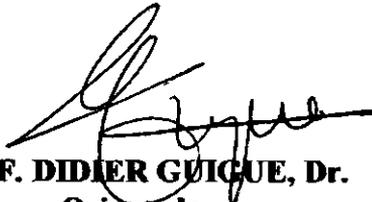
**SEGMENTAÇÃO AUTOMÁTICA DE FLUXOS MUSICAIS: UMA  
ABORDAGEM VIA MULTI-AGENTES**

**ERNESTO TRAJANO DE LIMA NETO**

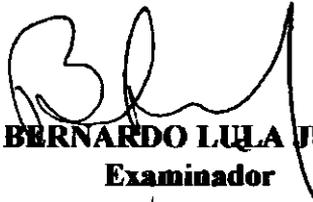
**DISSERTAÇÃO APROVADA EM 21.08.2001**



**PROF. EVANDRO DE BARROS COSTA, D.Sc**  
**Orientador**



**PROF. DIDIER GUIGUE, Dr.**  
**Orientador**



**PROF. BERNARDO LILA JÚNIOR, Dr.**  
**Examinador**



**PROF. GEBER LISBOA RAMALHO, Dr.**  
**Examinadora**

**CAMPINA GRANDE - PB**

## **Resumo**

A Análise Musical pode ser vista como sendo um ramo da Musicologia que procura compreender e explicar a estrutura de uma obra musical. Dentro deste âmbito, a segmentação de um fluxo musical, isto é, a divisão sucessiva de uma obra em unidades estruturais lógicas, é uma atividade de extrema importância. Muito tem sido feito em busca da automação da segmentação. Entretanto, poucos são os resultados musicalmente significantes. Neste trabalho apresenta-se um protótipo de um sistema de segmentação automática de fluxos musicais, o SOS. Ele baseia-se em dois modelos: um modelo de Análise Musical—a Análise Orientada a Objetos Sonoros— e um arcabouço para a construção de Sistemas Multi-Agentes—o MATHEMA. O resultado é um Sistema Multi-Agentes capaz de segmentar um fluxo musical de acordo com o modelo analítico proposto.

## **Abstract**

Many works deal with the automatization of the segmentation of musical flows. However, most of the results are not significant, musically speaking. In this work we present a prototype, the SOS, that deals with the automatization of the segmentation of musical flows. It is based on two basic models, one of musical analysis—the Sonic Object Analysis—and another for building Multi-Agents Systems—the MATHEMA. The result is a Multi-Agent System that is able to segment a musical flow following the analytical model.

## Agradecimentos

Aos meus orientadores e amigos Evandro e Didier, pela ajuda e enorme paciência.

Aos professores Bernardo Lula Jr. e Angelo Perkusich, pela ajuda em momentos menos interessantes.

Aos amigos do LabPetri, aos que lá estão e aos que já passaram desta para uma melhor, pela acolhida calorosa.

À equipe em Maceió, Klebson, Cristine e Hyggo, sem os quais partes cruciais deste trabalho não estariam aqui descritas.

À COPIN e à CAPES, pelo apoio financeiro.

A todos os colegas no mestrado, aos que me ajudaram e aos que não atrapalharam, na esperança de que eu mesmo, ao menos, não os tenha atrapalhado!

Por fim, à Ada. Sem ela *nada* disto existiria.

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Objetivos . . . . .	2
1.2	Estrutura da Dissertação . . . . .	3
<b>2</b>	<b>Segmentação Automática de Fluxos Musicais</b>	<b>4</b>
2.1	Análise Musical e Segmentação . . . . .	4
2.2	Segmentação Automática de Fluxos Musicais . . . . .	6
2.2.1	Redes Neurais . . . . .	7
2.2.2	<i>Morphologie</i> . . . . .	11
2.2.3	<i>Humdrum</i> . . . . .	13
2.2.4	GCTMS: <i>General Computational Theory of Musical Structure</i> . . . . .	15
2.2.5	Considerações Finais . . . . .	17
2.3	Análise Orientada a Objetos Sonoros . . . . .	18
2.3.1	Um Modelo Mínimo para a Análise Orientada a Objetos Sonoros . . . . .	22
<b>3</b>	<b>Sistemas Multi-Agentes</b>	<b>24</b>
3.1	Inteligência Artificial Distribuída . . . . .	24
3.2	Agentes . . . . .	25
3.3	Sistemas Multi-Agentes . . . . .	29
3.4	Segmentação Automática de Fluxos Musicais e Sistemas Multi-Agentes . . . . .	33
3.5	MATHEMA: um Arcabouço para a Construção de Sistemas Multi-Agentes . . . . .	35
3.6	Considerações Finais . . . . .	40
<b>4</b>	<b>SOS: um Protótipo para a Segmentação Orientada a Objetos Sonoros</b>	<b>42</b>
4.1	O Processo de Segmentação . . . . .	42

4.2	Algoritmos de Segmentação . . . . .	46
4.3	Descrição Geral do Sistema . . . . .	53
4.4	Modelo do Sistema . . . . .	56
4.5	Agentes Segmentadores: Arquitetura Interna . . . . .	61
<b>5</b>	<b>Implementação e Testes</b>	<b>66</b>
5.1	Aspectos de Implementação . . . . .	66
5.2	Testes . . . . .	69
<b>6</b>	<b>Conclusões e Trabalhos Futuros</b>	<b>74</b>
6.1	Conclusões e Contribuições do Trabalho . . . . .	74
6.2	Trabalhos futuros . . . . .	76
<b>A</b>	<b>MIDI</b>	<b>86</b>
A.1	Introdução . . . . .	86
A.2	Classificação das Mensagens MIDI . . . . .	87
A.3	Conteúdo das Mensagens MIDI . . . . .	88

# Lista de Figuras

2.1	Critérios de segmentação: durações longas (A), pausas (B) e rupturas de continuidade (C) . . . . .	8
2.2	Bach Sinfonia 3 e duas segmentações possíveis . . . . .	10
2.3	Bach Sinfonia 3 e segmentação adequada . . . . .	10
2.4	Zona nebulosa de segmentação . . . . .	12
2.5	Representação das cadências . . . . .	14
2.6	Comando usado na extração das cadências . . . . .	14
2.7	A GCTMS . . . . .	16
2.8	Os componentes de um objeto sonoro . . . . .	19
3.1	Um agente para Russel e Norvig . . . . .	26
3.2	O MATHEMA . . . . .	35
3.3	Um agente MATHEMA . . . . .	38
3.4	Adaptação do MATHEMA—auxílio ao projetista de sistemas . . . . .	39
3.5	Adaptação do MATHEMA—gerência de redes de computadores . . . . .	40
3.6	Modelo mínimo da arquitetura do MATHEMA . . . . .	41
4.1	Macro-segmentação . . . . .	43
4.2	Proto-segmentação . . . . .	43
4.3	Resultado da primeira segmentação . . . . .	44
4.4	Resultado da segunda segmentação . . . . .	45
4.5	Resultado da terceira segmentação . . . . .	46
4.6	Exemplo de uma “interrupção” no fluxo de notas . . . . .	49
4.7	Arquitetura do SOS . . . . .	55
4.8	Hierarquia de páginas do SOS . . . . .	57

---

4.9	Encaminhamento de um fluxo . . . . .	58
4.10	Modelo da macro-segmentação . . . . .	59
4.11	Modelo da proto-segmentação . . . . .	60
4.12	Modelo da fase de segmentação . . . . .	61
4.13	Arquitetura do agente mediador . . . . .	62
4.14	Arquitetura detalhada do Agente Mediador . . . . .	64
4.15	Arquitetura dos agentes reativos . . . . .	65
5.1	Visão geral do SOS . . . . .	67
5.2	O pacote <i>Agente</i> . . . . .	67
5.3	O pacote <i>AgenteMediador</i> . . . . .	69
5.4	O pacote <i>AgenteSegmentador</i> . . . . .	70
5.5	Trecho final da Página 1 (© Universal Edition Ltda, Toronto, Canadá) . . . .	71
5.6	O protótipo após a segmentação da página 1 . . . . .	73
A.1	Uma mensagem MIDI . . . . .	88

# Lista de Tabelas

2.1	Resultados dos três experimentos: reconhecimento de pausas (Exp. 1), identificação de durações mais longas (Exp. 2) e verificação das rupturas de similaridade (Exp. 3) . . . . .	8
2.2	Classificação das intensidades em relação à notação musical tradicional . .	23
2.3	Classificação dos registros do piano . . . . .	23
3.1	Resumo da classificação das situações de interação . . . . .	31
4.1	Série de Fibonacci e grupos associados . . . . .	51
4.2	Classificação das velocidades e grupos associados . . . . .	51
4.3	Classificação dos registros do piano . . . . .	53
5.1	Resultados da primeira seqüência de testes . . . . .	72

# Capítulo 1

## Introdução

O homem sempre gostou de estudar a Música através das máquinas. A caixa de música e a pianola, entre outras tantas máquinas, são testemunhas deste fato. Quando Charles Babagge, nos idos de 1842, concebeu a sua Máquina Analítica, Lady Ada Lovelace fez a seguinte observação acerca de suas possibilidades (citada por Rich & Knight [RK93, p. 7–8]):

*O mecanismo operacional pode até entrar em ação independentemente de ter um objeto no qual operar (é claro que neste caso nenhum resultado poderá ser desenvolvido). Novamente, ele pode operar sobre outras coisas que não números, caso haja objetos cujas relações mútuas fundamentais possam ser expressas pela ciência abstrata das operações e que devam também ser suscetíveis a adaptações, à ação da notação operacional e ao mecanismo da máquina. Supondo-se, por exemplo, que as relações fundamentais dos sons na ciência da harmonia e da composição musical sejam suscetíveis a tais expressões e adaptações, a máquina poderia então compor peças de música elaboradas e científicas em qualquer grau de complexidade ou extensão.*

Foi, no entanto, o advento dos computadores, após a Segunda Guerra Mundial, que propiciou um maior número de experiências, levando ao estabelecimento de uma área de pesquisas, hoje denominada Computação Musical. Diversas áreas da Música se beneficiaram e continuam a se beneficiar desta disciplina: a Composição, a Análise Musical, a Educação Musical, entre outras, são exemplos de atividades que têm utilizado o computador intensamente [R<sup>+</sup>96; B<sup>+</sup>92; Dan00; Cop99; HL84].

Apesar de fazerem uso intenso das máquinas, as Artes, e a Música em particular, possuem problemas de difícil formalização e investigação, uma vez que estão em jogo elementos subjetivos, tais como criatividade, emoção e intuição, entre outros. Ainda assim, o pensamento musical mantém uma íntima ligação com forma, estruturação e leis sobre as relações entre eventos sonoros, permitindo, por exemplo, a classificação de uma obra musical como pertencente a um determinado estilo ou a um compositor em particular.

A Análise Musical Assistida por Computador tem como um de seus objetivos a construção de modelos computacionais que “ajudem na formulação de teorias capazes de descrever atividades e tarefas musicais de maneira explícita e consistente” [Cam98, p. 2]<sup>1</sup>. Ela reúne um conjunto de problemas significantes, mas de difícil resolução. Um destes problemas é a automação da segmentação de fluxos musicais. Um fluxo musical é definido como sendo uma obra ou um trecho de obra e a segmentação é a divisão de um fluxo em unidades elementares. A segmentação é importante pois é a partir dela que o analista constrói a sua análise da obra. Ela é de difícil resolução porque parte dos princípios e métodos de segmentação, além de serem de formalização complexa, variam de obra para obra, de fluxo para fluxo.

Por outro lado, parte da pesquisa em Inteligência Artificial (IA) está diretamente ligada à escrita de programas para a solução de problemas que não possuem solução algorítmica direta. Neste âmbito, muito tem sido feito em uma área de pesquisa denominada Inteligência Artificial Distribuída (IAD). Nela, tenta-se construir soluções distribuídas para problemas complexos, como os problemas advindos da investigação da Música com o uso de computadores. Para tanto, estas soluções são construídas a partir de entidades computacionais chamadas agentes. Cada agente possui um conhecimento específico sobre o problema ou sobre o domínio do problema e a solução resulta da interação entre os agentes.

## 1.1 **Objetivos**

O objetivo geral desta dissertação é a concepção de um sistema para segmentação automática de fluxos musicais. Este objetivo global passa por diversos objetivos específicos.

O primeiro deles é a identificação e enumeração das dificuldades computacionais existentes quando da automação do processo de segmentação de fluxos musicais. Isto será feito

---

<sup>1</sup>As traduções são do autor.

através da análise e discussão das diversas abordagens utilizadas na construção de sistemas de segmentação até o presente momento. Como resultado, tem-se uma espécie de conjunto dos requisitos necessários para a construção de sistemas deste tipo. Em seguida, apresenta-se uma metodologia de análise musical e justifica-se seu uso no sistema de segmentação a ser concebido.

De posse do conjunto de requisitos e da metodologia de análise musical, aponta-se uma possível maneira para a construção de um sistema de segmentação que permita a satisfação deste conjunto de requisitos e do processo de segmentação prescrito pela metodologia de análise.

Por fim, modela-se, implementa-se e testa-se a solução concebida para a automação da segmentação de fluxos musicais.

## **1.2 Estrutura da Dissertação**

Esta dissertação é estruturada da seguinte maneira: no Capítulo 2, trata-se dos aspectos musicais relacionados ao tema desta dissertação, em particular dos problemas relacionados à automação da segmentação de fluxos musicais. Aqui, apresenta-se também a metodologia de análise musical escolhida.

No Capítulo 3, apresenta-se e justifica-se os conceitos de Inteligência Artificial que serão utilizados no sistema de segmentação desenvolvido, a saber: Inteligência Artificial Distribuída, agentes e Sistemas Multi-Agentes.

No Capítulo 4, apresenta-se a solução desenvolvida, incluindo o modelo do sistema e os algoritmos de segmentação.

No Capítulo 5, apresenta-se aspectos da implementação do protótipo e os testes realizados com este.

No Capítulo 6, apresenta-se conclusões e trabalhos futuros. Além disso, enumera-se as contribuições obtidas.

Por fim, um Apêndice explica em maiores detalhes a linguagem de representação da música utilizada neste trabalho.

## Capítulo 2

# Segmentação Automática de Fluxos Musicais

Neste capítulo trata-se dos aspectos musicais relacionados ao tema desta dissertação, em particular, dos problemas relacionados à automação da segmentação de fluxos musicais. Explica-se, em primeiro lugar, o que é análise musical e segmentação de fluxos musicais, assim como evidencia-se a importância da segmentação para a análise. Em seguida, identifica-se algumas das dificuldades da atividade de segmentação de fluxos musicais, faz-se um exame de algumas das diversas alternativas existentes para a segmentação automática destes fluxos e apresenta-se considerações que podem levar a uma melhora na automação do processo de segmentação. Por fim, apresenta-se a metodologia de análise musical que será contemplada nesta dissertação.

### 2.1 Análise Musical e Segmentação

A análise musical pode ser vista como sendo um ramo da musicologia que procura compreender e explicar a estrutura de uma obra musical. Musicólogos a definem das maneiras mais diversas. Bent, por exemplo, define-a como sendo “a resolução de uma estrutura musical em elementos constituintes relativamente mais simples e a investigação das funções destes elementos dentro desta estrutura” [Ben80]. Cook, por sua vez, define análise musical como a atividade que investiga “como os componentes da música se relacionam entre si e quais destas relações são mais importantes que outras” [Coo94]. Meeùs vê uma obra musical como

um “complexo de elementos interdependentes articulados entre si” [Mee94].

Nas definições anteriores encontram-se palavras como “estrutura musical” e relacionamentos entre “componentes” ou “elementos” da música. Mas, como obter os componentes ou elementos da música, partindo de uma obra musical completa ou mesmo de um trecho de obra? É a segmentação, uma das primeiras tarefas dentro do processo de análise musical, a responsável por encontrar configurações particulares destes componentes ou elementos da música. A segmentação pode ser vista como “a maneira que o analista divide a música em segmentos formais” [Coo94]<sup>1</sup> ou ainda como “uma parte do processo de análise musical que consiste em delimitar em uma obra os segmentos utilizados como entidades significativas na análise” [Mes94]. Ela é de extrema importância dentro do processo de análise musical pois é apenas a partir de seu resultado que o analista começa a estabelecer relacionamentos entre os segmentos a fim de determinar a estrutura da obra. Como escreveu Cook, “exceto por decisões finais de interpretação, tudo na análise depende da segmentação porque é nela que todas as decisões musicais são feitas” [Coo94].

À primeira vista, o processo de segmentação definido acima pode parecer simples, mas, na verdade, ele não é tão simples. Os questionamentos envolvidos no processo de segmentação não são de forma alguma triviais. A primeira dificuldade reside na existência de uma grande variedade de metodologias de análise. Na verdade, o problema não é a variedade de metodologias em si, mas sim a quantidade de critérios de segmentação que esta variedade de metodologias impõe. De uma forma geral, cada metodologia de análise possui pelo menos um critério segmentador próprio, o que faz com que a tarefa de segmentação esteja atrelada à abordagem adotada pelo analista. Isto é compreensível, pois cada metodologia de análise parte de premissas de compreensão da estrutura musical diferentes, sendo aplicada a repertórios também diferentes.

Além do problema da variedade de critérios de segmentação, não é incomum que muitos destes critérios estejam atrelados a considerações estéticas, estilísticas e até históricas: na análise de música tonal, por exemplo, “tensão” e “relaxamento” são critérios frequentemente utilizados na segmentação. No entanto, estes critérios possuem significados diferentes em épocas diferentes: o que era tensão para Mozart e seus contemporâneos não é mais considerada como tal para compositores da Bossa Nova, por exemplo.

---

<sup>1</sup>O termo *formal* aqui relaciona-se às diversas formas musicais e não a formalismos matemáticos.

Por vezes, a metodologia de análise não chega nem a estabelecer critérios para proceder com a segmentação. Este é o caso, por exemplo, da Teoria dos Conjuntos de Notas<sup>2</sup>. Pode ocorrer também que a metodologia forneça critérios claros para a segmentação, mas não estabeleça um procedimento para decidir, ou ao menos indicar, dentre as várias possibilidades, qual é a segmentação correta ou mais adequada<sup>3</sup>. O primeiro caso é mais sério, pois implica na inexistência de um procedimento para a realização da segmentação. No segundo caso, é possível encontrar segmentos dentro de uma obra musical. A dificuldade reside na decisão de quais segmentos considerar dentro da análise. Isto é, mesmo de posse de critérios claros de segmentação, encontrar segmentos dentro de uma obra musical não necessariamente indica que estes segmentos sejam estruturalmente importantes. Cabe ao analista, após uma comparação sob diversos aspectos, decidir que segmentos devem ser considerados para a compreensão da estrutura da obra.

Em resumo, deve-se ter em mente que a música é construída a partir de vários componentes. Notas, ritmos, leis harmônicas, parâmetros estatísticos, entre outros, são utilizados pelos compositores na estruturação de uma obra musical. A atividade de decompor esta obra em estruturas menores, em algum momento passará por resultados conflitantes: superposição de linhas melódicas, parâmetros estatísticos conflitantes, leis independentes e por vezes antagônicas de distribuição dos sons no espaço e no tempo, podem ser alguns dos problemas enfrentados pelos analistas.

## 2.2 Segmentação Automática de Fluxos Musicais

A seção anterior foram apresentados alguns dos problemas existentes na tarefa de segmentação. Nesta seção explica-se como alguns autores abordam estes problemas, quando da tentativa de automação da segmentação. Isto será feito através do exame de algumas das diversas alternativas existentes para a segmentação automática de fluxos musicais. A tentativa aqui é identificar as características de cada abordagem, confrontá-las com os problemas

---

<sup>2</sup>Críticas acerca da falta de critérios de segmentação desta metodologia podem ser encontradas em [Coo94, p. 146] e [Gui97b, p. 27].

<sup>3</sup>Este é o caso da análise paradigmática, que fornece um critério claro de segmentação, a repetição, mas não apresenta nenhum mecanismo para que o analista decida quais são as repetições de maior importância [Coo94, p. 180].

apresentados na Seção 2.1, verificando que espécie de estratégia foi utilizada na tentativa de resolução de tais problemas.

### 2.2.1 Redes Neurais

As redes neurais têm sido utilizadas na Computação Musical para diversos fins [TL91]. Uma das utilizações é na segmentação automática. Aqui analisa-se duas abordagens que utilizam esta técnica<sup>4</sup>.

#### Redes Neurais e Segmentação Rítmica

A primeira abordagem—descrita em [Car95]—utiliza redes neurais para a realização da segmentação rítmica de passagens melódicas. Esta segmentação é realizada a partir de três critérios: durações longas, pausas e rupturas de similaridade. O primeiro critério, notas longas, é utilizado quando, em uma determinada seqüência de notas, uma das notas possui maior duração. Assim, supondo uma seqüência de quatro notas  $n_1n_2n_3n_4$ , onde a segunda nota,  $n_2$ , é mais longa que as outras obteríamos dois segmentos, sendo o primeiro formado pelas notas  $n_1n_2$  e o segundo formado pelas notas  $n_3n_4$ . Uma segmentação por pausas ocorre quando em uma dada seqüência de notas existe uma pausa entre duas notas. Assim, considerando a mesma seqüência anterior, existindo uma pausa entre duas notas,  $n_2$  e  $n_3$  por exemplo, os segmentos resultantes seriam  $n_1n_2$  e  $n_3n_4$ . O terceiro critério, ruptura de similaridade, ocorre quando existe uma quebra na continuidade rítmica. Considere uma seqüência de oito notas  $n_1n_2n_3n_4n_5n_6n_7n_8$  onde as quatro primeiras notas,  $n_1n_2n_3n_4$ , possuem durações idênticas e as quatro últimas,  $n_5n_6n_7n_8$ , também possuem durações idênticas, mas os grupos de notas possuem durações diferentes entre si. Neste caso ocorre uma ruptura de similaridade rítmica entre os dois grupos de notas. A Figura 2.1 ilustra os critérios descritos acima.

A segmentação é feita por redes neurais em três camadas, submetidas a três seqüências de treinamentos, uma para cada critério de segmentação, com seqüências de padrões gerados aleatoriamente. As redes treinadas foram então testadas com seis obras de J. S. Bach. As obras escolhidas foram duas invenções a duas vozes (números nove e treze), duas invenções a três vozes (números três e treze) e duas fugas do primeiro volume do Cravo Bem-Temperado

---

<sup>4</sup>Em ambos os casos não foi possível obter as implementações e realizar testes. Desta forma, a análise feita aqui é baseada nos artigos que descrevem as implementações e nos experimentos feitos pelos autores.



Figura 2.1: Critérios de segmentação: durações longas (A), pausas (B) e rupturas de continuidade (C)

(números quatro e dezessete). Cada voz foi entrada em separado e o resultado pode ser visto na tabela 2.1. Nela, PN—o número de padrões negativos—expressa a quantidade de padrões rítmicos que não são caso de segmentação, isto é, este número identifica os casos que a rede neural classificou como sendo um padrão rítmico mas este padrão identificado não se constitui em um caso de segmentação. PP—o número de padrões positivos—representa o número de padrões rítmicos que são casos de segmentação, isto é, este número identifica os casos que a rede neural classificou como sendo um padrão rítmico e este padrão identificado é um caso de segmentação. PNE e PPE representam, respectivamente, a porcentagem de padrões negativos classificados com erro e a porcentagem de padrões positivos classificados com erro.

	Experimento 1				Experimento 2				Experimento 3			
Peças	PN	PP	PNE	PPE	PN	PP	PNE	PPE	PN	PP	PNE	PPE
1	814	2	0	0	764	52	2	0	764	52	2	0
2	784	16	0	0	790	10	0	0	790	10	0	0
3	1188	12	0	0	1171	25	10	0	1171	25	10	0
4	2293	10	0	60	2253	39	18	8	2253	39	18	8
5	4556	44	0	14	4483	98	26	15	4483	98	26	15
6	2204	36	0	0	2185	48	22	6	2185	48	22	6

Tabela 2.1: Resultados dos três experimentos: reconhecimento de pausas (Exp. 1), identificação de durações mais longas (Exp. 2) e verificação das rupturas de similaridade (Exp. 3)

Os resultados dos experimentos apontam para a possibilidade de automação do processo de segmentação utilizando redes neurais. O autor verificou que redes neurais de mesma topologia são capazes de segmentar a partir de critérios diferentes, com o uso de aprendizagem supervisionada. No entanto, a segmentação é realizada em apenas um dos componentes

da música, o ritmo, e, uma vez que a música é construída a partir de vários componentes, pode-se questionar a validade de uma segmentação que lida apenas com um componente. Este problema fica mais evidente quando se observa o repertório utilizado nos experimentos: parte do repertório de Bach, incluindo-se aí suas fugas e invenções, é construído a partir da dinâmica existente entre as partes ou vozes. Na tentativa de automação apresentada aqui, uma vez que o sistema recebe cada voz em separado, fica impossível estabelecer uma segmentação, mesmo rítmica, levando em consideração o principal aspecto de estruturação da obra: o “jogo” existente entre as diversas vozes. Além disto, outros componentes podem ser tão ou mais importantes que o aspecto rítmico. É o caso, por exemplo, da influência das relações harmônicas estabelecidas pela superposição das diferentes vozes na determinação da estrutura das fugas. A própria superposição das vozes—polifonia—é ignorada nesta abordagem.

Também não fica claro se os padrões de ritmo são apenas dos tamanhos especificados para o treinamento, isto é, segmentações por pausas e por notas longas são observadas apenas entre a segunda e terceira notas de padrões de tamanho quatro? No caso das rupturas de continuidade, a segmentação só ocorre entre a quarta e quinta notas em padrões de tamanho oito? Caso isto se verifique, o processo de segmentação fica extremamente pobre e longe da realidade. O autor também não cogita outros padrões rítmicos para a segmentação. Por fim, propõe-se o seguinte problema: qual seria o resultado da segmentação do trecho ilustrado na Figura 2.2?

Neste exemplo em particular, não fica claro que tipo de segmentação seria obtida aplicando as redes neurais. A Figura 2.2 mostra duas segmentações possíveis: a primeira—Segmentação 1—resulta da aplicação do critério de durações longas; a segunda—Segmentação 2—resulta da utilização do critério de rupturas de continuidade. No entanto, ambos os resultados não são satisfatórios. Uma segmentação mais adequada seria a mostrada na Figura 2.3.

A segmentação pouco satisfatória parece residir no nível de observação da estrutura musical que neste caso é muito granular<sup>5</sup>.

---

<sup>5</sup>É importante frisar que o exemplo foi retirado do mesmo *corpus* de obras utilizado nos experimentos.

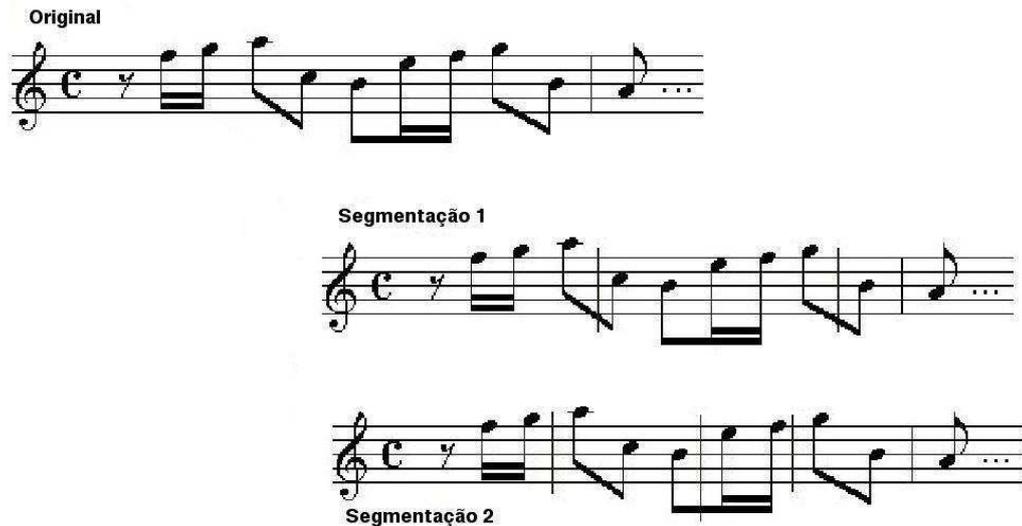


Figura 2.2: Bach Sinfonia 3 e duas segmentações possíveis



Figura 2.3: Bach Sinfonia 3 e segmentação adequada

### Redes Neurais e Teoria dos Conjuntos de Notas

A segunda abordagem—descrita em [Isa97]—utiliza redes neurais na construção de ferramentas que auxiliem o analista nas tarefas analíticas da Teoria dos Conjuntos de Notas. O autor utiliza uma rede neural com uma camada escondida, treinada utilizando a técnica de retropropagação (*Backpropagation*), nas tarefas de reconhecimento de classes intervalares, conteúdo intervalar, relações entre conjuntos de classes de notas e segmentação musical, sendo esta última a que será analisada aqui. O modelo de rede neural utilizado nesta abordagem é a rede IAC (*Interactive Activation and Competition*), pois ela é “apropriada para o problema de segmentação”, segundo o autor da abordagem.

O principal problema desta abordagem é a metodologia de análise escolhida: como a Teoria dos Conjuntos de Notas não possui critérios de segmentação definidos, o treinamento das redes fica comprometido, uma vez que não é possível estabelecer critérios para o treinamento. Para contornar este problema, o autor estabelece critérios próprios, a saber: tempo/andamento, altura, voz, grupo de ligadura, grupo dinâmico, dinâmica, ataque e articu-

lação. Apesar de serem critérios “musicalmente defensáveis”, como o próprio autor salienta, eles não são em nenhum momento explicados e tampouco relacionados com a Teoria dos Conjuntos de Notas. No final do artigo, o autor apresenta alguns *insights* importantes a cerca das características que um sistema de apoio à atividade de análise musical deveriam possuir, em particular sistemas que automatizem a segmentação:

1. devem existir “meios fáceis para testar várias hipóteses sobre a relevância de diversos critérios musicais”;
2. deve existir algum mecanismo que “permita que segmentações feitas anteriormente informem as segmentações posteriores”;
3. princípios analíticos “sugerem que mecanismos de mais alto nível possuem um papel importante [na determinação da estrutura musical]”;
4. o sistema “deve ser capaz de encontrar uma estruturação hierárquica na música”.

Estas características serão retomadas mais adiante durante o exame final dos sistemas de automação da segmentação.

### 2.2.2 *Morphologie*

*Morphologie* é uma biblioteca de funções de análise, reconhecimento, classificação e reconstituição de seqüências simbólicas e numéricas [BSV98] desenvolvida para a aplicação *OpenMusic*<sup>6</sup>. Ela foi concebida a partir da análise de 300 melodias dos esquimós que habitam a costa leste da Groenlândia. Notou-se que as análises realizadas até então não conseguiam lidar com sucesso com este tipo de música. O problema residia no fato das abordagens utilizadas identificarem apenas segmentos muito longos, enquanto que neste tipo de música observou-se que a estrutura é resultante da concatenação de segmentos curtos, ou microsegmentos, formados por três, quatro ou cinco notas [BSV98, p. 10]. Utilizando critérios de segmentação como marcas, contrastes ou descontinuidades, chegou-se ao seguinte algoritmo:

---

<sup>6</sup>*OpenMusic* foi desenvolvida pelo IRCAM—Institut de Recherche et Coordination Acoustique/Musique—e funciona como uma interface gráfica tanto para Common Lisp, quanto para CLOS [Ago98; Had99; ARL<sup>+</sup>99].

1. apontar todas as marcas/critérios possíveis em um determinado componente (alturas, ritmo, etc.);
2. suprimir os elementos idênticos simultâneos (uma vez que dois elementos idênticos não criam uma descontinuidade);
3. segmentar sistematicamente a seqüência em cada componente e para cada critério;
4. fazer o inventário, critério por critério, dos segmentos obtidos;
5. substituir cada segmento por seu “rótulo” a fim de obter a estrutura da seqüência de acordo com cada critério.

Analisando o algoritmo, pode-se notar pelo menos um avanço em relação às abordagens anteriores: aqui é possível realizar segmentações em vários componentes e com vários critérios, na verdade uma segmentação para cada par componente/critério. As diversas segmentações podem determinar pontos fortes de segmentação, quando dois ou mais pares coincidem no mesmo ponto do tempo, ou podem determinar zonas nebulosas de segmentação—*zone floue de segmentation*—quando não existe coincidência temporal entre as segmentações realizadas (Figura 2.4). Segundo o autor, este fenômeno “corresponde melhor à realidade musical, pelo menos do ponto de vista da percepção” [BSV98, p. 11].

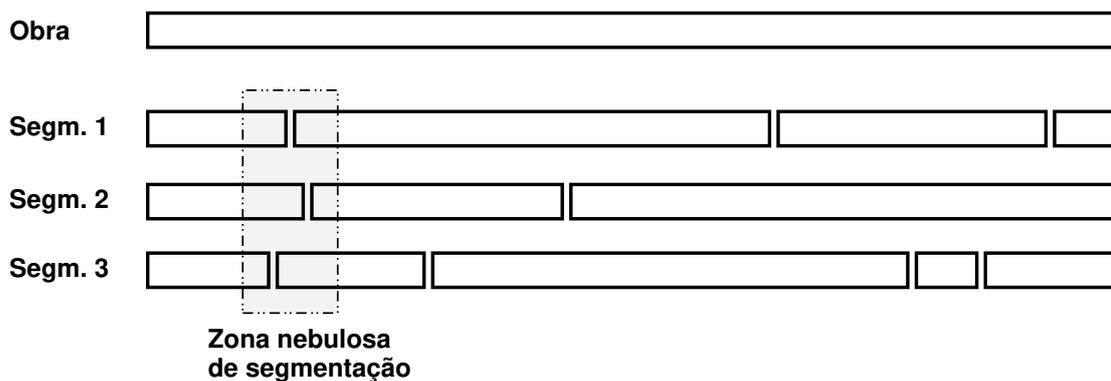


Figura 2.4: Zona nebulosa de segmentação

Apesar destes avanços, *Morphologie* também apresenta algumas limitações: o fato da biblioteca ter sido concebida para lidar com micro-segmentos em melodias faz com que a análise de qualquer outro tipo de música fique comprometida. O comprometimento acontece porque a quantidade de informação pouco pertinente que se origina das comparações feitas

pelos algoritmos é muito grande. Assim, o analista passa muito tempo tentando descobrir alguma informação interessante no meio de informação redundante e sem significação para a determinação da estrutura musical.

### 2.2.3 *Humdrum*

*Humdrum* é um sistema informático de apoio à análise musical que, através de um conjunto de ferramentas, permite que analistas e pesquisadores façam “perguntas” sobre a estrutura de uma determinada obra musical [Hur99]. Ao contrário dos sistemas anteriores, *Humdrum* não se baseia em um modelo específico de compreensão da estrutura musical. A idéia é que ele funcione como um ambiente de programação permitindo que os usuários representem obras musicais e que construam procedimentos analíticos específicos combinando as diversas ferramentas disponíveis.

O sistema é formado por dois componentes distintos: *Humdrum Syntax* e *Humdrum Toolkit*. O primeiro componente é, na verdade, uma gramática para a representação de informação. Sua função é fornecer um arcabouço para a representação de todo o tipo de informação, desde notação de musical medieval até dedilhados de piano, passando por passos de dança e dados MIDI<sup>7</sup>. O segundo componente, é um conjunto de mais de 70 ferramentas capazes de manipular dados ASCII descritos pela gramática. Estas manipulações podem ser de vários tipos: busca, contagem, classificação e comparação são alguns exemplos de manipulações realizadas pela ferramenta.

No que diz respeito à segmentação, *Humdrum* oferece alguns algoritmos de seleção e extração de informação, de reconhecimento de padrões e de classificação que podem ser utilizados com algum sucesso. No manual [Hur99], por exemplo, existe um exemplo capaz de classificar cadências em uma obra tonal. Ele parte de uma representação das funções harmônicas e tenta encontrar cadências—determinadas configurações de funções harmônicas— na obra a ser analisada<sup>8</sup>. A Figura 2.5 mostra o *script cadences*, que associa uma seqüência harmônica a um tipo de cadência. A primeira linha, por exemplo, associa a seqüência V I à

<sup>7</sup>Segundo o autor “teoricamente, qualquer tipo de dado simbólico sequencial pode ser representado” [Hur99].

<sup>8</sup>*Humdrum* já possui duas representações previamente definidas: `**harm`, que representa as funções harmônicas, e `**kern`, que é utilizada na representação das obras musicais.

cadência denominada `authentic`.

```
s/V I;/authentic/
s/V7 I;/authentic/
s/V i;/authentic/
s/V7 i;/authentic/
s/IV I;/plagal/
s/iv i;/plagal/
s/iv I;/plagal/
s/V vi;/deceptive/
s/V VI;/deceptive/
```

Figura 2.5: Representação das cadências

A partir desta representação, o analista pode iniciar a busca por cadências e, uma vez que as cadências geralmente determinam pontos de segmentação na música tonal, o analista pode encontrar segmentos. A busca por cadências é feita através do comando mostrado na Figura 2.6. Este comando é interpretado da seguinte maneira: extraia do arquivo `chorales` toda a informação referente ao campo `**harm`; retire todas as linhas de compasso (símbolos =) e agrupe dois a dois a seqüência de funções harmônicas resultante; execute o `script cadences`, que retornará todas as cadências que ocorrem no arquivo; por fim, associe este resultado a uma nova representação, denominada `**cadences`.

```
extract -i '**harm' chorales | context -o = -n 2 \ |
humsed -f cadences | sed 's/\**harm/**cadences/'
```

Figura 2.6: Comando usado na extração das cadências

Como se pode notar, apesar de *Humdrum* ser bastante flexível sua utilização é bastante obscura, principalmente para músicos. O próprio autor adverte: “o primeiro impedimento para uma solução rápida do problema proposto é a habilidade do usuário de interconectar as ferramentas corretas” [Hur99]. Além disso, apesar do sistema facilitar as investigações sobre a estrutura musical, caso o analista não possua um problema ou pergunta clara a ser

colocada o sistema será de pouca utilidade. Este problema parece estar ligado ao fato de *Humdrum* não estar atrelado a uma metodologia específica de análise musical, fazendo com que o analista necessite construí-la à medida que investiga uma determinada obra musical.

#### 2.2.4 GCTMS: *General Computational Theory of Musical Structure*

A GCTMS—*General Computational Theory of Musical Structure*—é uma “teoria que pode ser empregada para se obter uma descrição estrutural—ou conjunto de descrições—de uma superfície musical” [Cam98]. A teoria é independente de estilo ou idioma musical, uma vez que se baseia em princípios lógicos e cognitivos gerais, e pode ser aplicada a qualquer superfície musical<sup>9</sup>. Sua meta é alcançar uma descrição estrutural da obra musical que possa ser considerada como “plausível” ou “possível” pelo analista musical humano.

A teoria é formada por dois estágios distintos, mas intimamente ligados:

1. o estabelecimento de uma série de componentes individuais que abordam tarefas analíticas especializadas. Para tanto, o autor desenvolveu algoritmos que lidam com aspectos específicos da superfície musical, como por exemplo, um algoritmo que modela as estruturas métrica e de acentuação e outro para a indução de padrões.
2. Uma descrição elaborada e detalhada de como estes componentes se relacionam e interagem uns com os outros de forma que descrições estruturais plausíveis de uma superfície musical possam ser alcançadas. Este estágio é realizado comparando os resultados obtidos com os algoritmos do estágio anterior.

O processo pode ser acompanhado na Figura 2.7: uma superfície musical, composta por eventos discretos é convertida para uma outra superfície—*superfície musical (1)*—que é formada por perfis intervalares em vários níveis de abstração (para as alturas, por exemplo, intervalos exatos, *scale-step intervals*, *step-leap intervals*, etc.). Esta nova superfície é submetida a um algoritmo que localiza possíveis rupturas. O resultado, denominado proto-segmentação, é então submetido a outros algoritmos que tentam validar os segmentos encontrados ou mesmo identifica novos segmentos. Por fim, a segmentação, ou conjunto de

---

<sup>9</sup>O autor implementou um protótipo do sistema e algumas partes podem ser encontradas na biblioteca *OpenMusic* chamada *Kant* [Meu00].

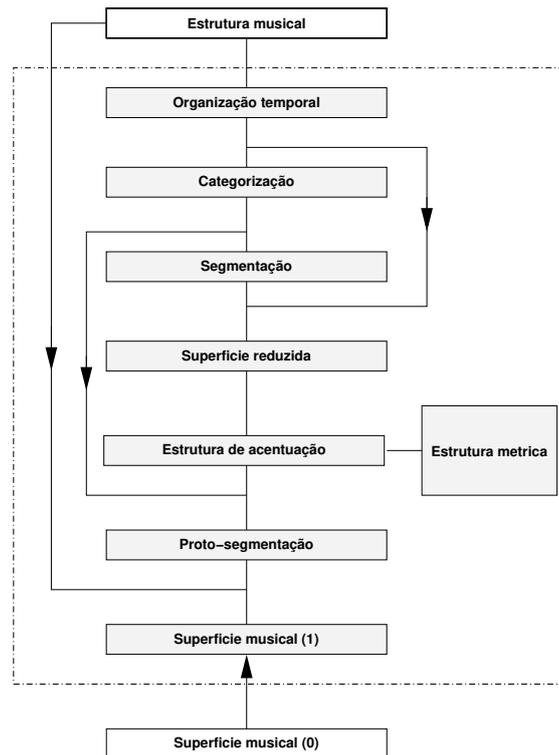


Figura 2.7: A GCTMS

segmentações, é organizada e rotulada baseada em conceitos de similaridade, fornecendo subsídios para que o analista escolha entre as várias segmentações possíveis.

Segundo o autor, a GCTMS pode ser útil nas seguintes áreas: teoria musical, pois provê uma teoria subjacente para a descrição da estrutura musical; sistemas musicais, pois pode servir de base para a construção de sistemas musicais inteligentes para educação, composição, análise, etc.); Inteligência Artificial, pois os modelos e algoritmos propostos são do interesse de áreas como representação do conhecimento, aprendizagem de máquina e casamento de padrões; e cognição musical, pois pode auxiliar o descobrimento dos processos cognitivos que ocorrem quando os seres humanos escutam música.

Esta é provavelmente a abordagem que fornece os melhores resultados. No entanto, apesar de construir a segmentação a partir de vários componentes, ela não consegue segmentar passagens polifônicas, ficando restrita a melodias ou trechos de melodias, o que reduz bastante seu uso<sup>10</sup>. Outro problema, é que o processo de análise não permite, em nenhum momento, que o analista intervenha ou que ele forneça ao sistema informações sobre a obra.

<sup>10</sup>A imensa maioria da música que se presta à análise musical é formada por música polifônica.

### 2.2.5 Considerações Finais

Viu-se nesta seção algumas tentativas de automação do processo de segmentação de fluxos musicais. A partir desta análise, de uma forma geral e desprezando os problemas particulares de cada abordagem, pode-se identificar os seguintes problemas:

1. algumas abordagens segmentam levando em consideração apenas um componente da estrutura musical.
2. O nível de observação da estrutura musical se situa no nível inferior, fazendo que a construção das estruturas musical aconteça a partir das notas, em uma abordagem *bottom-up*.
3. Não existe uma maneira de integrar as diversas abordagens.
4. O analista, detentor de conhecimento importante para o desenrolar do processo de segmentação, não é levado em consideração, uma vez que não pode nem prover os sistemas com informações, nem interagir com os sistemas de modo a melhorar os resultados.
5. Nenhuma das abordagens consegue segmentar passagens polifônicas, ficando limitadas à segmentação de melodias.

Do exposto acima e retomando as observações realizadas por Isaacson [Isa97], descritas na Seção 2.2.1, pode-se apontar algumas características que sistemas de segmentação automática de fluxos musicais devem possuir, a saber:

1. sistemas de segmentação automática de fluxos musicais devem fornecer subsídios para que os diversos componentes da música sejam considerados no processo de segmentação.
2. O sistema deve permitir a participação do analista no processo de segmentação, seja fornecendo informação para o sistema, seja interferindo no processo, rejeitando segmentações intermediárias consideradas inconsistentes. É necessário então que exista algum mecanismo de interação entre o analista e o sistema.

3. O sistema deve permitir que várias metodologias de segmentação “coabitem” em um mesmo ambiente. Esta característica, aliada à possível intervenção do analista no processo, pode melhorar muito o resultado obtido pelo sistema.
4. A segmentação deve tentar partir de um outro nível de estruturação da música. Como apontado anteriormente, as abordagens atuais partem das notas, em uma abordagem *bottom-up*, quando outras marcas de mais alto nível podem ser consideradas para a segmentação. Deve-se então tentar uma abordagem *top-down*. É possível que uma abordagem como esta possibilite a segmentação de passagens polifônicas.

## 2.3 Análise Orientada a Objetos Sonoros

Nesta seção apresenta-se a metodologia de segmentação considerada nesta dissertação: a Análise Orientada a Objetos Sonoros [Gui97b; Gui97a]. Ela se baseia no conceito de objeto sonoro e estabelece um critério para a realização da segmentação: a ruptura de continuidade.

Um objeto sonoro pode ser definido como “uma estrutura complexa descrita pela interação de vários componentes da escrita musical e cuja articulação é capaz de conter a forma como um todo ou em parte” [Gui97a]. A Figura 2.8 ilustra esta definição, mostrando os componentes de um objeto sonoro. Estes componentes são, na verdade, operações sobre valores que representam as notas, as durações, os andamentos, as intensidades, etc. de um objeto sonoro  $O$ <sup>11</sup>. Aqui, três componentes principais podem ser identificados:  $S$ ,  $T$  e  $q$ . O primeiro,  $S$ , descreve o objeto sonoro a partir do eixo espacial (estrutura acrônica), isto é, é um valor que representa a forma como o espaço—alturas disponíveis—é ocupado pelo objeto. O segundo componente,  $T$ , descreve o objeto sonoro a partir do eixo temporal (estrutura diacrônica), isto é, trata-se de um valor que representa a forma como o tempo é ocupado pelo objeto. O terceiro e último componente,  $q$ , descreve as qualidades sonoras de um objeto a partir de uma descrição matemática das propriedades acústicas de um determinado instrumento.

A análise de uma obra musical a partir desta metodologia passa por uma releitura de sua superfície visando identificar as unidades elementares existentes na obra, isto é, os objetos

---

<sup>11</sup>Este conjunto de operações encontra-se parcialmente implementado como uma biblioteca de funções para o aplicativo *OpenMusic* [Ago98; Had99; ARL<sup>+</sup>99].

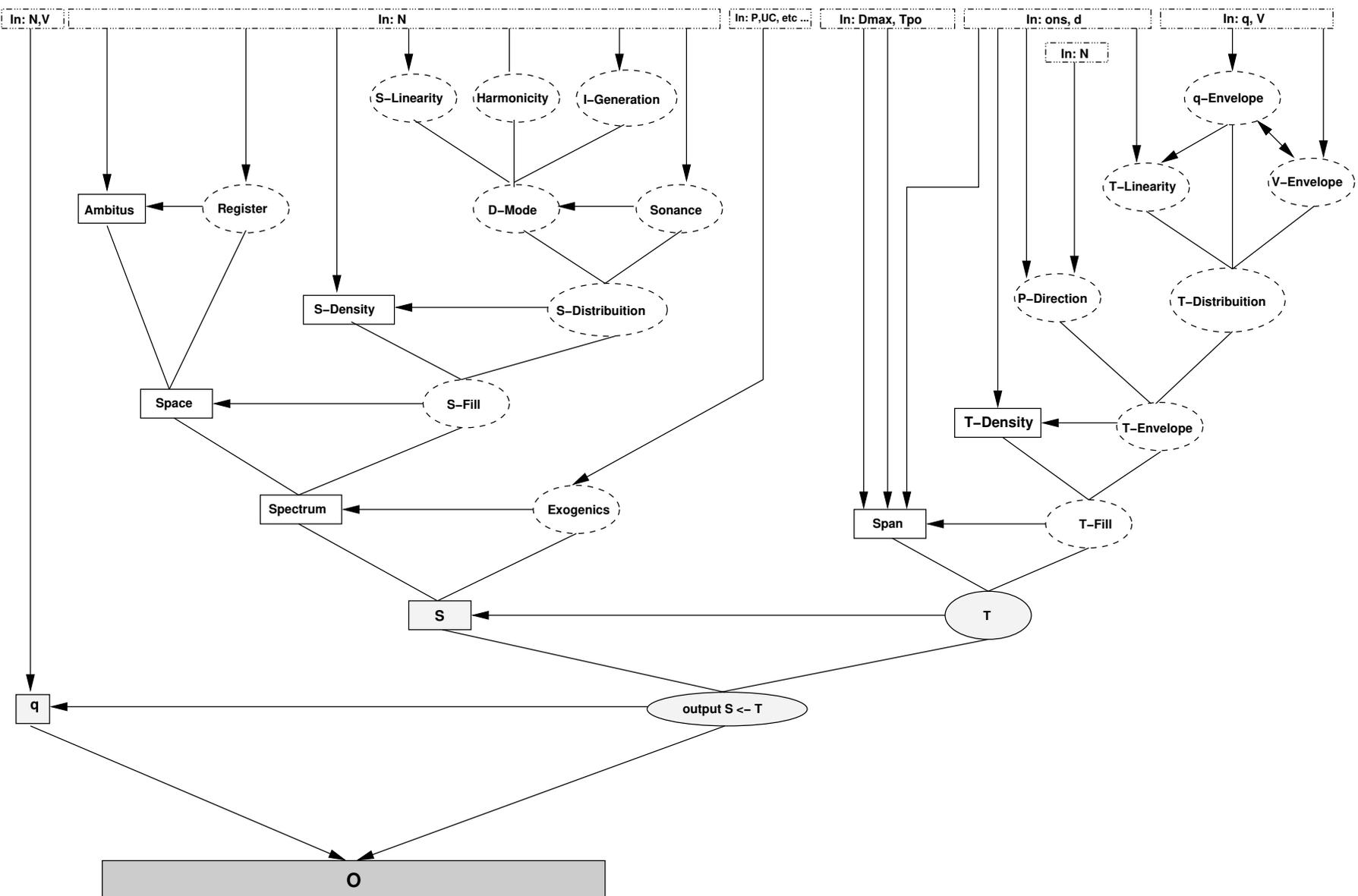


Figura 2.8: Os componentes de um objeto sonoro

sonoros, que, por concatenação linear e/ou hierárquica, geram a forma. “Sendo o objeto sonoro o produto da combinação de um número variado de componentes, a ruptura na continuidade estrutural de ao menos um de seus componentes implica em uma ruptura na continuidade sonora e, conseqüentemente, identifica uma nova etapa estrutural, isto é, um novo objeto” [Gui97b]. Assim, segmentar nesta metodologia de análise significa encontrar as rupturas de continuidade e, em conseqüência disto, as unidades lógicas—objetos sonoros—decorrentes destas rupturas.

Todavia, estes componentes ou parâmetros do objeto sonoro não são equivalentes na capacidade de gerar rupturas no contínuo, nem linearmente permutáveis. Eles agem sobre níveis relativamente globais que variam de acordo com a técnica de escritura do compositor, da obra ou do contexto local. Por exemplo, os parâmetros relevantes para a identificação dos objetos sonoros em uma determinada obra podem ser silêncios, mudanças na pulsação básica e mudanças na instrumentação. Em uma outra obra tais parâmetros podem não mais fazer sentido e parâmetros como mudanças de fraseado e/ou articulação, rupturas de intensidade e de registro podem ser mais significativos. Cabe ao analista decidir que parâmetros devem ser levados em consideração para uma boa segmentação.

Em linhas gerais, um procedimento adequado para a automação segmentação nesta metodologia de análise musical deve levar em consideração as seguintes características [TGF00a; TGF00b]:

1. A partir de uma entrada de dados musicais (arquivos MIDI, descrições gráficas ou sonoras, arquivos áudio, por exemplo), tenta-se encontrar rupturas do contínuo musical. Os critérios para a determinação de rupturas podem, em geral, ser dos seguintes tipos:
  - (a) Macro rupturas:
    - i. silêncios, isto é, interrupções do som;
    - ii. mudanças macro formais: mudanças de tempo, de unidade métrica, etc.
    - iii. mudanças globais de sonoridade: alterações na estrutura sonora, na distribuição ou na densidade instrumental (ações comumente atribuídas ao parâmetro orquestração); alterações globais na produção individual do som (início e fim de uso de pedais no piano, surdinas em instrumentos de metais, mudança de

- registro no órgão, *pizzicato* e outras modalidades de produção de som nos instrumentos de arco, etc);
- iv. final ou início de uma articulação global—geralmente indicada por ligaduras ou semelhantes;
  - v. rupturas globais de intensidades;
  - vi. rupturas globais de registro.
- (b) Mudança na configuração de outros componentes do som: densidade relativa, distribuição espacial e/ou temporal do som, direcionalidade, marcas de articulação, padrões de pulsação, etc.
2. Cada peça—ou seção de um peça, dependendo do escopo da análise—gera sua própria ordem na aplicação dos critérios de rupturas estruturais. O procedimento deve ser capaz de inferir esta ordem. Além disso, nem todos os critérios *supra* citados são válidos para qualquer música. O procedimento deve permitir que o conjunto de critérios a ser aplicado seja de alguma forma determinado.
  3. Uma primeira análise do arquivo de entrada deve permitir ao algoritmo definir, para cada critério relevante, o valor interválico mínimo<sup>12</sup>, a partir do qual será considerada a existência de uma ruptura. Obviamente, quanto menor este intervalo mais microscópica será a análise.
  4. O procedimento deve ter como resultado dois grupos de dados:
    - (a) O resultado da segmentação, na forma de seqüências lineares de unidades lógicas (este resultado pode ser representado por uma coleção de pequenos arquivos MIDI ou arquivos áudio, por exemplo).
    - (b) Um arquivo que mostre em que proporção um objeto *B*, resultante da segmentação, entra em relação de descontinuidade com o objeto imediatamente anterior *A*, sendo que esta proporção é quantificada em relação à média da avaliação das

---

<sup>12</sup>Intervalo aqui não possui o sentido usual em música, isto é, o da distância entre duas notas. Aqui intervalo significa a medida a partir da qual se vai considerar um dos eventos citados acima como uma ruptura de continuidade. No caso dos silêncios, por exemplo, o algoritmo será capaz de inferir a partir de que momento o silêncio será uma ruptura - uma pausa de semínima, de colcheia, 1 segundo de silêncio, 0,5 segundos, etc.

descontinuidades observadas no conjunto das listas de valores obtidas através da análise dos parâmetros de nível inferior (aqueles que, conjuntamente, constituem o objeto sonoro), no total da obra e não só entre os objetos *A* e *B*.

É necessário dizer que esta metodologia possui características que outras metodologias não possuem, características estas que facilitam a automação do processo de segmentação:

1. esta metodologia possui um critério de segmentação bem definido: a ruptura de continuidade;
2. ela possui uma lista definida de parâmetros onde se pode verificar a existência de rupturas de continuidade;
3. os parâmetros são objetivos e quantificáveis.

### 2.3.1 Um Modelo Mínimo para a Análise Orientada a Objetos Sonoros

A Análise Orientada a Objetos Sonoros, apesar de bastante clara em seus procedimentos, possui alguns parâmetros que dificultariam a implementação de um sistema capaz de realizar a segmentação nesta metodologia, como a mudança na orquestração, por exemplo. Desta forma, construiu-se um modelo mínimo capaz de segmentar consistentemente um fluxo musical a partir desta metodologia. A solução apresentada nesta dissertação (*vide* Capítulo 4) baseia-se neste modelo mínimo e, por hora, aplica-se exclusivamente à Música Clássica composta para piano no século XX. Este modelo foi obtido a partir de observações empíricas de como as análises são realizadas por especialistas humanos. Neste modelo mínimo, apenas os parâmetros *silêncio*, *pedal*, *registro*<sup>13</sup>, *densidade de notas* e *intensidades* são considerados. Verificou-se também que os parâmetros *silêncio* e *pedal* devem ser aplicados em primeiro lugar e nesta ordem, pois se constituem nos parâmetros que apontam elementos de mais alto nível na estrutura musical, não causando, assim, prejuízos para o bom andamento da segmentação<sup>14</sup>.

<sup>13</sup>Registros são divisões particulares no conjunto de notas de um instrumento, neste caso o piano.

<sup>14</sup>Prejuízo, aqui, significa que a ordem estabelecida—silêncio e pedal em primeiro lugar—não levará, *a priori*, a nenhum tipo de inconsistência ou “deformidade” em futuras segmentações.

Além disto, adotou-se o protocolo MIDI como forma de representação da música<sup>15</sup>. A esta representação associou-se uma semântica, construída a partir dos valores MIDI, sobre a qual a segmentação é baseada. No parâmetro *intensidade*, ou *velocidade* no jargão MIDI, os valores associados à notação tradicional encontram-se na tabela 2.2. Para o parâmetro *registro*, os registros do piano são determinados a partir da tabela 2.3, que se encontra de acordo com [Gui97b, p. 211]<sup>16</sup>. Para o parâmetro *pedal*, foi utilizado o controlador 64—*sustain pedal*—com os valores 127, significando pedal ativado, e 0, significando pedal desativado.

Intensidades	Valores MIDI
ressonância	1
<i>pppp</i>	12—23
<i>ppp</i>	24—34
<i>pp</i>	35—45
<i>p</i>	46—58
<i>mp</i>	59—68
<i>mf</i>	69—81
<i>pf</i>	82—92
<i>f</i>	93—103
<i>ff</i>	104—115
<i>fff</i>	116—126
<i>ffff</i>	127

Tabela 2.2: Classificação das intensidades em relação à notação musical tradicional

Registros	Valores MIDI	Notas
-3	21—34	Lá0—Lá#1
-2	35—43	Si1—Sol2
-1	44—52	Sol#2—Mi3
0	53—77	Fá3—Fá5
1	78—86	Fá#5—Ré6
2	87—98	Ré#6—Ré7
3	99—108	Ré#7—Dó8

Tabela 2.3: Classificação dos registros do piano

<sup>15</sup>Mais informações sobre este protocolo podem ser obtidas no Apêndice A, p. 86.

<sup>16</sup>A notação aqui utilizada é a notação americana onde o dó central (261.625 Hz.) = dó 4.

# Capítulo 3

## Sistemas Multi-Agentes

No Capítulo 2, apresentou-se diversas considerações que devem ser tomadas como ponto de partida para a construção de sistemas de segmentação automática de fluxos musicais. A partir destas considerações, pode-se apontar Sistemas Multi-Agentes como uma boa ferramenta para a construção de sistemas deste tipo. Este Capítulo tem como objetivo justificar a utilização de Sistemas Multi-Agentes na construção de sistemas de segmentação. Para tanto, ele está organizado da seguinte maneira: em primeiro lugar, apresenta-se conceitos pertencentes à Inteligência Artificial e que estão diretamente relacionados ao trabalho realizado nesta dissertação, a saber: noções de Inteligência Artificial Distribuída, de agente e de Sistemas Multi-Agentes. Em seguida, justifica-se a utilização desta abordagem na concepção de sistemas de segmentação de fluxos musicais. Ao final do capítulo, apresenta-se um ambiente multi-agentes denominado MATHEMA, que servirá de modelo de ambiente a partir do qual será construída a solução para o problema em questão.

### 3.1 Inteligência Artificial Distribuída

Nos anos oitenta, “diversos desenvolvimentos provocaram o interesse em concorrência e distribuição por parte da comunidade da Inteligência Artificial: o desenvolvimento de computadores concorrentes mais poderosos, a proliferação das redes de computadores e a percepção de que muito da resolução de problemas e das atividades dos seres humanos envolvem grupos de pessoas” [BG88]. O resultado foi a criação de uma sub-área de pesquisa na Inteligência Artificial, denominada Inteligência Artificial Distribuída que,

via de regra, lida com problemas complexos e que podem demandar conhecimentos de diferentes domínios. Seu objetivo é “criar organizações de sistemas capazes de resolver problemas através de raciocínio, geralmente baseado em manipulação simbólica” [Fer95; Fer99] ou ainda “estudar o conhecimento e as técnicas de raciocínio que podem ser necessárias ou úteis para que agentes computacionais participem de sociedades de agentes” [Bit96].

Esta última citação levanta duas noções de importância fundamental para a Inteligência Artificial Distribuída: a existência de uma entidade computacional, denominada agente, capaz de resolver problemas, e a idéia de sociedades de agentes, que representam agrupamentos destas entidades. Estes conceitos serão explicados em maiores detalhes nas próximas seções.

## 3.2 Agentes

A idéia da existência de entidades computacionais que são capazes de resolver problemas é antiga dentro da Inteligência Artificial. McCarthy e Hayes, nos idos de 1969, já desejavam que programas de computadores possuíssem características que hoje são atribuídas a estas entidades: “o que queremos é um programa de computador que decida o que fazer através da inferência em uma determinada linguagem formal que uma certa estratégia que irá levá-lo à sua meta” [MH69].

Apesar da idéia não ser nova, ainda não existe uma definição única para a denominação dada a esta entidade: agente. Carl Hewitt já chegou a dizer que a pergunta “o que é um agente?” é tão embaraçosa para a comunidade de computação baseada em agentes, quanto a pergunta “o que é a inteligência?” para a comunidade de Inteligência Artificial<sup>1</sup>. Em geral, os autores definem o conceito de agentes de uma maneira vaga e tendo em mente o uso que eles farão do conceito. A definição mais ligada à origem do nome agente—relacionado com *ação*—talvez seja a de Russel e Norvig: “um agente é qualquer coisa que perceba o ambiente no qual está inserido através de sensores e atue sobre este ambiente através de efetadores” [RN95, p. 31].

Outra definição, esta bem detalhada, é a de Ferber: “um agente é uma entidade real ou

---

<sup>1</sup>Esta afirmação foi feita no Décimo Terceiro Workshop Internacional de Inteligência Artificial Distribuída e citada em [WJ95].

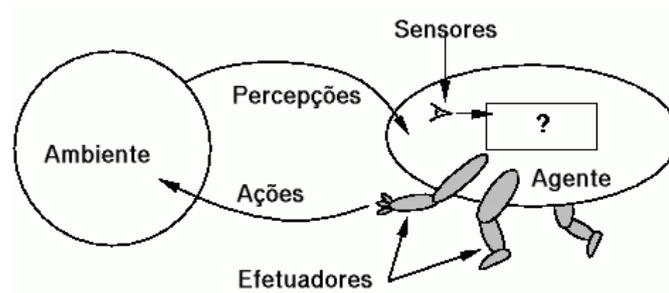


Figura 3.1: Um agente para Russel e Norvig

virtual capaz de agir em um ambiente e de se comunicar diretamente com outros agentes, guiado por um conjunto de tendências (na forma de objetivos individuais ou de uma função de satisfação/sobrevivência que ele tenta otimizar), possuidor de recursos próprios, com capacidades de percepção do ambiente no qual está inserido (mas até um determinado limite) e com uma representação parcial do ambiente (e talvez a nenhuma representação deste), dono de habilidades, capaz de oferecer serviços e de se reproduzir, e dono de um comportamento que tende à satisfação de seus objetivos, levando em conta os recursos e habilidades disponíveis e dependendo de sua percepção, suas representações e as mensagens que ele recebe” [Fer95; Fer99].

Franklin e Graesser, ao contrário de Ferber, tentaram encontrar uma definição mínima mas que possuísse a essência da noção de agente. Após uma análise de diversas definições de autores diferentes, eles chegaram à seguinte definição: “um agente autônomo é um sistema que se encontra dentro de um ambiente, sendo também uma parte dele, que percebe este ambiente e age sobre ele durante o tempo, seguindo uma agenda própria e de forma tal que o que ele percebe tenha efeito no futuro” [FG96]<sup>2</sup>.

Agentes possuem uma série de características ou propriedades<sup>3</sup>. Dentre elas pode-se destacar as seguintes [FG96; WJ95; RN95; HS98; LO98]:

- autonomia: agentes operam sem a intervenção direta de humanos ou outros agentes, possuindo algum tipo de controle sobre suas ações e estado interno.

<sup>2</sup>Segundo os autores, a autonomia é uma característica central da agência e por isto um agente é sempre um agente autônomo.

<sup>3</sup>Uma outra maneira de definir agente é através destas propriedades. Segundo [WJ95] um agente, em uma noção fraca, é um dispositivo de hardware ou de software que goza das quatro primeiras propriedades da lista que se segue.

- **Reatividade:** agentes percebem o ambiente no qual estão inseridos e respondem a mudanças que ocorram neste ambiente.
- **Comunicabilidade:** agentes interagem com outros agentes através de alguma linguagem de comunicação. Esta propriedade também é conhecida como habilidade social.
- **Orientação a metas (pró-atividade):** o comportamento dos agentes não é apenas uma mera resposta ao ambiente onde ele se encontra. Ele também é determinado pelas metas que o agente possui.
- **Adaptabilidade:** agentes com esta propriedade são capazes de modificar seu comportamento, baseado em sua experiência prévia.
- **Inteligência:** esta é uma propriedade que está longe de uma definição precisa, mas de maneira simplista pode-se dizer que um agente é inteligente quando ele é capaz de executar tarefas que, caso executadas por humanos, os fariam ser considerados inteligentes.
- **Racionalidade:** é uma propriedade relacionada com a anterior, mas de definição mais fácil. Agentes são racionais quando eles fazem a coisa certa. Por “coisa certa” entende-se que, em uma determinada situação, o agente escolherá a ação que o levará a um resultado melhor.
- **Mobilidade:** é a capacidade que um agente possui de se mover, de, por exemplo, se transportar de uma máquina para outra em uma rede. Um agente móvel, portanto, não está ligado ao sistema onde ele inicia sua execução. Esta capacidade de se transportar de um sistema para outro possibilita que o agente se mova para o sistema que contém objetos ou informações necessárias para que o agente complete sua computação.
- **Flexibilidade:** agentes com esta propriedade não possuem uma ordem de execução de suas ações previamente estabelecida.
- **Veracidade:** assume-se aqui que os agentes não comunicarão informação falsa ou errada.
- **Benevolência:** assume-se aqui que os agentes não possuem metas conflitantes e que, desta maneira, vão sempre tentar fazer o que lhes foi pedido.

Além de possuírem ou não as propriedades acima, os agentes podem ainda ser classificados de acordo com a sua estrutura interna, sua arquitetura. A arquitetura de um agente pode ser definida como uma “metodologia particular para a construção [de agentes]” [Mae91, citado em [WJ95]]. Ela pode ser de três tipos: deliberativa ou cognitiva, reativa e híbrida.

Agentes de arquitetura deliberativa são agentes que “contêm um modelo do mundo explícito e simbolicamente representado e cujas decisões (que ações tomar, por exemplo) são obtidas através de raciocínio lógico (ou ao menos pseudo-lógico), baseado em casamento de padrões e manipulação simbólica” [WJ95]. Este tipo de arquitetura se baseia na *Hipótese do Sistema de Símbolos Físicos* [NS76] e apresenta pelo menos dois problemas significativos: como traduzir o mundo real em uma descrição simbólica correta e adequada, e como representar informações deste mundo de forma que os agentes raciocinem sobre esta informação. Exemplos deste tipo de arquitetura são: STRIPS [FN71], *Intelligent Resource-bounded Machine Architecture* (IRMA) [BIP88], GRATE\* [Jen93] e BDI [RG91; RG95].

Agentes de arquitetura reativa são agentes que, ao contrário dos agentes deliberativos “não possuem nenhum tipo de modelo simbólico central do mundo e não utilizam nenhum raciocínio simbólico complexo” [WJ95]. Este tipo de pensamento origina-se na hipótese de que comportamento inteligente pode ser obtido a partir da “decomposição da inteligência em módulos geradores de comportamento individual, cuja coexistência e cooperação fazem emergir comportamentos mais complexos” [Bro90]. Um dos maiores problemas das arquiteturas reativas é a falta de uma metodologia de desenvolvimento: ao contrário dos agentes de arquitetura deliberativa, os agentes reativos não possuem uma metodologia de desenvolvimento associada conhecida por engenheiros de software e cientistas da computação. Além disso, “a natureza da inteligência reativa coloca a responsabilidade da decisão de como reagir aos vários estados do ambiente nas mãos do programador e não nos próprios agentes” [VW97]. Exemplos deste tipo de arquitetura são: arquitetura de subsunção (*subsumption architecture*) [Bro86], PENGI [AC87] e *Agent Network Architecture* (ANA) [Mae91].

Agentes de arquitetura híbrida são agentes que tentam integrar as duas abordagens anteriores. Diversos pesquisadores argumentam que abordagens puras, sejam deliberativas, sejam reativas, não são apropriadas para a construção de agentes. Aqui, a principal estratégia é a da “construção de um agente a partir de dois ou mais subsistemas” [WJ95]: um contendo

uma descrição simbólica do mundo—a parte deliberativa do agente—e outro que é capaz de reagir aos eventos do ambiente sem realizar qualquer tipo de raciocínio complexo—relativa à parte reativa do agente. Em geral, o resultado é uma arquitetura em camadas—*layers*—, sendo as camadas superiores responsáveis por atividades de maiores níveis de abstração. Um dos principais problemas neste tipo de arquitetura é a integração das diversas camadas, principalmente no que diz respeito às interações entre elas. Exemplos deste tipo de arquitetura são: *TouringMachines* [Fer92] e *InteRRap* [FMP96].

### 3.3 Sistemas Multi-Agentes

Viu-se na seção 3.2 definições, propriedades e maneiras de construção de entidades computacionais denominadas agentes. Em geral, estas entidades não se encontram isoladas, mas sim em “companhia” de outras entidades, de outros agentes, formando, desta forma, uma sociedade de agentes ou um Sistema Multi-Agentes. Assim, Sistemas Multi-Agentes são “sistemas computacionais nos quais dois ou mais agentes interagem ou trabalham em conjunto para realizar um conjunto de objetivos ou metas” [Les99]. A pesquisa nesta disciplina está centrada no “desenvolvimento de princípios e modelos computacionais para a construção, descrição, implementação e análise de padrões de interação e coordenação tanto em grandes sociedades de agentes, quanto nas pequenas sociedades” [Les99]. Estas duas características, interações entre agentes e coordenação de ações, serão expostas a seguir.

As possíveis interações entre os agentes são particularidades centrais em um Sistema Multi-Agentes. São, ao mesmo tempo, a origem de boa parte capacidade de resolução de problemas complexos, e a fonte de uma série de problemas. São a origem de grande parte da capacidade de resolução de problemas pois é a partir das interações entre os diversos agentes que problemas complexos podem ser resolvidos. É fonte de problemas pois é a partir dela que problemas como coordenação de ações e resolução de conflitos surgem.

Interações são eventos que fazem com que os agentes mantenham algum tipo de contato, seja este contato direto ou indireto—através de um outro agente ou do ambiente. Uma situação de interação é “um conjunto de comportamentos resultantes de um grupo de agentes que têm que agir em busca de seus objetivos, prestando atenção aos recursos disponíveis e a suas habilidades individuais” [Fer95; Fer99]. As diversas situações de interação podem ser clas-

sificadas em relação a três critérios: os objetivos ou intenções dos agentes, as relações destes agentes com os recursos que estão disponíveis e os meios ou habilidades que eles possuem para alcançar os seus objetivos. As diversas situações de interação podem ser classificadas da seguinte maneira:

- **colaboração simples:** consiste na simples adição de habilidades, não requerendo qualquer coordenação suplementar entre os agentes envolvidos. Ela ocorre quando os agentes possuem metas compatíveis, recursos suficientes e habilidades também suficientes. Este tipo de interação ocorre principalmente em sistemas onde a cada agente cabe uma parte do conhecimento, sendo necessário apenas que eles compartilhem este conhecimento para que um problema seja resolvido.
- **Obstrução:** este tipo de situação de interação ocorre quando os agentes possuem metas compatíveis, recursos insuficientes e habilidades suficientes, sendo característica de quando agentes se põem “à frente” de outros, não necessitando destes para completar suas tarefas, como por exemplo no trânsito e no controle de tráfego aéreo.
- **Colaboração coordenada:** este é o tipo de coordenação mais complexa. Ocorre quando os agentes possuem metas compatíveis, recursos insuficientes e habilidades também insuficientes. Isto faz com que mesmo a sociedade mais simples de robôs autônomos, por exemplo, tenha grandes problemas na alocação de tarefas e uso dos poucos recursos disponíveis.
- **Competição individual pura:** este tipo de situação de interação ocorre quando os diversos agentes possuem metas incompatíveis, recursos suficientes e habilidades também suficientes. É uma situação típica de uma corrida, por exemplo, onde os competidores possuem recursos e habilidades compatíveis, mas com metas incompatíveis.
- **Competição coletiva pura:** situação de interação que ocorre quando os vários agentes possuem metas incompatíveis, recursos suficientes e habilidades insuficientes. Como, neste caso, as habilidades não são suficientes, faz-se necessário, em um primeiro momento, que os agentes criem grupos e depois que os grupos se oponham uns aos outros. Exemplos característicos são as competições esportivas em grupo.

- Conflito individual por recursos: tipo de situação de interação que ocorre quando as metas dos agentes são incompatíveis, os recursos são insuficientes e estes agentes possuem habilidades suficientes. Neste caso, como não existem recursos suficientes, os agentes entram em conflito. Um exemplo é a defesa de territórios no mundo animal.
- Conflito coletivo por recursos: metas incompatíveis, recursos insuficientes e habilidades insuficientes fazem com que este tipo de situação de interação ocorra. É a combinação de competição coletiva e conflitos individuais por recursos: coalizões lutam entre si para obter o domínio de algo. Todas as formas de guerra e competição industrial são exemplos desta situação.

Esta classificação encontra-se resumida na tabela 3.1.

Metas	Recursos	Habilidades	Tipos de situação	Categoria
Compatíveis	Suficientes	Suficientes	Independência	Indiferença
Compatíveis	Suficientes	Insuficientes	Colaboração simples	Cooperação
Compatíveis	Insuficientes	Suficientes	Obstrução	
Compatíveis	Insuficientes	Insuficientes	Colaboração coordenada	
Incompatíveis	Suficientes	Suficientes	Competição individual pura	Antagonismo
Incompatíveis	Suficientes	Insuficientes	Competição coletiva pura	
Incompatíveis	Insuficientes	Suficientes	Conflitos individuais por recursos	
Incompatíveis	Insuficientes	Insuficientes	Conflitos coletivos por recursos	

Tabela 3.1: Resumo da classificação das situações de interação

Outro problema que é particular dos Sistemas Multi-Agentes é a coordenação de ações. A ação é uma característica básica de um agente e, conseqüentemente, de Sistemas Multi-Agentes. Ela pode ser vista sob diversas perspectivas: uma ação pode ser uma transformação no estado global, um processo computacional, uma mudança de local ou ainda um comando. O problema básico quando se fala em ações em um Sistema Multi-Agentes é como os diversos agentes podem agir simultaneamente em prol de seus próprios objetivos, isto é, como organizar suas ações no tempo para que um determinado problema seja resolvido. De acordo com Ferber [Fer95], outras razões podem ser apontadas para que as ações sejam coordenadas:

1. um agente, em um determinado instante, pode necessitar de informações fornecidas por outro(s) agente(s). A coordenação de ações entre os agentes pode fazer com que uma determinada informação esteja disponível no menor tempo possível.
2. como os recursos são limitados, não existindo em quantidades infinitas, eles devem ser compartilhados de forma tal que otimizem as ações a serem realizadas e evitem, na medida do possível, conflitos por recursos.
3. A coordenação de ações minimiza os custos, uma vez que possibilita a redução de ações sem eficácia e evita ações redundantes.
4. A coordenação de ações permite a decomposição dos objetivos em sub-objetivos, fazendo com que os agentes possuam objetivos individuais diferentes mas interdependentes e em conformidade com o objetivo decomposto.
5. Algumas situações de interações indesejáveis podem ser evitadas através da coordenação de ações. A obstrução é um exemplo: ela pode ser minimizada através coordenação de ações que funcionaria, neste caso, como um gerente de recursos, evitando que agentes utilizem recursos por tempo maior que o necessário ou em um momento errado.

A coordenação de ações, ainda segundo Ferber [Fer95], pode ser feita de quatro formas principais:

1. coordenação por sincronização: é o tipo mais elementar de coordenação. Toda a coordenação é descrita com precisão, através de seqüências de ações, levando à sincronização na execução das ações. Esta técnica tem origem na disciplina de sistemas distribuídos.
2. Coordenação por planejamento: esta é a técnica mais tradicional de coordenação. Ela compreende duas fases: na primeira determina-se qual o conjunto de ações—plano—que devem ser tomadas para que uma meta seja alcançada, e na segunda, um dos planos é escolhido e executado. O problema é que, em ambientes dinâmicos, os planos podem necessitar de uma reformulação durante a sua execução. Além disso, os planos gerados pelos diversos agentes podem gerar conflitos de objetivos e/ou de recursos.

3. Coordenação reativa: é uma técnica mais recente de coordenação. Ela é utilizada em Sistemas Multi-Agentes formados por agentes reativos e faz uso das percepções e ações destes agentes, sem qualquer tipo de planejamento de ações. Esta técnica é adequada para contextos que apresentam evolução ou situações onde é difícil antecipar o que irá acontecer.
4. Coordenação por regras (*regulation*): técnica pouco descrita na literatura e que parte da idéia de formar regras de comportamento que tentam eliminar possíveis conflitos entre os agentes.

Além das interações e da coordenação de ações, vale salientar uma terceira característica de Sistemas Multi-Agentes: a comunicação entre agentes. A comunicação é a base das interações e da organização social, tanto em Sistemas Multi-Agentes, quanto para os seres humanos [Fer95; Fer99]. Ela é realizada através de sinais ou por meio de mediadores, afetando os agentes de alguma maneira. Uma abordagem muito utilizada na comunicação entre agentes, baseada em uma teoria filosófica da linguagem, é denominada atos de fala—*speech acts* [Sea69]. Neste tipo de abordagem, a comunicação é considerada como uma das possíveis ações dos agentes e se baseia em primitivas—*performatives*—como *inform*, *request* e *answer*. Os atos de fala possuem duas finalidades básicas: transferir informação para um agente (*inform* é um exemplo) e requerer que algum agente tome uma determinada ação (*request* é um exemplo). Uma implementação bastante conhecida de linguagem de comunicação baseada em atos de fala é KQML—*Knowledge Query and Manipulation Language* [FFMM94].

### 3.4 Segmentação Automática de Fluxos Musicais e Sistemas Multi-Agentes

Uma vez explicitada a noção de Sistemas Multi-Agentes e retomando as considerações apresentadas na Seção 2.2.5, pode-se apresentar algumas justificativas que conduzem para uma abordagem multi-agentes, quando da construção de sistemas para a segmentação de fluxos musicais.

Em primeiro lugar, para contemplar a necessidade de se levar em consideração diversos componentes da escrita musical durante a segmentação, o Sistema Multi-Agentes em questão pode ser construído de forma tal que cada agente que integra o sistema seja capaz de segmentar a música em um componente específico. Os agentes então apresentam seus pontos de vista, isto é suas versões para a segmentação, e, através de mecanismos de coordenação de ações, negociação e de resolução de conflitos, decidem qual segmentação considerar. Isto pode permitir que diversos componentes ou parâmetros da música sejam considerados no processo de segmentação.

Um segundo ponto, a incorporação do analista no processo de segmentação, pode ser alcançado quando se considera o analista como um agente do sistema<sup>4</sup>. Desta forma, o analista pode interferir no processo em seu benefício, seja expondo seus pontos de vista, fornecendo informações que ele julgue interessantes ou importantes, ou mesmo servindo como uma espécie de agente oracular, optando por uma ou outra segmentação (quando, em situações extremas, os mecanismos de negociação ou de resolução de conflitos não consigam resolver que segmentação considerar, por exemplo).

No que concerne a incorporação de diversas abordagens analíticas, pode-se apontar para a seguinte possibilidade: assumindo que um agente, ou mesmo um conjunto de agentes dentro da sociedade, é capaz de segmentar um fluxo musical de acordo com uma determinada metodologia de análise, e que estes agentes são capazes de se comunicar uns com os outros, é possível fazer com que diversas metodologias de segmentação “coabitem” em um mesmo ambiente, dado que será possível a troca de informações entre as sociedades ou agentes que representam as metodologias de análise.

Além disto, uma construção multi-agentes pode ainda permitir que o sistema ofereça ferramentas para que experimentações mais radicais sejam realizadas pelo analista, como por exemplo:

- a possibilidade da inserção de outros agentes, sem prejuízo para o funcionamento do sistema, permite que outros parâmetros possam ser incorporados ao sistema pelo próprio analista. É claro que, para tanto, é necessário que mecanismos ou ferramentas para a construção de agentes segmentadores estejam incorporados ao sistema.

---

<sup>4</sup>Isto pode ser feito sem prejuízo para a definição de agente (*vide*, por exemplo, a definição de agente de Russel & Norvig [RN95]).

- A modificação da política de coordenação ou do mecanismo de resolução de conflitos por parte do analista, permite que segmentações não cogitadas em uma análise mais tradicional possam, eventualmente, ser alcançadas. Mesmo com a possibilidade da existência de algumas segmentações inconsistentes, a quantidade de resultados diferentes e que podem fornecer subsídios para uma análise mais aprofundada, supera este “inconveniente”. Além disso, o analista, como parte ativa no processo, pode rejeitar resultados inconsistentes.

### 3.5 MATHEMA: um Arcabouço para a Construção de Sistemas Multi-Agentes

O MATHEMA é um sistema multi-agentes idealizado para servir de “modelo de ambiente interativo de aprendizagem baseado no computador” [Cos97]. Como tal, tem sido utilizado para a construção de Sistemas Tutores Inteligentes em diversos domínios do conhecimento, dentre eles Álgebra, Lógica, Geometria Euclidiana Plana e Harmonia Musical [CBF97; CTF97; Tei97; CTFC00]. A arquitetura geral do MATHEMA, vista na Figura 3.2, baseia-se na “integração de entidades humanas e artificiais dispostas a interagir cooperativamente” [Cos97, p. 46]. A interação é feita em prol da aprendizagem de uma das entidades, o Aprendiz Humano.

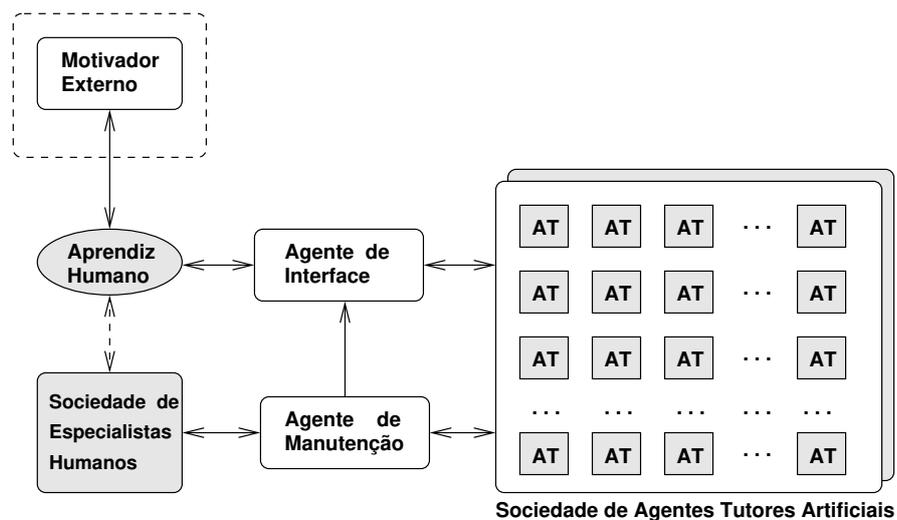


Figura 3.2: O MATHEMA

Sem entrar em maiores detalhes, pode-se descrever os componentes da arquitetura da seguinte maneira:

- **Aprendiz Humano (AH):** entidade humana que deseja aprender algo sobre um determinado domínio.
- **Sociedade de Agentes Tutores Artificiais (SATA):** conjunto de agentes que cooperam entre si a fim de promover a aprendizagem—aquisição de conhecimento—de um determinado AH. Cada agente pertencente à SATA possui conhecimento sobre um determinado aspecto do domínio. Desta forma, o conhecimento se encontra distribuído pela SATA.
- **Sociedade de Especialistas Humanos (SEH):** fonte externa de conhecimento integrada ao sistema. Ela funciona, na verdade, como uma entidade oracular, capaz de resolver problemas que a SATA não foi capaz de resolver. Ela também é responsável pela manutenção da SATA. Esta manutenção é feita através da inclusão ou exclusão de agentes ou ainda através da modificação do conhecimento dos agentes da SATA.
- **Agente de Interface (AI):** agente responsável pela observação do mundo exterior, sendo o elemento de ligação entre o sistema e o mundo. Aqui, ele é responsável pela ligação entre o Aprendiz Humano e a SATA.
- **Agente de Manutenção (AM):** agente responsável pela ligação entre a SEH e a SATA. É ele que oferece os meios para a SEH fazer a manutenção da SATA.
- **Motivador Externo (ME):** componente da arquitetura que representa entidades humanas responsáveis pela motivação do Aprendiz Humano. Pode ser um professor ou seus próprios colegas, por exemplo.

Uma situação hipotética possibilita a descrição genérica das interações que podem ocorrer no MATHEMA. Suponha que um certo Aprendiz Humano, incentivado por um Motivador Externo, deseja aprender sobre um determinado assunto. As interações do Aprendiz com o sistema começa por aquele informado este último de seus objetivos. Isto é feito através do Agente de Interface. Este, por sua vez, analisa os objetivos do Aprendiz e o ajuda a escolher um agente em particular na SATA que será o Agente Supervisor do Aprendiz. O

Agente Supervisor, uma metáfora do orientador, passa a ser o responsável pelo Aprendiz, oferecendo-lhe orientação e suporte pedagógico necessários ao Aprendiz. A situação mais simples é quando o próprio Supervisor é capaz de responder ao Aprendiz. A complexidade das interações aumenta à medida que outros agentes são necessários para que o Aprendiz seja atendido. Neste caso é a cooperação entre os diversos agentes da *SATA* que faz com a aprendizagem ocorra. É possível, no entanto, que a *SATA* não seja capaz de atender à requisição do Aprendiz. Neste caso, o Supervisor avisa, em primeiro lugar, o Aprendiz que o sistema não é capaz de responder à sua pergunta e, em seguida, notifica a *SEH*, através do Agente de Manutenção, que uma determinada requisição do Aprendiz não pode ser atendida. A *SEH* trata então de resolver o problema não resolvido e, através do Agente de Manutenção, informa o Supervisor que o problema foi resolvido e que ela está disponível para interagir com o Aprendiz. Eventualmente, caso necessário, a *SEH* promove uma operação de manutenção na *SATA*.

Pelo cenário descrito, pode-se perceber que os agentes mais complexos do sistema são os agentes que compõem a *SATA*. Eles, além de se tornarem eventualmente supervisores das atividades do aprendiz, são responsáveis pela cooperação em prol deste aprendiz. A *SATA* é, portanto, “um Sistema Tutor Multi-Agentes constituído por uma coleção estruturada de agentes capazes de, através de linguagens e protocolos estabelecidos, cooperarem entre si e, em alguns casos, também com a Sociedade de Especialistas Humanos (*SEH*)” [Cos97, p. 52]. Os agentes da *SATA* são do tipo deliberativo e possuem as seguintes propriedades: benevolência, autonomia, orientação a objetivos, comunicabilidade e habilidade social. A arquitetura destes agentes é a mostrada na Figura 3.3.

Como se pode notar, um agente da *SATA* possui três camadas:

1. Sistema Tutor: camada responsável pela interação direta com o aprendiz humano. Ela, se vista isoladamente, é um sistema tutor inteligente. Aqui estão os conhecimentos que o agente possui para resolver problemas o para realizar outras operações pedagógicas no domínio em questão.
2. Sistema Social: esta é a camada que viabiliza o comportamento cooperativo entre os agentes, inclusive entre os agentes da *SATA* e da *SEH*. Ela é formada por bases de conhecimento e mecanismos de raciocínio que possibilitam que o agente decida com

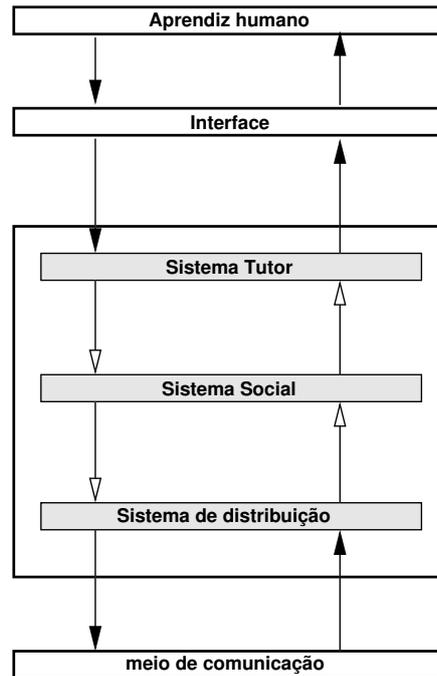


Figura 3.3: Um agente MATHEMA

quem ele deve cooperar para promover a resolução de um determinado problema, neste caso a aprendizagem.

3. Sistema de Distribuição: camada responsável pelo envio/recebimento de mensagens do agente da *SATA*. Além disso, ele também é responsável pelo gerenciamento da distribuição das mensagens internas do agente

Apesar de ter sido concebido para servir de arcabouço para a construção de Sistemas Tutores Inteligentes em diversos domínios, o MATHEMA já foi utilizado para outros fins: como ambiente de auxílio ao projetista de sistemas [CPF96] e para a construção de uma ferramenta de auxílio à gerência de redes de computadores [BC00].

No primeiro caso, o MATHEMA serviu como arcabouço para a construção de um sistema de auxílio à construção de modelos de sistemas de controle de tráfego. Este sistema, cuja arquitetura encontra-se ilustrada na Figura 3.4, funciona da seguinte maneira: o projetista fornece ao sistema uma representação que descreve as diversas direções e possíveis interseções nas vias a serem construídas. O sistema então decompõe esta representação em seções<sup>5</sup> e identifica que agentes sabem como projetar cada seção ou conjunto de seções. O

<sup>5</sup>Uma seção é um modelo dos diferentes padrões existentes em um sistema de controle de tráfego.

conhecimento do agente é representado por uma rede de Petri, que modela cada tipo de seção e as invariantes destas seções.

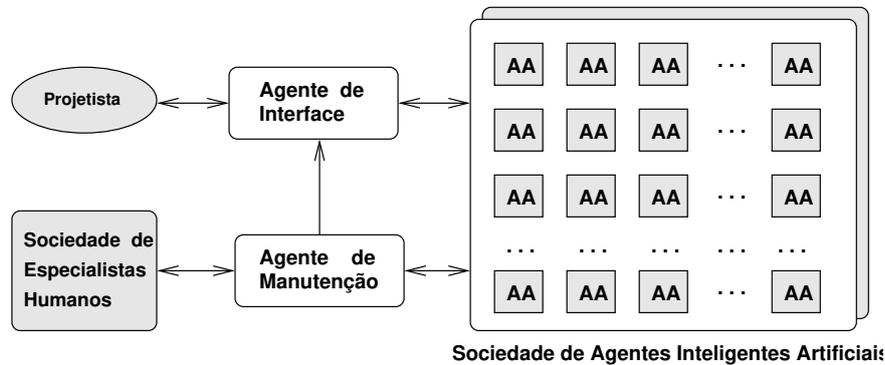


Figura 3.4: Adaptação do MATHEMA—auxílio ao projetista de sistemas

No segundo caso, o MATHEMA foi utilizado para a construção de um sistema de auxílio à gerência de redes de computadores. Este sistema, ilustrado na Figura 3.5 parte do princípio que algumas das atividades de um gerente de redes são repetitivas e rotineiras, e, assim, passíveis de automatização. Ele possui três tipos de agentes: o Agente Local (AL), o Agente Social (AS) e o Agente Remoto (AR). O Agente Local funciona como *interface* para o gerente. É através deste agente que o gerente da rede monitora o que está acontecendo e realiza tarefas de manutenção. O Agente Social é utilizado pelos agentes remotos para a sincronização dos dados enviados na rede. O Agente Remoto corresponde ao objeto gerenciável na rede (uma máquina ou um roteador, por exemplo) e tem como função o fornecimento de informações de monitoramento e controle, além de tratar as informações geradas pelo gerente da rede via Agente Local. Uma vez que o Agente Remoto representa os pontos de uma rede, em uma seção de gerência podem existir diversas instâncias deste agente.

Examinando estas duas adaptações e a versão original do MATHEMA, pode-se apontar características comuns às três versões:

- um agente que está interessado no que o sistema pode fazer (Agente Usuário): o aprendiz, nos sistemas tutores; o projetista de sistemas, no caso do sistema para o controle de tráfego; e o gerente de rede, no sistema de auxílio a gerência. Este agente é, via de regra, um humano.
- Uma sociedade de agentes capaz de resolver problemas em um determinado domínio, propostos pelo Agente Usuário (Sociedade de Agentes Artificiais): a SATA que é capaz

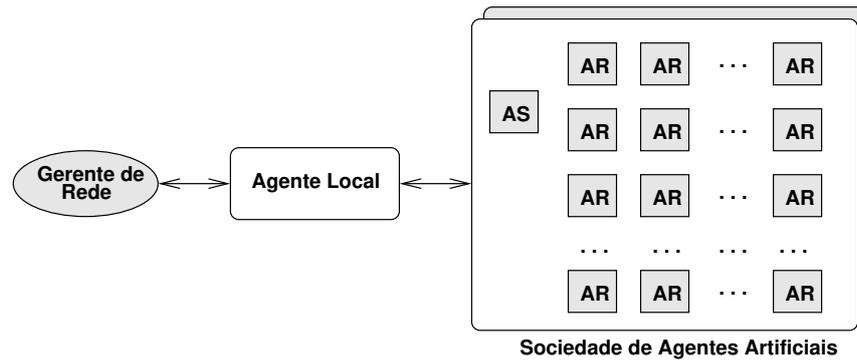


Figura 3.5: Adaptação do MATHEMA—gerência de redes de computadores

de promover a aprendizagem de um agente, no caso dos sistemas tutores; uma sociedade de agentes que é capaz de apresentar modelos de vias de tráfego, no sistema de controle de tráfego; e diversos Agentes Remotos, que são capazes de realizar tarefas de gerência em uma rede de computadores. Estes agentes, ao contrário do anterior, são agentes artificiais.

- Uma entidade, agente ou sociedade, responsável pela manutenção da Sociedade de Agentes Artificiais (Agente(s) de Manutenção): a Sociedade de Especialistas Humanos, nos sistemas tutores e no sistema de controle de tráfego; e o próprio gerente de rede, no sistema de auxílio à gerência. Como o primeiro, este é um agente humano.
- Agentes que funcionam como *interface* entre o Agente Usuário, o(s) Agente(s) Mantenedor(es) e o próprio sistema(Agentes de Interface): os Agentes de Interface e de Manutenção, nos sistemas tutores e no sistema de controle de tráfego; e o Agente Local, no sistema de auxílio a gerência de redes, incorporando as duas funções. Estes agentes são, normalmente, agentes artificiais.

Este modelo mínimo da arquitetura MATHEMA encontra-se ilustrado na Figura 3.6.

### 3.6 Considerações Finais

Viu-se neste capítulo conceitos pertencentes à Inteligência Artificial Distribuída, a saber: agentes e Sistemas Multi-Agentes. Do exposto e desprezando particularidades e problemas, pode-se destacar o seguinte:

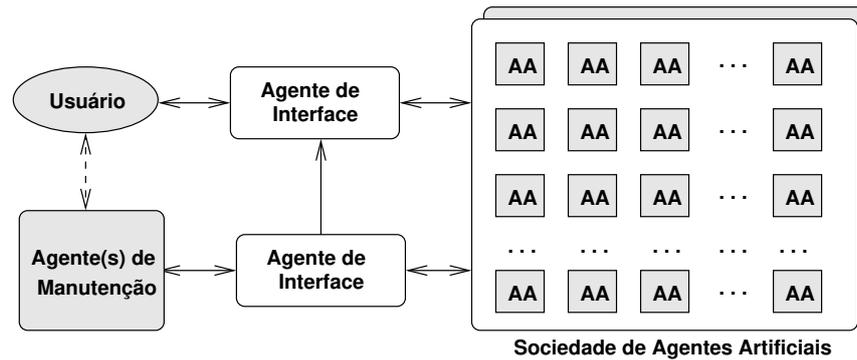


Figura 3.6: Modelo mínimo da arquitetura do MATHEMA

1. no âmbito da Inteligência Artificial Distribuída existe uma entidade computacional, denominada agente, que é capaz e de resolver problemas. Além desta capacidade, um agente possui diversas características, dentre elas pode-se destacar a autonomia, a comunicabilidade, a reatividade e a orientação a metas.
2. Estes agentes, via de regra, “vivem” junto de outros agentes, formando sociedade de agentes ou Sistemas Multi-Agentes. Dentro destas sociedades, os agentes são capazes de interagir e de se comunicar com outros agentes.
3. Um arcabouço para a construção Sistemas Multi-Agentes, denominado MATHEMA, incorpora a idéia de agentes humanos e artificiais cooperando em prol da solução de um problema.

## Capítulo 4

# SOS: um Protótipo para a Segmentação Orientada a Objetos Sonoros

Neste capítulo apresenta-se a solução adotada para a automação do processo de segmentação de fluxos musicais. Em primeiro lugar, descreve-se o processo de segmentação segundo a Análise Orientada a Objetos Sonoros. Em seguida, apresenta-se os algoritmos de segmentação desenvolvidos para contemplar o processo de segmentação descrito. Por fim, descreve-se o SOS, sigla construída a partir de Sistema para a Segmentação Orientada a Objetos Sonoros, que é o sistema de segmentação automática. Nesta descrição inclui-se a modelagem do sistema, a explicação das atribuições de cada um dos agentes que o compõe, assim como a arquitetura destes agentes.

### 4.1 O Processo de Segmentação

Nesta Seção descreve-se, por meio de um cenário e sem entrar em pormenores, o processo de segmentação segundo a Análise Orientada a Objetos Sonoros. Suponha que um analista deseja segmentar uma obra qualquer. Na página 22, viu-se que os parâmetros *silêncio* e *pedal* “devem ser aplicados em primeiro lugar e nesta ordem”. Assim, o processo de segmentação, de início, passa pela aplicação destes parâmetros<sup>1</sup>. Este primeiro resultado é denominado macro-segmentação, uma vez que os parâmetros em questão identificam marcas de nível superior na estrutura musical. No exemplo (Figura 4.1), o resultado hipotético da aplicação

---

<sup>1</sup>Os algoritmos de segmentação em cada um dos parâmetros serão descritos em detalhes na Seção 4.2

dos parâmetros *silêncio* e *pedal* é constituído de 4 macro-segmentos.

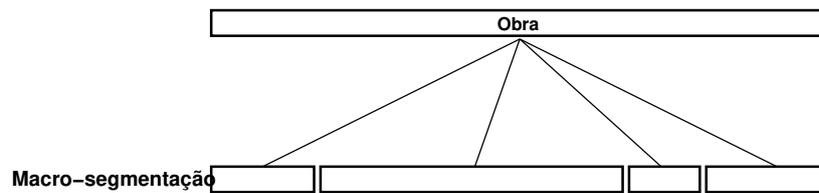


Figura 4.1: Macro-segmentação

O próximo passo é denominado proto-segmentação. Ele consiste na identificação sucessiva de locais onde possíveis segmentações posteriores podem ocorrer. Isto é feito em cada um dos macro-segmentos. No exemplo (Figura 4.2), quatro proto-segmentos foram identificados para o primeiro macro-segmento, seis para o segundo e dois para o terceiro e o quarto proto-segmentos. Isto significa que no primeiro macro-segmento existem três locais onde segmentações podem ocorrer, no segundo macro-segmento existem cinco e nos dois últimos macro-segmentos existe um local em cada um deles.

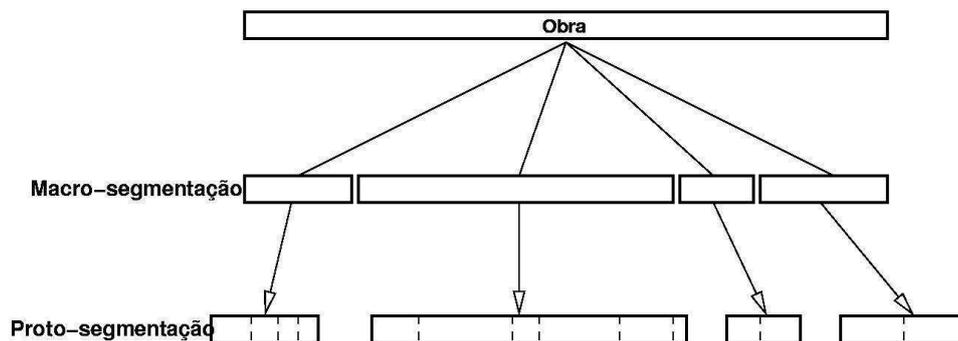


Figura 4.2: Proto-segmentação

Ao contrário do que acontece com os parâmetros *silêncio* e *pedal*, não existe uma ordem predefinida na aplicação dos parâmetros restantes, a saber: *densidade de notas*, *velocidades* e *registro*. Posto isto, o processo de segmentação prossegue da seguinte maneira: o primeiro macro-segmento é avaliado nos três parâmetros restantes. Possivelmente, o resultado das segmentações em cada um dos parâmetros será diferente. Decide-se, através de uma análise comparativa de cada uma das versões, que parâmetro deve ser considerado em primeiro lugar<sup>2</sup>. Uma vez eleito o parâmetro, observa-se se é possível continuar o processo com os

<sup>2</sup>Os pormenores do processo de decisão da ordem de aplicação dos parâmetros serão explicados mais adiante, na Seção 4.5.

segmentos resultantes, isto é, verifica-se se nos segmentos resultantes ainda existe alguma marca relativa à proto-segmentação. Caso exista, o processo continua de forma análoga, exceto pela não inclusão do parâmetro já utilizado nesta continuação<sup>3</sup>.

Voltando ao exemplo hipotético, neste primeiro estágio, após o exame das diferentes versões da segmentação, decidiu-se pela segmentação resultante da aplicação do parâmetro *velocidades*, cujo resultado é formado por dois segmentos (Figura 4.3). Note que o primeiro dos segmentos gerados não possui local onde se possa realizar uma segmentação. Assim, como não é mais possível continuar a segmentá-lo, o processo, neste segmento, pára.

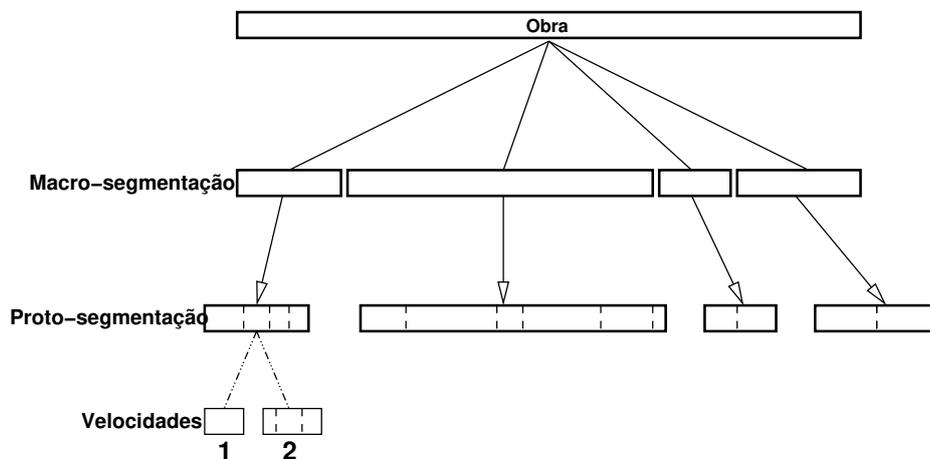


Figura 4.3: Resultado da primeira segmentação

Mas, como dito anteriormente, o processo continua caso exista algum segmento com marcas resultantes da proto-segmentação. Este é o caso do segundo segmento 2 (Figura 4.3). A segmentação continua, então, com a análise do segundo segmento nos parâmetros restantes (neste caso específico, *densidade de notas* e *registro*) e com uma nova decisão de que parâmetro considerar. Aqui, o parâmetro escolhido foi o parâmetro *densidade de notas* (Figura 4.4) e sua aplicação gerou mais uma vez dois segmentos.

Coincidentemente, como no caso anterior, o segmento 2a não pode ser segmentado pois não apresenta locais onde uma segmentação possa ser realizada. Já o segmento 2b, por ainda possuir uma marca resultante da proto-segmentação, pode continuar a ser segmentado. Como só resta um parâmetro a ser analisado (*registro*), ele é aplicado ao segmento 2b. Note que, apesar de ainda possuir um local de possível segmentação, não foi possível segmentar o

<sup>3</sup>Isto acontece porque não faz sentido segmentar com um mesmo parâmetro repetidas vezes. Admite-se que a segmentação em um determinado parâmetro deve esgotar as possibilidades de aplicação deste.

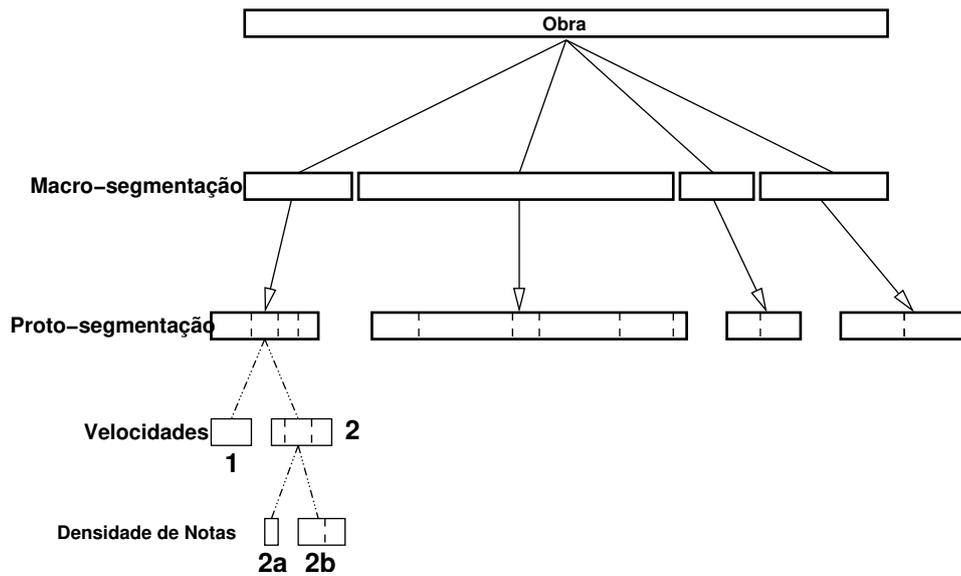


Figura 4.4: Resultado da segunda segmentação

segmento *2b* a partir da aplicação do parâmetro *registro* (Figura 4.5).

A segmentação nos parâmetros *densidade de notas*, *velocidades* e *registro*, descrita acima, é repetida para cada um dos macro-segmentos gerados. Ela termina quando não mais houver macro-segmentos passíveis de segmentação.

O processo de segmentação segundo a Análise Orientada a Objetos Sonoros pode, desta forma, ser agrupado em três fases distintas:

1. macro-segmentação: esta é a primeira fase e consiste na aplicação dos parâmetros *silêncio* e *pedal*. Ela tem como função identificar estruturas de nível superior na obra analisada.
2. Proto-segmentação: segunda fase do processo. Nela tenta-se identificar, nos resultados advindos da fase anterior, locais onde eventuais segmentações futuras podem ocorrer.
3. Segmentação: esta é a terceira fase do processo e consiste na aplicação sucessiva dos parâmetros *densidade de notas*, *velocidade* e/ou *registro* nos macro-segmentos já proto-segmentados.

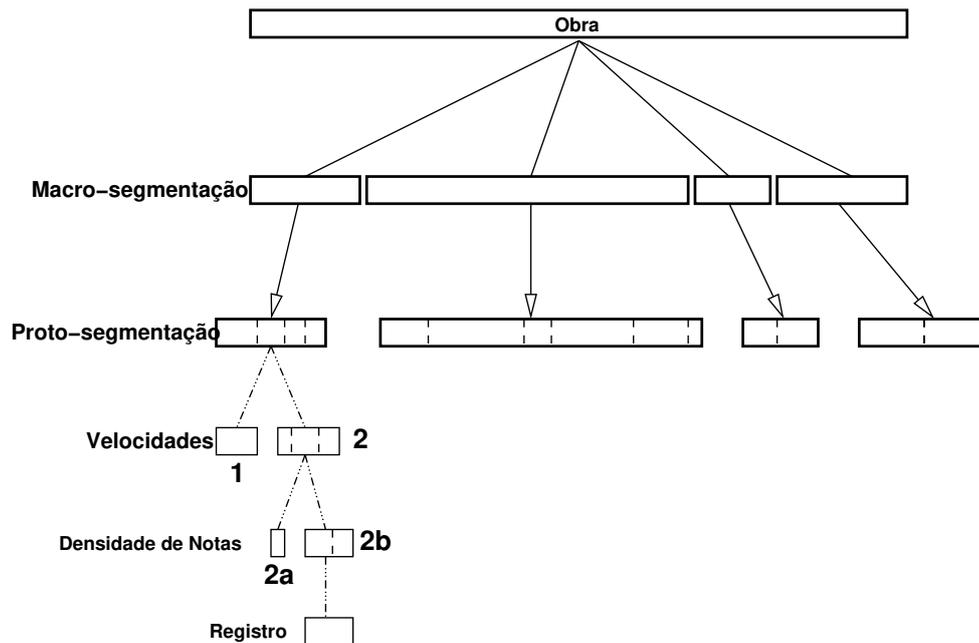


Figura 4.5: Resultado da terceira segmentação

## 4.2 Algoritmos de Segmentação

Na tentativa de construir a segmentação em uma abordagem *top-down*, como exposto na Seção 2.2.5, desenvolveu-se alguns algoritmos para encontrar rupturas de continuidade em parâmetros de nível estrutural mais alto que a nota. Estes correspondem aos parâmetros especificados na Seção 2.3.1, a saber: silêncios, pedais, densidade de notas, registro e velocidades.

O primeiro algoritmo (Algoritmo 1) serve para encontrar rupturas de continuidade no parâmetro *silêncio*. Ele possui as seguintes entradas: um arquivo MIDI (*arquivo-midi*), que descreve a música a ser segmentada, e um intervalo de tempo (*parâmetro-tempo*), que é fornecido pelo analista e representa o menor silêncio a ser considerado durante a segmentação<sup>4</sup>. De posse destes dois dados, o que o algoritmo faz é percorrer o arquivo MIDI de entrada do início ao fim, tentando encontrar silêncios de duração maior que a duração especificada pelo analista. Isto é feito verificando a existência de silêncio entre duas notas consecutivas e se o tempo de duração do silêncio identificado é maior que o tempo fornecido pelo analista ( $INÍCIO(nota-seguinte) - FINAL(nota-atual) >$

<sup>4</sup>Esta entrada evita que o algoritmo considere silêncios muito pequenos, isto é, silêncios que não possuem qualquer significado musical para a análise.

*parâmetro-tempo*). Quando o silêncio for maior que o tempo indicado pelo analista, o arquivo é segmentado neste ponto e o procedimento continua até o fim do arquivo MIDI. O resultado é um conjunto de arquivos MIDI, ou, em casos extremos—quando não houver silêncios maiores do que o analista especificou—o próprio arquivo MIDI de entrada.

---

**Algoritmo 1** Identificação dos silêncios
 

---

**Entrada:** *arquivo-midi* e *parâmetro-tempo*  
*nota-atual*  $\leftarrow$  PRIMEIRA-NOTA(*arquivo-midi*)  
*nota-seguinte*  $\leftarrow$  PRÓXIMA-NOTA(*nota-atual*)  
**enquanto** não for fim do *arquivo-midi* **faça**  
     **se** INÍCIO(*nota-seguinte*) – FINAL(*nota-atual*) > *parâmetro-tempo* **então**  
         SEGMENTE(*nota-atual*,*nota-seguinte*)  
     **fim se**  
     **se** FINAL(*nota-seguinte*) > FINAL(*nota-atual*) **então**  
         *nota-atual*  $\leftarrow$  *nota-seguinte*  
         *nota-seguinte*  $\leftarrow$  PRÓXIMA-NOTA(*nota-seguinte*)  
     **senão**  
         *nota-seguinte*  $\leftarrow$  PRÓXIMA-NOTA(*nota-seguinte*)  
     **fim se**  
**fim enquanto**

---

O segundo algoritmo (Algoritmo 2) identifica rupturas de continuidade no parâmetro *pedal*. Ele é específico para a música para piano, uma vez que trata de uma tecnologia particular deste instrumento. Ao receber, como no algoritmo anterior, um arquivo MIDI, o algoritmo, em primeiro lugar, procura por um acionamento de pedal. Ao encontrar um acionamento, o arquivo é segmentado e o algoritmo passa a procurar por uma desativação do pedal. Ao encontrar uma desativação, o algoritmo segmenta mais uma vez o arquivo e passa a procurar por nova ativação do pedal. Este processo continua até o fim arquivo MIDI. Caso não exista nenhuma ativação de pedal no arquivo a ser segmentado, o arquivo resultante será idêntico ao de entrada.

Como apontado anteriormente (Seção 4.1), a identificação de rupturas nos parâmetros silêncio e pedal corresponde a uma fase do processo de segmentação denominada macro-segmentação. Este resultado, *a priori*, não é muito interessante para o analista, pois as marcas estruturais identificadas podem ser visualizadas diretamente na partitura. Todavia, no

**Algoritmo 2** Identificação de pedais

---

```

Entrada: arquivo-midi
status-esperado ← on
enquanto não for fim do arquivo-midi faça
  se PEDAL então
    se status ← status-esperado então
      SEGMENTE
      status-esperado ← Contrário de status-esperado
    fim se
  fim se
fim enquanto

```

---

processo de segmentação aqui idealizado, ele é importante pois constitui a base do processo *top-down*, uma vez que delimita as estruturas de nível superior que poderão ser posteriormente segmentadas.

O próximo algoritmo (Algoritmo 3) prepara o resultado da macro-segmentação para a identificação de rupturas nos parâmetros restantes. Denominada proto-segmentação, esta preparação consiste na identificação sucessiva de pontos onde uma segmentação futura poderá ocorrer. O algoritmo de proto-segmentação recebe o(s) arquivo(s) MIDI resultante(s) da macro-segmentação e tenta encontrar, em cada um dos arquivos, “interrupções” no fluxo de notas. Uma “interrupção” aqui significa a inexistência de sobreposição de notas, isto é, quando o início de uma nota não coincide com nenhuma nota já em execução, como ilustrado na Figura 4.6<sup>5</sup>. Quando isso ocorre, o algoritmo marca o local criando um proto-segmento.

A seguir, descreve-se os algoritmos que integram a terceira parte do processo descrito na Seção 4.1. Nela, segmenta-se os macro-segmentos devidamente marcados, isto é, proto-segmentados, a partir dos parâmetros *velocidades*, *densidade de notas*, e *registro*.

O primeiro destes algoritmos (Algoritmo 4) identifica rupturas de continuidade no parâmetro *densidade de notas*, isto é, ele identifica os locais onde ocorrem mudanças no número absoluto de notas. Em primeiro lugar, o que o algoritmo faz é contar o nú-

---

<sup>5</sup>Nesta representação, o eixo horizontal representa a duração de uma nota, enquanto o eixo vertical representa a altura da nota. O ponto 0 da linha vertical marca o dó central do piano—dó 4, na notação americana. Assim, quanto maior a faixa horizontal, maior é a duração da nota; quanto mais acima do ponto 0 da linha vertical mais aguda é nota; e quanto mais abaixo deste mesmo ponto, mais grave é a nota.

**Algoritmo 3** Determinação dos proto-segmentos

---

**Entrada:** *arquivo-midi*

**enquanto** não for fim do *arquivo-midi* **faça**

*nota1* ← PRÓXIMA-NOTA

*nota2* ← PRÓXIMA-NOTA-NÃO-SUPERPOSTA

PROTO-SEGMENTE(*nota1*,*nota2*)

**fim enquanto**

---

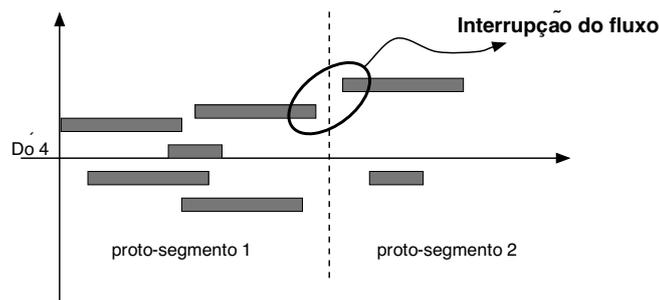


Figura 4.6: Exemplo de uma “interrupção” no fluxo de notas

mero de notas em proto-segmentos consecutivos (NÚMERO-NOTAS (*proto-segmento*) e NÚMERO-NOTAS (PRÓXIMO (*proto-segmento*))). Em seguida, ele segmenta o fluxo caso exista uma diferença significativa no número de notas dos proto-segmentos analisados.

Aqui cabem algumas explicações. Nos algoritmos anteriores (Algoritmos 1–3), a macro-segmentação, ou proto-segmentação, era realizada a partir da verificação de um determinado “fato” musical (a existência de um silêncio de duração previamente definida, uma ativação/desativação de pedal ou a inexistência de sobreposição de notas). Este “fato” era determinado ou conhecido antes do início do processo e valia para todos os casos. No caso da densidade de notas, não se pode determinar, *a priori*, o número máximo de notas que cada proto-segmento vai possuir, por razões óbvias—não se pode pressupor o número máximo de notas que um compositor vai utilizar. Também não se pode determinar um número único, que seja válido para todos os casos, a partir do qual se possa dizer que existe uma ruptura de continuidade na densidade de notas. Isto ocorre porque a percepção da ruptura se dá em um contexto local. Por exemplo, uma diferença de cinco notas pode não ser percebida como ruptura quando o contexto local é de cinquenta notas um proto-segmento e cinquenta e cinco em um outro. Mas, esta mesma diferença será certamente percebida como ruptura quando

o contexto local for uma nota em um proto-segmento e seis em outro. Assim, é necessário construir uma maneira de organizar o número de notas que seja tanto geral, quanto leve em consideração as mudanças de contexto. Uma maneira que se mostrou eficiente do ponto de vista musical, isto é, cujos resultados faziam sentido musical quando analisados, foi a construção de *grupos* de quantidades de notas determinados a partir da série de Fibonacci. Um grupo de notas  $G$  é determinado pelo intervalo  $[m, n - 1]$ , onde  $m$  e  $n$  são dois números consecutivos da série de Fibonacci. Assim, o primeiro grupo vai de 1 a 1, o segundo de 2 a 2, o terceiro de 3 a 4, o quarto de 5 a 7 e assim por diante, como mostra a Tabela 4.1.

Retornando à explicação do algoritmo, diz-se que uma diferença significativa no número de notas de dois proto-segmentos acontece quando os grupos relativos ao número de notas de cada proto-segmentos são diferentes. Por exemplo, suponha que um determinado proto-segmento  $A$  possua 4 notas e um outro proto-segmento  $B$  possua 7 notas. Observando a Tabela 4.1, nota-se que grupo associado ao número de notas do proto-segmento  $A$  é o grupo 3. Já para o proto-segmento  $B$ , o grupo associado é o grupo 4. Neste caso, o algoritmo irá segmentar entre o proto-segmento  $A$  e o proto-segmento  $B$ . Já se o proto-segmento  $A$  possuíse 3 notas e o proto-segmento  $B$  as mesmas 4, não ocorreria uma segmentação, uma vez que os proto-segmentos possuem um mesmo grupo associado, o grupo 3.

---

#### **Algoritmo 4** Identificação de rupturas no parâmetro *densidade de notas*

---

**Entrada:** *proto-segmentos*

*proto-segmento*  $\leftarrow$  PRIMEIRO(*proto-segmentos*)

**enquanto** existir *proto-segmentos* **faça**

*grupo1*  $\leftarrow$  GRUPO(NÚMERO-NOTAS(*proto-segmento*))

*grupo2*  $\leftarrow$  GRUPO(NÚMERO-NOTAS(PRÓXIMO(*proto-segmento*)))

**se** *grupo1*  $\neq$  *grupo2* **então**

SEGMENTE(*grupo1*,*grupo2*)

**fim se**

*proto-segmento*  $\leftarrow$  PRÓXIMO(*proto-segmento*)

**fim enquanto**

---

O próximo algoritmo (Algoritmo 5) identifica rupturas de continuidade no parâmetro *intensidade* ou *velocidade*. Como no caso da densidade de notas, é necessário estabelecer uma maneira que permita levar em consideração os contextos locais dos proto-segmentos. Assim,

Série de Fibonacci	Grupos
1	1
2	2
3 4	3
5 6 7	4
8 9 10 11 12	5
13 14 ...	6
...	...

Tabela 4.1: Série de Fibonacci e grupos associados

utiliza-se a *moda* das velocidades nos proto-segmentos para estabelecer a velocidade global de cada um destes. A moda representa, então, a velocidade de referência que será utilizada durante a segmentação. Ela é obtida pelo método OBTEM-VELOCIDADE: caso exista uma única moda, ele retorna esta moda; caso existam duas modas, o método retorna a maior delas; e caso não exista moda, o método retorna a média das velocidades encontradas no proto-segmento. O algoritmo, então, passa a verificar em que grupos se encontram os valores de referência obtidos (os grupos estão identificados na Tabela 4.2). Quando o algoritmo encontra diferença significativa na velocidade de referência de proto-segmentos consecutivos, isto é, quando os grupos dos proto-segmentos são diferentes ( $grupo1 \neq grupo2$ ), o fluxo é segmentado.

Intensidades	Valores MIDI	Grupos
ressonância	1	1
<i>pppp</i>	12—23	2
<i>ppp</i>	24—34	3
<i>pp</i>	35—45	4
<i>p</i>	46—58	5
<i>mp</i>	59—68	6
<i>mf</i>	69—81	7
<i>pf</i>	82—92	8
<i>f</i>	93—103	9
<i>ff</i>	104—115	10
<i>fff</i>	116—126	11
<i>ffff</i>	127	12

Tabela 4.2: Classificação das velocidades e grupos associados

O último algoritmo (Algoritmo 6) identifica rupturas de continuidade no parâmetro *registro*. Ele funciona de forma semelhante ao Algoritmo 5, com-

---

**Algoritmo 5** Identificação de rupturas no parâmetro *velocidades*

---

**Entrada:** *proto-segmentos*

*proto-segmento* ← PRIMEIRO(*proto-segmentos*)

**enquanto** existir *proto-segmentos* **faça**

*grupo1* ← GRUPO(OBTEM-VELOCIDADE(*proto-segmento*))

*grupo2* ← GRUPO(OBTEM-VELOCIDADE(PRÓXIMO(*proto-segmento*)))

**se** *grupo1* ≠ *grupo2* **então**

        SEGMENTE(*proto-segmento*, PRÓXIMO(*proto-segmento*)))

**fim se**

*proto-segmento* ← PRÓXIMO(*proto-segmento*)

**fim enquanto**

**OBTEM-VELOCIDADE**

**Entrada:** *proto-segmento*

*moda* ← MODA-VELOCIDADES(*proto-segmento*)

**se** NÃO-EXISTE-MODA(*moda*) **então**

    Retorne MÉDIA-VELOCIDADES(*proto-segmento*)

**senão**

**se** EXISTE-MAIS-DE-UMA-MODA(*moda*) **então**

        Retorne MAIOR-MODA(*moda*)

**senão**

        Retorne *moda*

**fim se**

**fim se**

---

parando, em proto-segmentos consecutivos, a quantidade de registros do piano que estes proto-segmentos ocupam ( $\text{REGISTROS-OCUPADOS}(\text{proto-segmento})$ ) e  $\text{REGISTROS-OCUPADOS}(\text{PRÓXIMO}(\text{proto-segmento}))$ ) e segmentando nos locais onde o número de registros ocupados é diferente. Os registros do piano estão identificados com a Tabela 4.3<sup>6</sup>.

---

**Algoritmo 6** Identificação de rupturas no parâmetro *registro*


---

**Entrada:** *proto-segmentos*

*proto-segmento*  $\leftarrow$  PRIMEIRO(*proto-segmentos*)

**enquanto** existir *proto-segmentos* **faça**

*número-reg1*  $\leftarrow$  GRUPO( $\text{REGISTROS-OCUPADOS}(\text{proto-segmento})$ )

*número-reg2*  $\leftarrow$  GRUPO( $\text{REGISTROS-OCUPADOS}(\text{PRÓXIMO}(\text{proto-segmento}))$ )

**se** *número-reg1*  $\neq$  *número-reg2* **então**

        SEGMENTE(*proto-segmento*,  $\text{PRÓXIMO}(\text{proto-segmento})$ )

**fim se**

*proto-segmento*  $\leftarrow$  PRÓXIMO(*proto-segmento*)

**fim enquanto**

---

Registros	Valores MIDI	Notas	Grupos
-3	21—34	Lá0—Lá $\sharp$ 1	1
-2	35—43	Si1—Sol2	2
-1	44—52	Sol $\sharp$ 2—Mi3	3
0	53—77	Fá3—Fá5	4
1	78—86	Fá $\sharp$ 5—Ré6	5
2	87—98	Ré $\sharp$ 6—Ré7	6
3	99—108	Ré $\sharp$ 7—Dó8	7

Tabela 4.3: Classificação dos registros do piano

### 4.3 Descrição Geral do Sistema

Uma vez descritos o processo de segmentação (Seção 4.1) e os algoritmos de segmentação (Seção 4.2), e levando-se em consideração o exposto na Seção 3.4, onde se aponta os Sistemas Multi-Agentes como ferramentas computacionais apropriadas para a construção de

---

<sup>6</sup>Esta tabela é a mesma que se encontra na página 23, repetida aqui por conveniência.

sistemas de segmentação de fluxos musicais, descreve-se, a partir de agora, o sistema de segmentação multi-agentes desenvolvido neste trabalho.

Este sistema recebeu o nome de SOS, sigla construída a partir de Sistema para a Segmentação Orientada a Objetos Sonoros, e tem como arcabouço o modelo mínimo da arquitetura MATHEMA, descrito na Seção 3.5. Os componentes deste modelo mínimo, ilustrados na Figura 3.6, sofreram adaptações para contemplar o problema da segmentação. Os componentes resultantes desta adaptação são:

1. *Agente Analista*: este agente representa o analista no sistema. É ele que submete ao restante dos agentes do sistema a obra a ser segmentada. Ele também é responsável pela escolha dos parâmetros que deverão ser considerados durante o processo de segmentação. Além disso, caso os outros agentes do sistema não sejam capazes de resolver que parâmetro considerar em um determinado momento, o Agente Analista é responsável por esta decisão. Este agente, portanto, exerce as funções que cabiam originalmente ao Agente Usuário e ao Agente de Manutenção<sup>7</sup>.
2. *Sociedade de Agentes Segmentadores Artificiais (SASA)*: esta sociedade é equivalente à *Sociedade de Agentes Artificiais* no modelo original. Ela é a parte do sistema que realiza a segmentação da obra submetida pelo Agente Analista. A SASA é formada por seis agentes<sup>8</sup>: um agente, chamado *Agente Mediador*, responsável pela coordenação de ações dos outros agentes da sociedade e pelas decisões inteligentes que o sistema é capaz de fazer; e cinco outros agentes, cuja função individual é encontrar rupturas em parâmetros específicos da música, a saber, silêncios (*Agente Silêncio*), pedais (*Agente Pedal*), densidade de notas (*Agente Densidade de Notas*), registro (*Agente Registro*) e velocidades (*Agente Velocidade*).
3. *Agente de Interface*: agente que faz a ligação entre o analista e a SASA. Este agente é capaz de traduzir para os agentes da SASA as informações do analista, tanto informações acerca da segmentação, quanto operações de manutenção que o analista possa vir a fazer.

---

<sup>7</sup>Assume-se que o analista possui habilidades e conhecimentos suficientes para realizar qualquer operação de manutenção (exclusão ou inclusão de agentes no processo, por exemplo).

<sup>8</sup>Para maiores detalhes sobre cada um dos agentes *vide* Seção 4.5

A arquitetura resultante destas modificações encontra-se ilustrada na Figura 4.7.

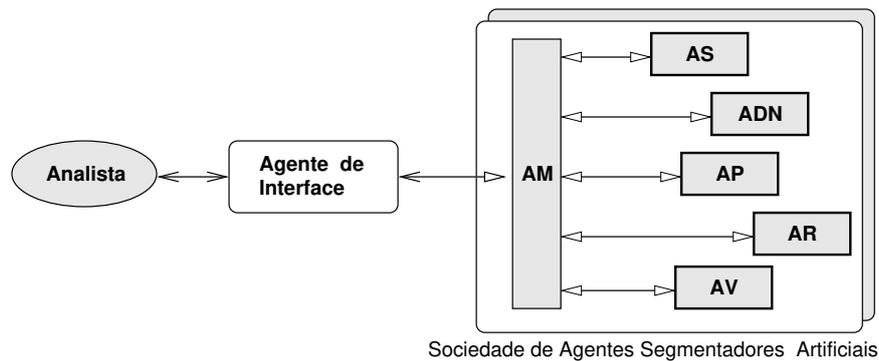


Figura 4.7: Arquitetura do SOS

Com base nesta descrição preliminar e na descrição do processo de segmentação feita da Seção 4.1, pode-se apontar um cenário hipotético ilustrando, em linhas gerais, o funcionamento do sistema: suponha que um analista deseja segmentar uma obra qualquer. Em primeiro lugar, ele fornece ao sistema, através do Agente Interface, a obra a ser segmentada. Esta obra é submetida ao Agente Mediador, que passa então a controlar o processo de segmentação. Seu primeiro passo é realizar a macro-segmentação, isto é, submeter a obra aos agentes responsáveis pela segmentação nos parâmetros silêncio e pedal (Agentes Silêncio e Pedal, respectivamente). Estes agentes, após realizarem a segmentação, devolvem seu resultado para o Agente Mediador. De posse da obra macro-segmentada, o Agente Mediador passa ao próximo estágio no processo de segmentação: a proto-segmentação. Neste estágio, o próprio Agente Mediador proto-segmenta cada um dos macro-segmentos obtidos no estágio anterior. No último estágio do processo, o Agente Mediador submete o resultado da proto-segmentação aos agentes restantes (Agentes Densidade de Notas, Velocidade e Registro). Cada um destes agentes retorna uma versão, uma segmentação<sup>9</sup>. O Agente Mediador decide, então, qual a segmentação mais apropriada<sup>10</sup>. Caso o Agente Mediador não seja capaz de decidir qual a segmentação mais apropriada, ele lança mão do analista, que decide então qual resultado considerar. Uma vez decidida a segmentação apropriada, o agente que a realizou é excluído do restante do processo, que se repete com os agentes restantes e com cada um dos segmentos resultantes. A segmentação termina quando não mais houver agentes

<sup>9</sup>Estas versões possivelmente serão diferentes entre si.

<sup>10</sup>Os pormenores deste processo de decisão são esclarecidos na Seção 4.5.

aptos a segmentar ou quando não houver mais proto-segmentos<sup>11</sup>.

## 4.4 Modelo do Sistema

Para a modelagem do SOS utilizou-se uma ferramenta de modelagem matemática chamada Redes de Petri [Mur89], que, segundo T. Murata, são ferramentas adequadas para “a descrição e estudo de sistemas de processamento da informação, caracterizados como concorrentes, assíncronos, distribuídos, paralelos, não-determinísticos e/ou estocásticos” [Mur89, p. 541]. Além disto, este formalismo foi escolhido por apresentar as seguintes características:

1. as Redes de Petri possuem uma notação gráfica associada, facilitando a compreensão dos modelos.
2. Sendo um formalismo matemático, as Redes de Petri permitem que propriedades sobre os modelos possam ser verificadas.
3. As Redes de Petri utilizadas nesta modelagem, as Redes de Petri Coloridas, possuem um conjunto de ferramentas que permitem a simulação dos modelos construídos, o *Design/CPN* [Met96b].

Sem entrar em detalhes, uma Rede de Petri é uma estrutura de rede e uma marcação inicial associada. Uma estrutura de rede é um grafo composto por dois tipos de nós: lugares (círculos, na representação gráfica) e transições (retângulos, graficamente). Estes nós são ligados por arcos direcionados, que representam o fluxo de dados—fichas—de uma Rede de Petri. A presença de zero ou mais fichas em um lugar é chamada marcação. Uma transição pode possuir dois tipos de lugares: lugares de entrada, representados pelos lugares que se situam nos extremos dos arcos que chegam a uma transição, e lugares de saída, representados pelos lugares nos extremos dos arcos que saem de uma transição. Uma transição está habilitada quando cada lugar de entrada possui no mínimo o número de fichas especificados nos arcos de entrada—peso do arco. A dinâmica de uma Rede de Petri é determinada pela regra de disparo, que diz que, estando uma transição habilitada, as fichas do lugar de

---

<sup>11</sup>A inexistência de proto-segmentos implica na impossibilidade de continuação do processo de segmentação, uma vez que segmentações só podem ocorrer nos locais marcados pela proto-segmentação.

entrada especificadas no arco serão retiradas deste e serão depositadas nos lugares de saída a quantidade de fichas especificadas nos arcos de saída<sup>12</sup>.

A esta definição, as Redes de Petri Coloridas agregam mais um elemento: a ficha passa a carregar informação, possuindo um tipo ou *cor*, o que possibilita a manipulação dos dados. Outra característica das Redes de Petri Coloridas é a possibilidade de construção hierárquica de modelos. Isto é possível relacionando diversas redes menores, cada uma delas modelando um aspecto do sistema. Na ferramenta de suporte à construção de modelos em Redes de Petri Coloridas, o *Design/CPN*, a hierarquização é realizada por meio de lugares e transições especialmente marcadas para tal. Elas fazem a ligação entre as diversas redes do modelo, que aqui são chamadas de *páginas*.

A modelagem do SOS utiliza-se desta capacidade de hierarquização. A Figura 4.8 mostra a *Página de Hierarquia* do SOS. Nela é possível observar todas as páginas existentes no modelo, assim como as ligações existentes entre estas páginas. O modelo contém as seguintes páginas: *SubmeteFluxo*, *MacroSegmentação*, *ProtoSegmentação* e *Segmentação*<sup>13</sup>.

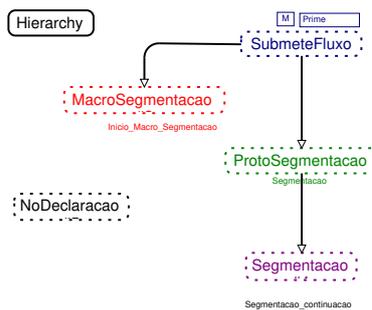


Figura 4.8: Hierarquia de páginas do SOS

A página *SubmeteFluxo* (Figura 4.9) modela o encaminhamento dos dados para o sistema. Aqui, os dados são representados pela cor *Dados* que é uma tripla  $\langle F, AM, AS \rangle$ , onde  $F$  representa o fluxo a ser segmentado,  $AM$  representa a lista dos agentes que devem participar da macro-segmentação e  $AS$  representa a lista dos agentes que estão aptos a participar na

<sup>12</sup>Para mais informações *vide* [Rei85].

<sup>13</sup>A página *NoDeclaracao* é uma página específica de modelos em Redes de Petri Coloridas. Nela, define-se cores, variáveis e eventuais funções utilizadas pelo modelo. Para tanto, utiliza-se uma extensão da linguagem ML [Ull98], chamada *CPN/ML* [Met96a].

fase de segmentação<sup>14</sup>. É necessário dizer que as atividades do Agente Interface encontram-se trivializadas neste modelo, uma vez que sua única atribuição é encaminhar os dados para o Agente Mediador.

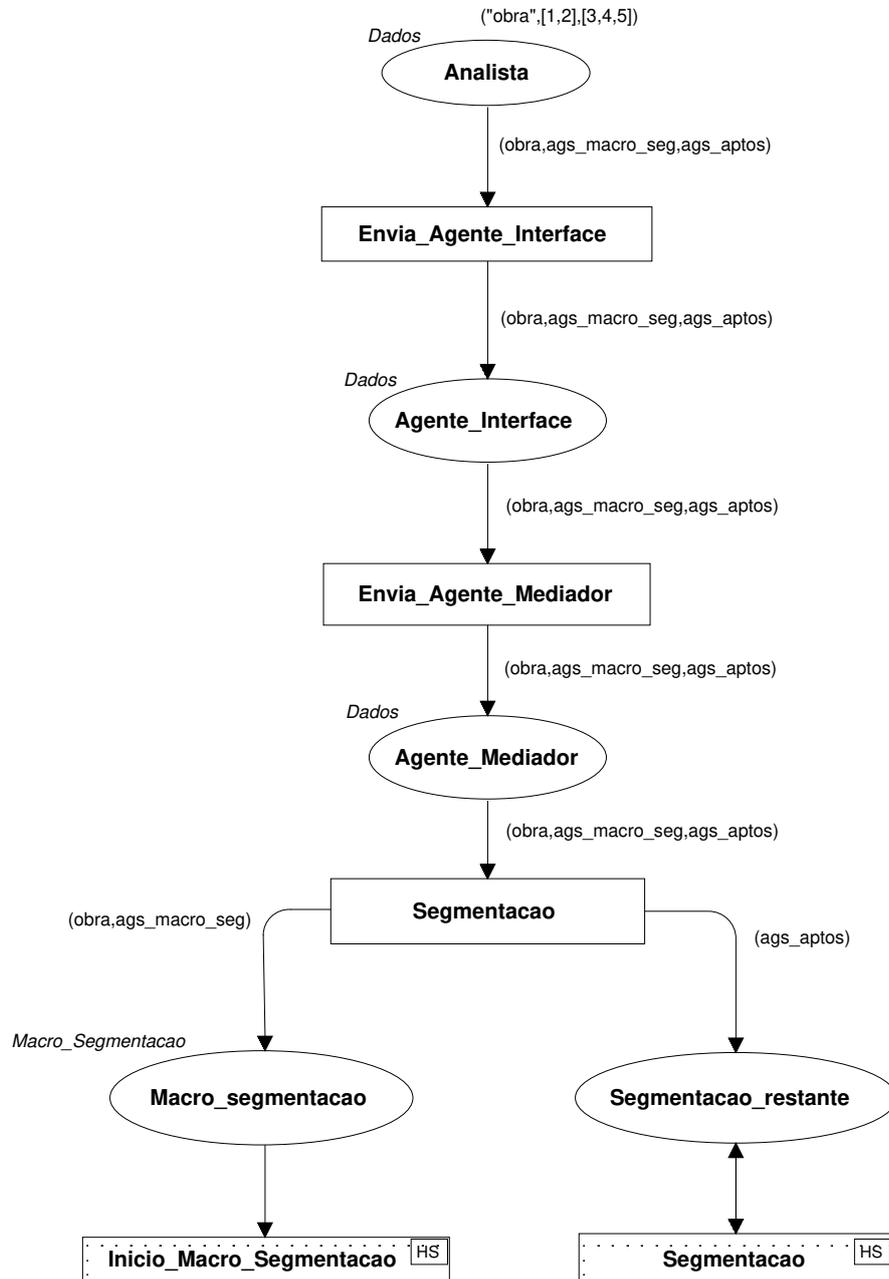


Figura 4.9: Encaminhamento de um fluxo

<sup>14</sup>Os agentes Silêncio, Pedal, Densidade de Notas, Registro e Velocidade estão representados aqui pelos inteiros 1, 2, 3, 4 e 5 respectivamente

A macro-segmentação está modelada na página *MacroSegmentação* (Figura 4.10). Esta página está ligada à página *SubmeteFluxo* através da transição *Início-Macro-Segmentação*, na página *SubmeteFluxo*, e do lugar *Início-macro-segmentação*. Aqui, guardas<sup>15</sup> nas transições *segmentação-todos*, *segmentação-pedal-ou-silêncio* e *sem-macro-segmentação* modelam o procedimento de segmentação de acordo com os agentes especificados pelo analista. A função `numeroMacroSegmentos()` gera um número aleatório, simulando, assim, a macro-segmentação realizada por estes agentes. No caso específico do analista não achar necessária a realização da macro-segmentação (indicando isto com uma lista vazia de agentes), o resultado é apenas 1 macro-segmento, isto é, a própria obra fornecida pelo analista.

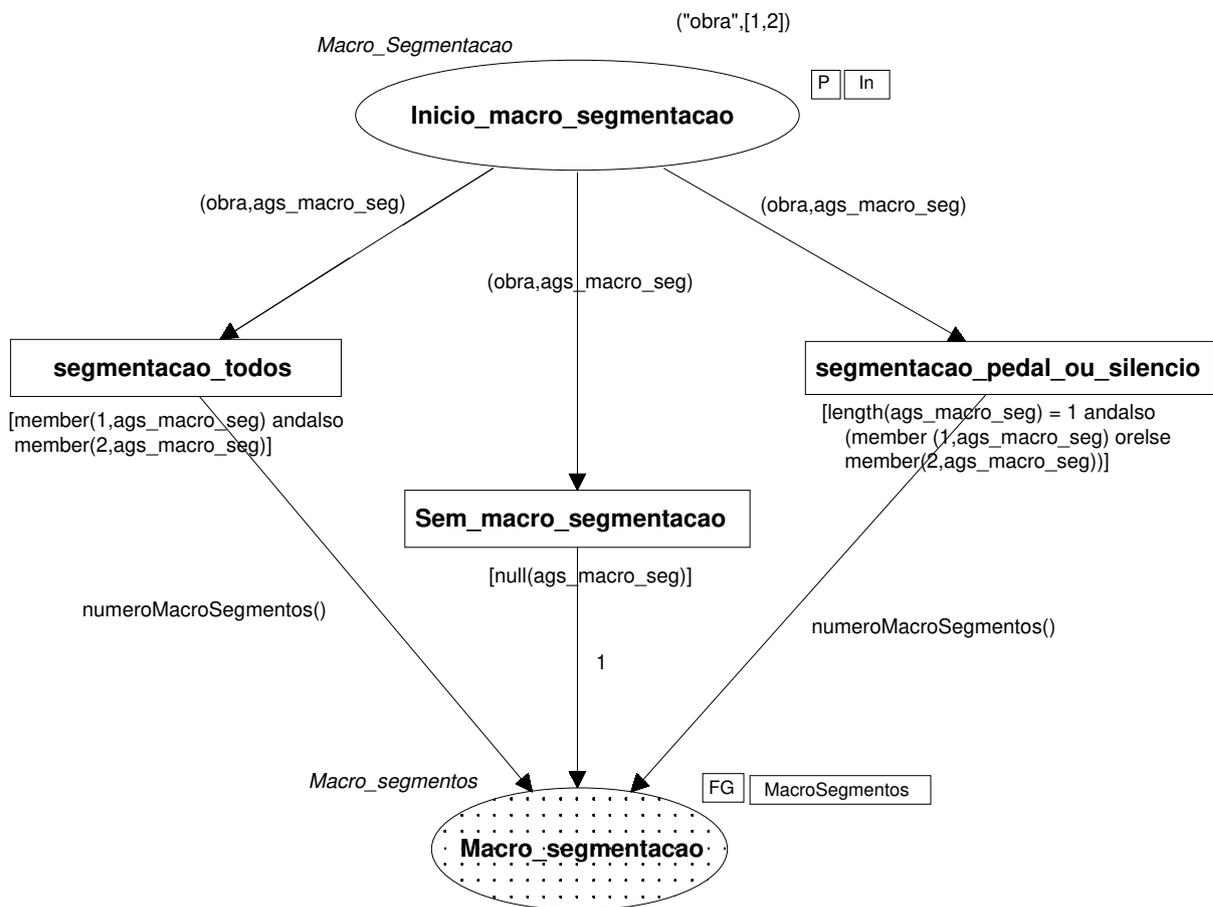


Figura 4.10: Modelo da macro-segmentação

Na página *ProtoSegmentação* (Figura 4.11) simula-se a segunda etapa do processo de segmentação, a proto-segmentação (transição *Proto-Segmentação*). Ainda nesta página do

<sup>15</sup>Nas Redes de Petri Coloridas é possível acoplar expressões booleanas—guardas—às transições que se avaliadas verdadeiras habilitam a transição, caso contrário fazem com que a transição permaneça desabilitada

modelo, cada macro-segmento, já proto-segmentado, é associado a uma lista que representa os agentes escolhidos pelo analista e que, assim, estão aptos a segmentar estes macro-segmentos. Isto é simulado na transição *Associa-macroSegmentos-agentes*.

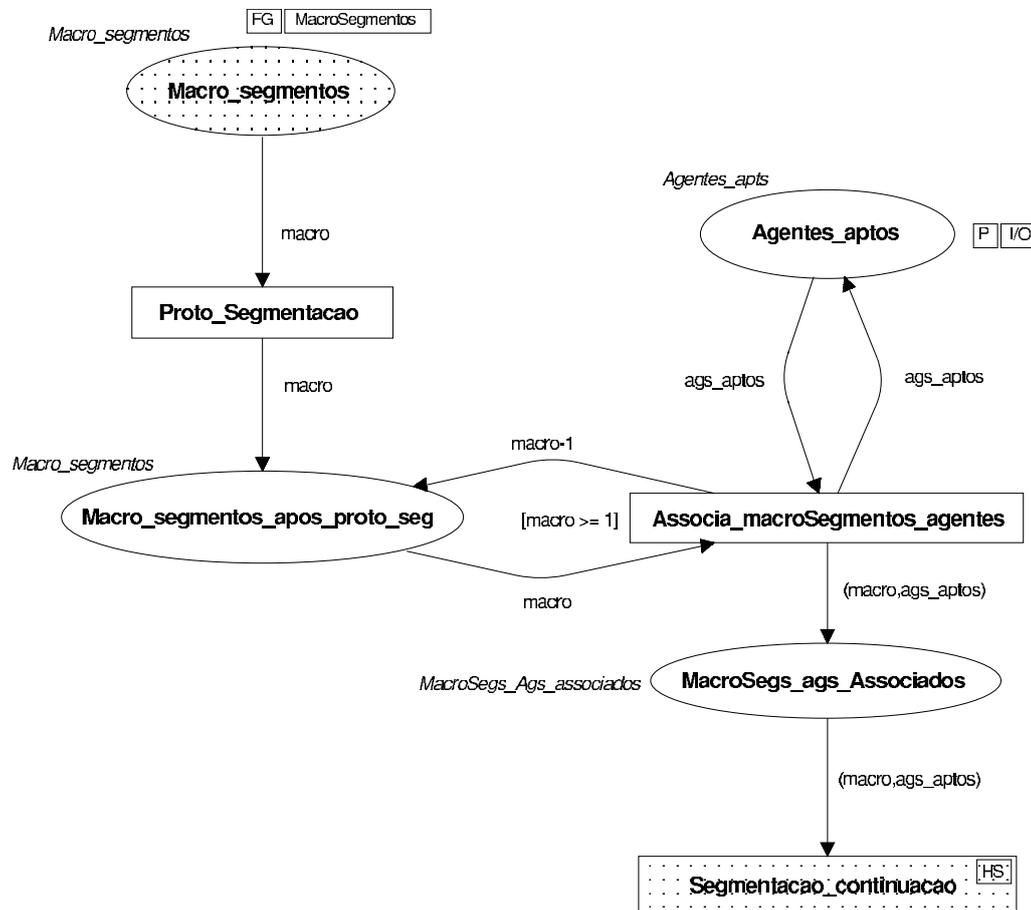


Figura 4.11: Modelo da proto-segmentação

Na última página do modelo, *Segmentação* (Figura 4.12), simula-se a terceira fase do processo de segmentação. Nela, cada agente fornece sua versão da segmentação ao agente mediador, que, então, decide qual versão é a correta. Este processo de decisão encontra-se modelado através da transição *Decisão*. Nela, gera-se um número aleatório, que representa o número de segmentos obtidos, e retira-se da lista de agentes aptos o agente que realizou esta segmentação. Este procedimento se repete até que a lista de agentes aptos fique vazia, quando então, o resultado é devolvido ao analista.

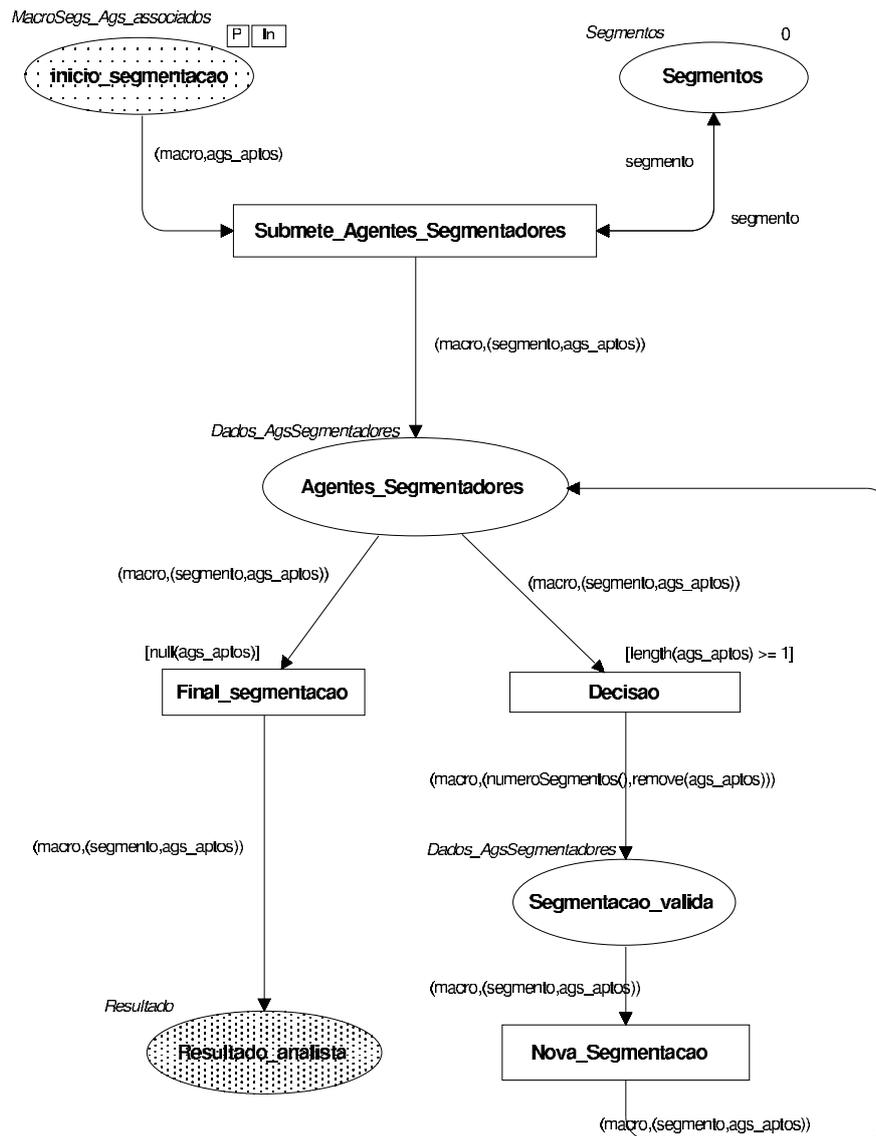


Figura 4.12: Modelo da fase de segmentação

## 4.5 Agentes Segmentadores: Arquitetura Interna

Como visto na Seção 4.3, é a SASA a responsável pela segmentação de uma obra submetida pelo analista. Ela é formada por seis agentes: o Agente Mediador (AM), o Agente Silêncio (AS), o Agente Pedal (AP), o Agente Densidade de Notas (ADN), o Agente Registro (AR) e o Agente Velocidade (AV). Nesta Seção, descreve-se como estes agentes são construídos e suas características internas.

O primeiro destes agentes, o Agente Mediador, é o agente mais complexo da SASA. De maneira semelhante aos agentes do MATHEMA, sua arquitetura interna é composta de

três camadas, a saber: *Sistema de Análise*, *Sistema Social* e *Sistema de Distribuição* (Figura 4.13).

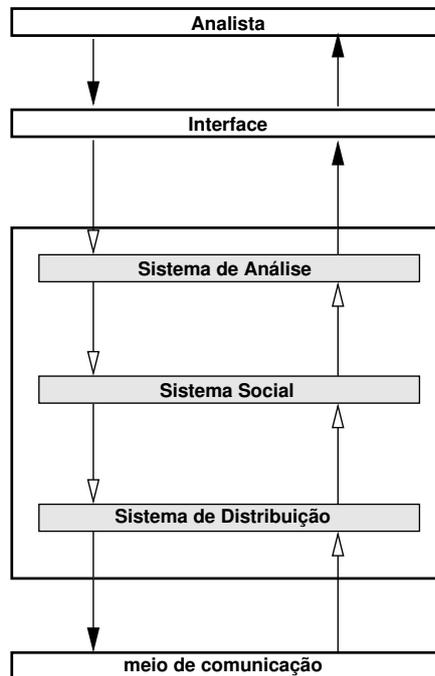


Figura 4.13: Arquitetura do agente mediador

A camada superior, o *Sistema de Análise*, é responsável por todas as decisões analíticas inteligentes realizadas pelo sistema. Ela possui um módulo resolvedor de conflitos, que é o módulo que decide que segmentação deve ser considerada em um determinado momento. Este processo de decisão acontece da seguinte maneira: ao receber as versões da segmentação de cada um dos agentes, o Agente Mediador verifica qual versão possui o maior número de *grandes rupturas*. Uma ruptura é considerada grande quando resulta de um salto de um ou mais grupos de referência na tabela de parâmetros<sup>16</sup>. Por exemplo, se, em um certo parâmetro, um segmento  $S_i$  está associado ao grupo 2 e o segmento consecutivo  $S_j$  está associado ao grupo 4, então, neste caso, existe uma grande ruptura pois acontece um salto de um grupo (o grupo 3, aqui). A versão que possuir o maior número deste tipo de ruptura é eleita como a versão da segmentação correta.

Além do módulo de resolução de conflitos, a camada superior do Agente Mediador pode

<sup>16</sup>Vale salientar que o processo de decisão só acontece durante a terceira fase do processo de segmentação, isto é, quando os parâmetros densidade de notas, velocidade e registro são analisados. Assim os grupos de referência são os que se encontram nas Tabelas 4.1, 4.2 e 4.3.

possuir outros módulos que sejam relevância para a análise de uma obra. Um destes módulos, e o único por hora, é o que incorpora o Algoritmo 3, identificando, portanto, os locais onde ocorrem proto-segmentações.

A camada intermediária, o *Sistema Social*, é responsável pelo comportamento cooperativo da sociedade. Para tanto, ela é formada pelos seguintes módulos:

- *Conhecimento Social*: neste módulo encontra-se descrito explicitamente o conhecimento dos outros agentes integrantes da SASA. Isto possibilita que o Agente Mediador saiba o que cada agente na sociedade é capaz de realizar—competências de cada agente. Ele sabe, por exemplo, que o Agente Registro sabe segmentar a partir do parâmetro registro, que o Agente Pedal é capaz de segmentar a partir do parâmetro pedal e assim por diante.
- *Conhecimento Heurístico*: neste módulo estão descritas quaisquer heurísticas que o Agente Mediador conheça. Neste caso específico, trata-se da heurística descrita na Seção 2.3.1, a de que “os parâmetros *silêncio* e *pedal* devem ser aplicados em primeiro lugar e nesta ordem, pois não apresentam prejuízos para o bom andamento da segmentação” (vide p. 22). Em geral, o conhecimento descrito neste módulo é o conhecimento necessário para a realização da macro-segmentação.
- *Controle da Segmentação*: este módulo permite que o processo de segmentação seja controlado. É através dele que o Agente Mediador sabe que agentes estão aptos a segmentar um determinado trecho da obra, em um certo momento do processo. Isto é feito mantendo informações sobre os parâmetros que já foram utilizados na segmentação de um trecho e sobre o próprio trecho, como por exemplo, se o trecho ainda possui proto-segmentos que permitam a aplicação de outros parâmetros. Desta forma, o Agente Mediador pode saber que agentes podem ou não segmentar o trecho, ou se o trecho pode ou não continuar a ser segmentado.

A camada inferior, o *Sistema de Distribuição*, é responsável pela comunicação deste agente com os outros agentes da sociedade e com o analista, tratando tanto as mensagens enviadas, quanto as recebidas pelo Agente Mediador. Pelo menos dois tipos de mensagens são tratadas por esta camada: mensagens enviadas e recebidas dentro da sociedade, como,

por exemplo, ordens para que os outros agentes da sociedade segmentem uma determinada obra; e mensagens trocadas com o meio externo, neste caso com o analista (*via* Agente Interface).

A Figura 4.14 ilustra este detalhamento da arquitetura interna do Agente Mediador.

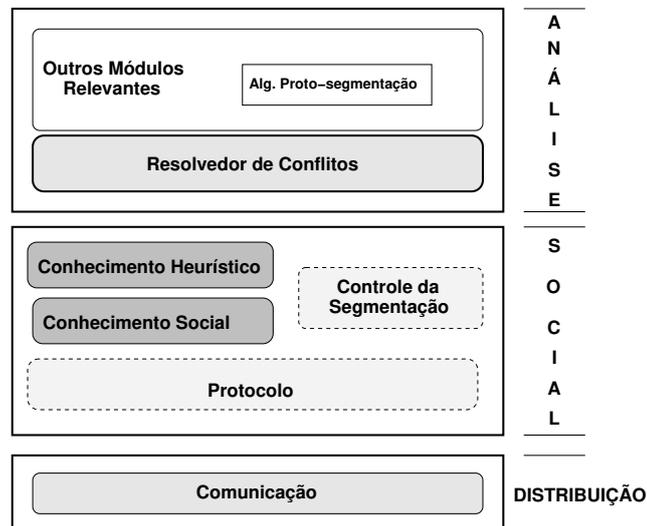


Figura 4.14: Arquitetura detalhada do Agente Mediador

Os outros agentes são agentes bem mais simples que o Agente Mediador. Eles são, na verdade, agentes reativos, cada um responsável por encontrar rupturas de continuidade em um parâmetro específico e a partir do trecho fornecido a eles pelo Agente Mediador. Os agentes deste tipo são os seguintes:

1. *Agente Silêncio (AS)*: agente responsável pela execução do algoritmo que identifica rupturas de continuidade no parâmetro silêncio (Algoritmo 1).
2. *Agente Pedal (AP)*: agente responsável pela execução do algoritmo que identifica rupturas de continuidade no parâmetro pedal (Algoritmo 2).
3. *Agente Densidade de Notas (ADN)*: agente responsável pela execução do algoritmo que identifica rupturas de continuidade no parâmetro densidade de notas (Algoritmo 4).
4. *Agente Registro (AR)*: agente responsável pela execução do algoritmo que identifica rupturas de continuidade no parâmetro registro (Algoritmo 6).

5. *Agente Velocidade (AV)*: agente responsável pela execução do algoritmo que identifica rupturas de continuidade no parâmetro intensidade, ou velocidade no jargão MIDI (Algoritmo 5).

Estes agentes preservam a mesma arquitetura interna—em três camadas—do Agente Mediador. Estas camadas, no entanto, foram simplificadas, pois estes agentes realizam tarefas já previamente definidas. A camada superior é encarregada unicamente da execução do algoritmo de segmentação de cada agente. A camada intermediária foi “esvaziada” pois não é necessária para a realização da tarefa destes agentes, uma vez que eles são agentes reativos e não necessitam de conhecimento sobre a sociedade para a realização de suas tarefas. A camada inferior possui as mesmas características da camada inferior do Agente Mediador, isto é, o tratamento das mensagens recebidas e enviadas pelo agente. O tipo de mensagens, no entanto, muda pois estes agentes não possuem qualquer contato com o meio externo. A troca de mensagens é feita exclusivamente com o Agente Mediador. A arquitetura destes agentes encontra-se ilustrada na Figura 4.15.

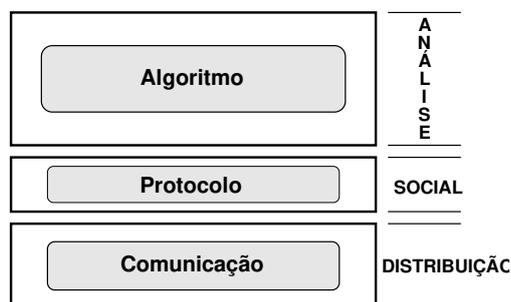


Figura 4.15: Arquitetura dos agentes reativos

# Capítulo 5

## Implementação e Testes

Neste capítulo, apresenta-se, em linhas gerais, as principais partes que integram o protótipo, implementado a partir do exposto no Capítulo 4. Nesta apresentação, descreve-se as principais classes concebidas para implementá-lo. Em seguida, apresenta-se alguns testes realizados com o protótipo.

### 5.1 Aspectos de Implementação

Para a implementação do protótipo procurou-se por uma linguagem que fosse multi-plataforma e que, na medida do possível, fornecesse ao programador recursos suficientes para a manipulação dos dados MIDI, utilizados como forma de representação da música. A linguagem escolhida foi *JAVA*, uma vez que atende às duas necessidades iniciais satisfatoriamente. Ademais, diversas partes do *MATHEMA* já se encontravam implementadas nesta linguagem, o que a tornou uma candidata natural para a implementação do *SOS*. Na descrição da implementação que segue, utiliza-se diagramas que ilustram as principais partes do protótipo. Estes diagramas—*diagramas de classes*—são construídos de acordo com a notação UML [BRJ99].

Na Figura 5.1, encontra-se uma visão geral do esquema de classes que compõem a implementação. Nela constam cinco pacotes, cada um agregando conjuntos de classes que implementam diferentes aspectos do sistema.

O primeiro destes pacotes, *Agente* (Figura 5.2), implementa a infra-estrutura básica dos agentes. Esta infra-estrutura básica serve tanto para o *Agente Mediador*, quanto

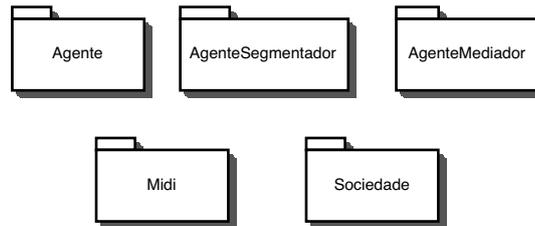
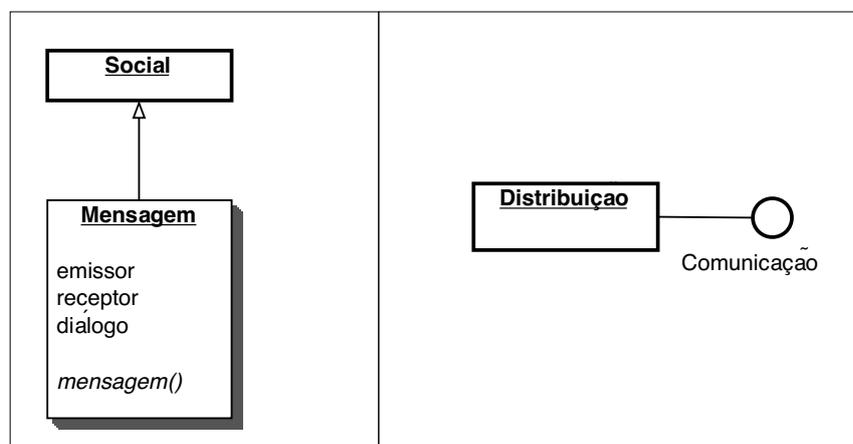


Figura 5.1: Visão geral do SOS

para os Agentes Segmentadores. As classes que compõem este pacote são: `Social` e `Distribuição`. A classe `Social` fornece as funções básicas da camada social dos agentes. Ela possui uma subclasse, `Mensagem`, que implementa as funções básicas de criação de mensagens que devem ser trocadas no sistema. Isto é feito por um método—`mensagem()`—que cria uma mensagem com três campos: *emissor*, que identifica o agente que está emitindo a mensagem; *receptor*, que identifica a que agente a mensagem se destina; e *conteúdo*, que corresponde ao conteúdo da mensagem em questão. A classe `Distribuição` provê a interface para a camada inferior de mesmo nome dos agentes, explicada no Capítulo 4. Através desta interface, o agente envia e recebe as mensagens trocadas no sistema.

Figura 5.2: O pacote *Agente*

O segundo pacote, `AgenteMediador` (Figura 5.3), agrega as classes que implementam o Agente Mediador do sistema. A classe `Análise` corresponde à camada de mesmo nome do Agente Mediador, explicada no capítulo anterior. Ela possui dois métodos: `contadorRupturas()`, que equivale ao módulo *Resolvedor de Conflitos* do Agente

Mediador, sendo, portanto, responsável pela decisão de que versão da segmentação considerar; e `protoSegmentação()`, que é, na verdade, a implementação do algoritmo de proto-segmentação (Algoritmo 3, na página 42). A classe `SocialMediador`, que especializa a classe `Social` do pacote `Agente`, possui três subclasses. A primeira subclasse, `ConhecimentoSocial`, implementa o módulo *Conhecimento Social*, descrito no capítulo anterior. Seu principal método—`habilidade()`—determina a habilidade de cada um dos agentes do sistema. A segunda subclasse, `ControleSegmentação`, implementa as funções de controle da segmentação, ou seja, o controle do estágio de segmentação em que se encontra cada segmento (método `estágio()`), e a alocação de agentes para cada segmento (método `assocAgentesSegmentos()`), que é feito de acordo com o resultado do método anterior. Por fim, a terceira subclasse, `AcionarAnalista`, implementa um método que aciona o analista no caso do Agente Mediador não conseguir decidir que versão da segmentação considerar. A classe `DistribuiçãoMediador` possui um método—`sincronizar()`—que recebe as mensagens criadas na camada *Social*, repassando-as para os agentes a que elas se destinam.

O pacote `AgenteSegmentador` (Figura 5.4) agrupa três classes: `AnáliseSegmentador`, `SocialSegmentador` e `DistribuiçãoSegmentador`. A primeira classe, `AnáliseSegmentador`, possui cinco subclasses, cada uma implementando a camada *Análise* de um Agente Segmentador diferente. Cada subclasse possui um método—`segmente()`—que implementa os algoritmos de segmentação para os parâmetros silêncio, pedal, densidade de notas, registro e velocidade (*vide* Seção 4.2). A classe `SocialSegmentador`, que depende da classe `Social` do pacote `Agente`, serve para a criação de mensagens contendo a segmentação realizada por cada um dos Agentes Segmentadores. Já a classe `DistribuiçãoSegmentador`, que depende da classe `Distribuição` do pacote `Agente`, envia e recebe mensagens para os agentes do sistema.

O quarto pacote, `Midi`, implementa as classes responsáveis pela manipulação dos arquivos MIDI utilizados na segmentação. Estas classes fazem uso da API—*Application Programming Interface*—`javax.sound.midi` e são utilizadas intensivamente pelos Agentes Segmentadores e pelo Agente Mediador durante o processo de segmentação. Este pacote, todavia, não será pormenorizado.

O último pacote, `Sociedade`, implementa a interface de janelas do SOS. Ele utiliza a

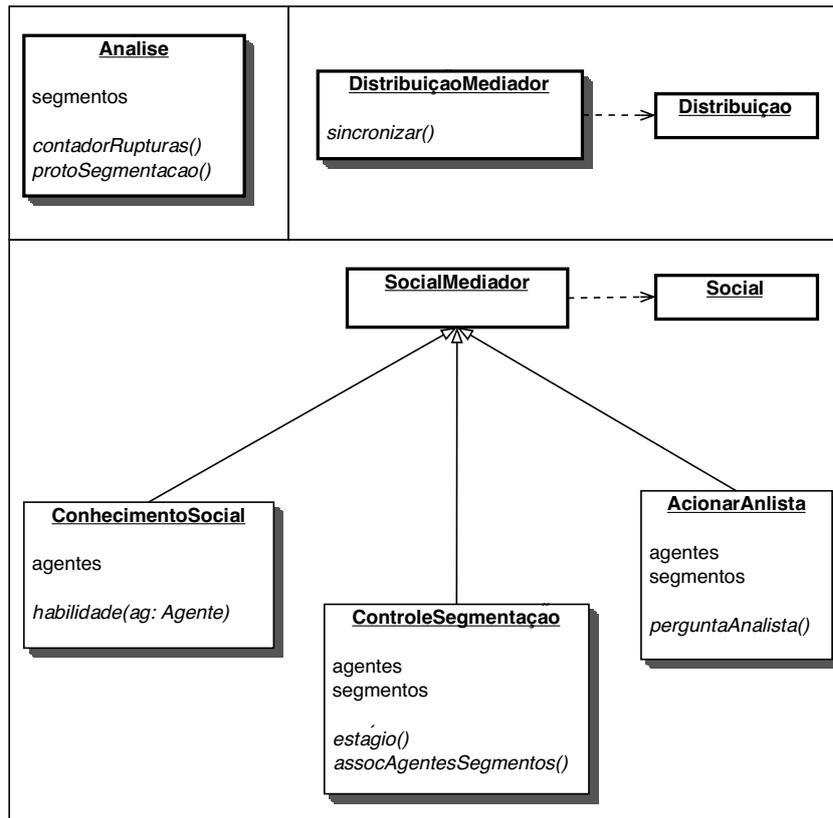


Figura 5.3: O pacote *AgenteMediador*

API *javax.swing* e, como o pacote anterior, não será descrito em maiores detalhes.

## 5.2 Testes

Os testes com o protótipo foram realizados utilizando a obra *Twentyfive Pages for 1 to 25 Pianos* do compositor Earle Brown [Bro75]. Composta em 1953, esta é uma obra particular do repertório pianístico do século XX pois é uma das primeiras obras construídas na chamada Forma Aberta [Eco97]. De uma maneira geral, as composições baseadas neste tipo de estruturação deixam diversos aspectos de sua “montagem” a cargo do intérprete. Nas palavras de Eco: pode-se notar uma “peculiar autonomia executiva concedida ao intérprete, o qual não só dispõe da liberdade de interpretar as indicações do compositor conforma sua sensibilidade pessoal (como se dá no caso da música tradicional), mas também deve intervir na forma da composição, não raro estabelecendo a duração das notas ou a sucessão dos sons num ato de improvisação criadora” [Eco97, p. 37]. Neste caso específico, o pianista é encarregado de

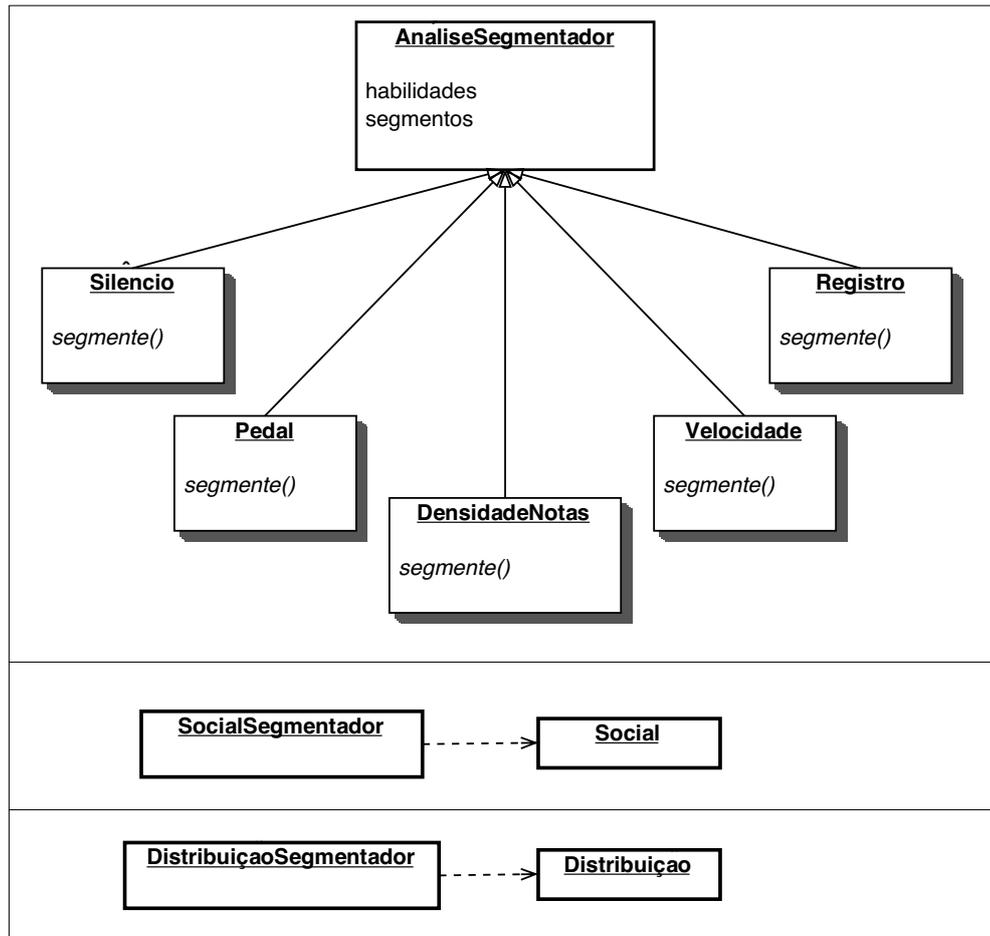


Figura 5.4: O pacote *AgenteSegmentador*

decidir a ordem na execução das páginas e se cada página deve ser executada de cabeça para baixo ou não. Além disto, a obra pode ser interpretada com 1 até 25 pianos<sup>1</sup>. A escolha desta obra justifica-se pelos seguintes motivos:

- as diversas páginas possuem desde de uma estruturação simples, com poucas superposições de notas e variações de intensidade, até páginas com estrutura complexa e intrincada. Isto permite que os testes sejam realizados a partir de diferentes níveis de complexidade e com a mesma obra.
- Ela possui uma notação gráfica, bastante diferente da tradicional, permitindo que não-músicos possam entendê-la, em linhas gerais, sem muita dificuldade. Nesta notação, as notas são marcadas por traços horizontais e a duração da nota é determinada pelo

<sup>1</sup>Mais informações acerca das possibilidades de execução desta obra podem ser obtidas em [GdA01].

tamanho destes traços (Figura 5.5).

- Permite a montagem de diferentes versões utilizando o mesmo conjunto de arquivos MIDI, o que aumenta consideravelmente o número e a complexidade dos testes que podem ser realizados. Isto não aconteceria com obras que não fossem construídas na Forma Aberta.
- Por fim, esta obra faz parte do repertório pianístico ao qual a Análise Orientada a Objetos Sonoros se aplica.

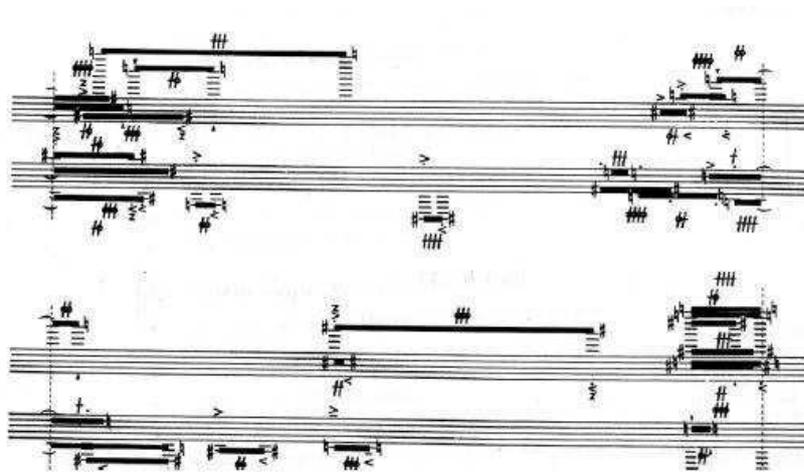


Figura 5.5: Trecho final da Página 1 (© Universal Edition Ltda, Toronto, Canadá)

Foram realizadas duas seqüências de testes: a primeira seqüência serviu para a validação individual dos algoritmos dos agentes reativos dos sistema, a saber: Agente Silêncio, Agente Pedal, Agente Densidade de Notas, Agente Registro e Agente Velocidade. A segunda seqüência serviu para o teste do processo de segmentação propriamente dito, servindo para validar a construção da sociedade e do SOS, conseqüentemente. Os resultados obtidos durante os testes foram confrontados com análises previamente realizadas por um analista humano.

Na primeira seqüência de testes, os resultados foram 100% corretos. Todos os agentes apresentaram resultados idênticos aos da análise manual previamente realizada. Para estes testes, foram utilizadas as páginas 5, 8 e 11<sup>2</sup>. Vale notar que os valores mínimos utilizados

<sup>2</sup>Nesta obra não existe nenhuma marcação de pedal. No intuito de simular uma segmentação neste parâmetro, foi introduzida uma marca deste tipo na página 11.

para a identificação de segmentações no parâmetro silêncio nestes testes foram os seguintes: um segundo para a página 5, dois segundos para página 8 e um segundo para a página 11. Os resultados encontram-se na Tabela 5.1.

Páginas	Silêncio		Pedal		Dens. Notas		Velocidade		Registro	
	Manual	Autom.	Manual	Autom.	Manual	Autom.	Manual	Autom.	Manual	Autom.
05	2	2	0	0	9	9	0	0	7	7
08	4	4	0	0	4	4	0	0	4	4
11	5	5	2	2	3	3	0	0	3	3

Tabela 5.1: Resultados da primeira seqüência de testes

Uma vez testada a segmentação em cada um dos parâmetros, realizou-se uma segunda etapa de testes. Esta serviu para testar o protótipo como um todo, isto é, se os agentes seriam realmente capazes de segmentar um determinado fluxo musical corretamente<sup>3</sup>.

A título de exemplo, descreve-se aqui um dos testes realizados com a página 1. Os agentes que participam da segmentação neste exemplo são o Agente Silêncio, o Agente Pedal, o Agente Densidade de Notas e o Agente Registro. O Agente Velocidade não participa do processo de segmentação porque a velocidade, em particular, não é um parâmetro de relevância para a análise desta obra<sup>4</sup>.

A primeira etapa no processo de segmentação é a macro-segmentação. Especificamente neste teste, ambos os agentes que integram esta etapa, o Agente Silêncio e o Agente Pedal, foram escolhidos pelo analista. Além disso, o valor a partir do qual pode existir uma segmentação no parâmetro silêncio estabelecido pelo analista é de 1000 segundos. O resultado nesta etapa foi que nenhum dos agentes foi capaz de segmentar a obra. A fase seguinte, a proto-segmentação, é feita internamente pelo Agente Mediador. Feita a proto-segmentação, o sistema passa para a etapa final do processo, que é a segmentação nos parâmetros restantes (aqui, densidade de notas e registro). Num primeiro momento, o Agente Mediador envia para o Agente Densidade de Notas e para o Agente Registro o resultado da etapa de

<sup>3</sup>Além das páginas utilizadas no teste anterior, também incluiu-se nesta seqüência de testes a página 1.

<sup>4</sup>Este fato fica mais evidente quando se observa os resultados obtidos nas tentativas de segmentações apresentados na Tabela 5.1: nota-se que em nenhuma das páginas ocorre segmentação no parâmetro velocidade. Possivelmente, o mesmo ocorreria com outras páginas, uma vez que, via de regra, cada nota possui uma velocidade diferente da outra, fazendo com o nível de rupturas neste parâmetro especificamente seja muito “granular”, podendo, assim, ser desprezado para fins de segmentação.

proto-segmentação. Em seguida, cada agente devolve para o Agente Mediador sua versão da segmentação. Este último decide que segmentação deve ser considerada. Neste teste, o resultado escolhido foi o do Agente Densidade de Notas, uma vez que é ele que apresenta o maior número de grandes rupturas. Este resultado—dez segmentos—é enviado para o Agente Registro, que é o único agente ainda apto a realizar segmentações. Este, no entanto, não pode realizar nenhuma segmentação, retornando para o Agente Mediador os mesmos dez segmentos recebidos. O resultado final do teste é consistente com a segmentação manual previamente realizada. A Figura 5.6 ilustra o protótipo ao final da segmentação da página 1. Pode-se notar, na janela de cada agente, parte dos diálogos entre os agentes segmentadores e o Agente Mediador.

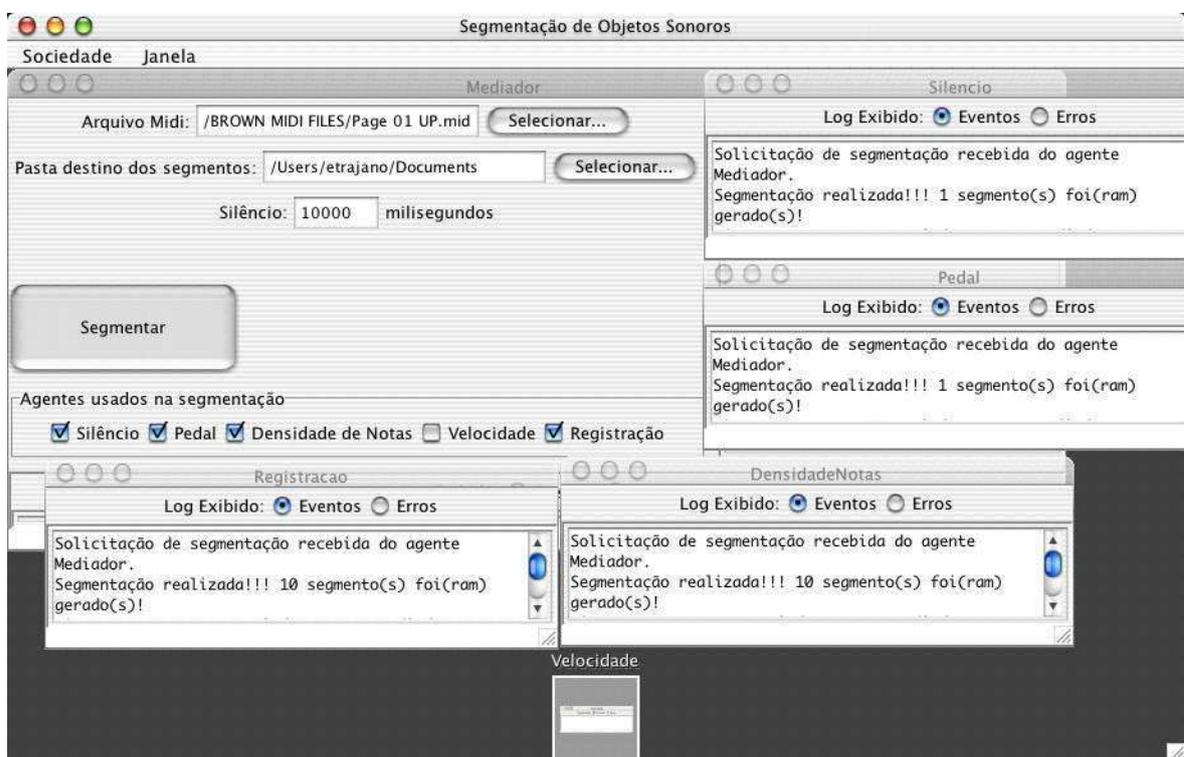


Figura 5.6: O protótipo após a segmentação da página 1

O restante dos testes realizados com esta e outras páginas da obra também revelam resultados consistentes com as análises manuais, o que evidencia a corretude do modelo de segmentação e do protótipo aqui implementado.

# Capítulo 6

## Conclusões e Trabalhos Futuros

Neste capítulo, apresenta-se conclusões sobre o presente trabalho. Além das conclusões, enumera-se pontos importantes que evidenciam que este trabalho apresenta avanços em relação às abordagens expostas no Capítulo 2. Por fim, apresenta-se trabalhos futuros, pontos que ainda necessitam de melhor estudo e aperfeiçoamento para que o SOS se constitua em um real sistema de apoio às atividades de análise musical.

### 6.1 Conclusões e Contribuições do Trabalho

O trabalho apresentado nesta dissertação teve como objetivo geral a concepção de um sistema para segmentação automática de fluxos musicais. Neste sentido, desenvolveu-se inicialmente um estudo sobre a complexidade existente na automação da segmentação. Analisando e discutindo algumas abordagens utilizadas na construção de sistemas de segmentação, obteve-se uma espécie de conjunto dos requisitos necessários para a construção de sistema de segmentação automática.

Após esta análise do problema, procurou-se por uma metodologia de análise que fosse suficientemente simples para uma tentativa inicial de automação, mas que preservasse dificuldades características deste tipo de problemas, como por exemplo, a multiplicidade de parâmetros de segmentação. A metodologia de análise escolhida foi a Análise Orientada a Objetos Sonoros <sup>1</sup>.

De posse do conjunto de requisitos e escolhida a metodologia de análise a ser utilizada,

---

<sup>1</sup>Na verdade, uma versão simplificada desta metodologia.

procurou-se por um modelo de construção de sistemas que permitisse que este conjunto de requisitos fosse satisfeito. Além disso, este modelo deveria contemplar o processo de segmentação prescrito pela metodologia de análise escolhida. O modelo adotado foi um Sistema Multi-Agentes, onde cada agente do sistema corresponde a um parâmetro de segmentação. Este modelo foi, em seguida, implementado e testado.

No que diz respeito ao conjunto de requisitos levantados (Capítulo 2), a solução desenvolvida atende aos seguintes pontos:

1. ela permite que diversos componentes da estrutura musical sejam considerados durante o processo de segmentação. Especificamente, ela permite que cinco componentes da Análise Orientada a Objetos Sonoros sejam examinados durante o processo de segmentação: silêncios, pedal, densidade de notas, registro e velocidade.
2. O analista participa do processo de segmentação, seja fornecendo informações pertinentes ao processo, seja decidindo, em casos extremos, que versão de segmentação deve ser considerada.
3. O protótipo é capaz de segmentar a partir de um nível da estrutura musical superior à nota, o que leva a resultados mais expressivos que as abordagens anteriores, incluindo aí a segmentação de música polifônica.

Um último ponto, a necessidade de incorporação de diversas abordagens analíticas no sistema, não pôde ser contemplado durante o trabalho. A solução proposta, no entanto, fornece mecanismos para que, em versões futuras, novas metodologias de análise possam ser incorporadas ao sistema. Pode-se construir o sistema de forma que determinados grupos de agentes, ou mesmo sociedades inteiras, sejam responsáveis pela segmentação de uma obra a partir de uma metodologia de análise. Desta maneira, torna-se possível que diversas metodologias coexistam em um mesmo ambiente de análise.

Do exposto, pode-se dizer que o conjunto de requisitos levantado após a análise das abordagens para a automação da segmentação utilizadas até então puderam ser validados, ainda que parcialmente. Além disso, os resultados obtidos com os testes do protótipo fornecem fortes evidências de que Sistemas Multi-Agentes se constituem em uma abstração adequada para a construção de sistemas de segmentação automática de fluxos musicais.

No que tange as contribuições, o conjunto de requisitos e a proposta do uso de Sistemas Multi-Agentes para a construção de sistemas de segmentação automática de fluxos musicais, se constituem nas principais contribuições deste trabalho. Outra contribuição é o protótipo implementado, que poderá servir de base para a construção de um real sistema de suporte às atividades de análise musical assistida por computador.

## 6.2 Trabalhos futuros

Apesar destas contribuições, muito ainda deve ser feito para que a segmentação seja completamente automatizada. A primeira coisa a ser feita com o protótipo implementado é testá-lo intensivamente. Os testes descritos na Seção 5.2 mostraram resultados excelentes. Entretanto, a obra utilizada nos testes, apesar de, por exemplo, ser polifônica, possui uma estruturação bastante simples. Obras mais complexas merecerão mais atenção o que, provavelmente, trará problemas não esperados.

Pode-se dizer, ainda, que alguns algoritmos são aproximações um tanto grosseiras da realidade da música. O algoritmo que encontra rupturas no parâmetro *pedal* é um exemplo. Na versão aqui apresentada, assume-se que um pedal nunca ocorrerá enquanto uma nota esteja sendo executada. Isto é, aqui pedais só são ativados e desativados antes, concomitantemente ou após o final de uma nota, nunca durante. Isto é uma distorção do mundo real e precisa ser tratada em versões futuras do protótipo.

Ainda no âmbito da Análise Orientada a Objetos Sonoros, algumas tarefas devem ser realizadas. A primeira delas é a integração do sistema de segmentação com as ferramentas de análise dos segmentos resultantes. Como dito anteriormente, encontra-se em desenvolvimento uma biblioteca de funções para a Análise Orientada a Objetos Sonoros, denominada SOAL (*Sonic Object Analysis Library*) e, por hora, o usuário deve utilizar os arquivos MIDI resultantes da segmentação nesta biblioteca de análise. No entanto, um sistema único onde esta integração aconteça de forma transparente é mais recomendável. Um outro ponto importante é a extensão da representação da música para outros tipos de arquivos. Isto porque a representação MIDI é bastante limitada, não captando, por exemplo, nenhuma informação acerca do timbre do instrumento ou instrumentos para os quais a obra foi composta. O arquivo áudio pode se constituir em uma alternativa viável, fornecendo ao sistema informa-

ções complementares, não captadas pelo formato MIDI, mas de extrema importância para a Análise Orientada a Objetos Sonoros, como o timbre. Além disso, a extensão para outras representações pode se constituir em um primeiro passo para a segmentação de música composta para outros instrumentos e não apenas de música para piano.

No que diz respeito ao sistema e seus agentes, alguns pontos podem se constituir em avanços. O primeiro deles é a mudança da política de coordenação de ações dos agentes do protótipo. É necessário reformular a forma como a coordenação é realizada. Apesar de ter se mostrado suficiente para os propósitos deste trabalho, da forma como se encontra, a coordenação não é capaz de lidar com situações mais complexas e mais próximas da realidade do mundo da Análise Musical. Além disso, a instauração de protocolos de negociação e de resolução de conflitos mais refinados e elaborados também se constituem em pontos de trabalho de relevância para o melhoramento do sistema.

Outro ponto de trabalho são os próprios agentes que integram o sistema. No atual estágio, estes agentes possuem conhecimentos sobre o domínio e habilidades bastante limitadas. Assim, para se obter um sistema mais capaz de obter resultados interessantes para o analista, é necessário que os agentes sejam mais “competentes” em seus subdomínios específicos. Isto possivelmente implicará, em algum momento, no uso de alguma forma aprendizagem por parte dos agentes. Além disso, a própria mudança na coordenação e instauração de protocolos de negociação e de resolução de conflitos implicará em grandes mudanças nos agentes.

Um terceiro ponto, e um ponto crucial para um sistema de segmentação automática, é a integração de outras abordagens de Análise Musical à abordagem já existente. É fato que determinadas metodologias de análise são aplicáveis apenas a determinados repertórios, a determinados conjuntos de obras, não se adequando a todo e qualquer repertório<sup>2</sup>. É necessário, então, que o sistema seja capaz de segmentar a partir de diversas metodologias. Imagina-se que isto pode ser feito a partir da construção de sociedades, cada uma representando uma metodologia de análise. Este tipo de integração entre as abordagens pode levar a problemas como classificação, negociação e resolução de conflitos, uma vez que o sistema deverá realizar uma análise prévia da obra para decidir a qual sociedade esta deverá ser encaminhada para que a segmentação seja realizada.

---

<sup>2</sup>Pelo menos não sem modificações significativas na metodologia.

# Bibliografia

- [AC87] P. Agre e D. Chapman. PENGI: An implementation of a theory of activity. In *Proc. of the Sixth National Conference on Artificial Intelligence (AAAI-87)*, páginas 268–272, Seattle (WA), 1987.
- [Ago98] Carlos Agon. *OpenMusic: Un langage visuel pour la composition musicale assistée par ordinateur*. PhD thesis, IRCAM–Paris VI, Paris, 1998.
- [ARL<sup>+</sup>99] Gérard Assayag, Camilo Rueda, Mikael Laurson, Carlos Agon, e Olivier Dele-rue. Computer-assisted composition at IRCAM: From PatchWork to OpenMusic. *Computer Music Journal*, 23(3), 1999.
- [B<sup>+</sup>92] Mira Balaban et al., editores. *Understanding Music with AI: Perspectives on Music Cognition*. AAAI Press e MIT Press, Cambridge(MA), 1992.
- [BC00] Rosfran Lins Borges e Evandro de Barros Costa. Agentes para o gerenciamento de redes locais e remotas. In Flávio Morais de Assis Silva, editor, *Anais da VIII Semana de Informática da UFBA—Seminfo2000*, páginas 1–14, 2000.
- [Ben80] Ian Bent. Analysis. In *New Grove’s Dictionary of Music*, páginas 340–388. Mac-Millan, Londres, 1980.
- [BG88] Alan H. Bond e Les Gasser. *Readings in Distributed Artificial Intelligence*. Morgan Kaufman, San Mateo, 1988.
- [BIP88] M. E. Bratman, D. J. Israel, e M. E. Pollack. Plans and resource-bounded practical reasoning. *Computational Intelligence*, 4:349–355, 1988.
- [Bit96] Guilherme Bittencourt. *Inteligência Artificial: Ferramentas e Teorias*. Instituto de Computação - UNICAMP, Campinas, 1996.

- [BRJ99] Grady Booch, James Rumbaugh, e Ivar Jacobson. *The Unified Modeling Language User Guide*. Addison-Wesley, Reading (MA), 1999.
- [Bro75] Earle Brown. *Twentyfive Pages for 1 to 25 Pianos*. Universal Edition, Toronto, 1975.
- [Bro86] Rodney A. Brooks. A robust layered control system for mobile robot. *IEEE Journal of Robotics and Automation*, 2(1):14–23, 1986.
- [Bro90] Rodney A. Brooks. Elephants don't play chess. *Journal of Robotics and Autonomous Systems*, 6:3–15, 1990.
- [BSV98] Jacopo Baboni-Schilingi e Frédéric Voisin. *Morphologie: Fonctions d'Analyse, de Reconnaissance, de Classification et de Reconstitution de Séquences Symboliques et Numeriques*. IRCAM, Paris, second edição, 1998.
- [Cam98] Emiliós Cambouropoulos. *Towards a General Computational Theory of Musical Structure*. PhD thesis, The University of Edinburgh, 1998.
- [Car95] Otávio A. S. Carpinteiro. A neural model to segment musical pieces. In Eduardo Reck Miranda, editor, *Proc. of the Second Brazilian Symposium on Computer Music, Fifteenth Congress of the Brazilian Computer Society*, páginas 114–120, 1995.
- [CBF97] Evandro de Barros Costa, G. Bittencourt, e E. Flemming. Um sistema tutor multi-agentes em geometria. In *Anais do VIII Simpósio Brasileiro de Informática na Educação—SBIE97*, volume 2, São José dos Campos, Nov. 1997.
- [CIR98] Tim Crawford, Costas S. Iliopoulos, e Rajeev Raman. String matching techniques for musical similarity and melodic recognition. *Computing in Musicology*, 11:73–100, 1998.
- [Coo94] Nicholas Cook. *A Guide to Musical Analysis*. Oxford University Press, Oxford, 1994.
- [Cop99] David Cope. One approach to musical intelligence. *IEEE Intelligent Systems*, Mai–Jun 1999.

- [Cos97] Evandro de Barros Costa. *Um Modelo de Ambiente Interativo de Aprendizagem Baseado numa Arquitetura Multi-Agente*. PhD thesis, UFPB/CCT/COPELE, 1997.
- [CPF96] Evandro de Barros Costa, Angelo Perkusich, e Jorge César Abrantes Figueiredo. A multi-agent based environment to aid in the design of petri nets based software systems. In *Proc. of the 8th International Conference on Software Engineering and Knowledge Engineering—SEKE96*, Illinois, Jun. 1996.
- [CTF97] Evandro de Barros Costa, Luciênio de Macedo Teixeira, e Edilson Fereda. Um sistema tutor inteligente multi-agentes em harmonia musical. In *Anais do VIII Simpósio Brasileiro de Informática na Educação—SBIE97*, volume 1, São José dos Campos, Nov. 1997.
- [CTFC00] Adriano de Oliveira Caminha, Luciênio de Macêdo Teixeira, Edilson Fereda, e Evandro de Barros Costa. MHTIS—a musical harmony intelligent tutoring system. In *Anais do VII Simpósio Brasileiro de Computação Musical, XX Congresso Nacional da Sociedade Brasileira de Computação—SBC*, volume 1. Champagnat, 2000.
- [Dan00] Roger B. Dannenberg. Artificial intelligence, machine learning and music understanding. In *Anais do VII Simpósio Brasileiro de Computação Musical, XX Congresso Nacional da Sociedade Brasileira de Computação—SBC*, volume 1. Champagnat, 2000.
- [Dix00] Simon Dixon. A lightweight multi-agent musical beat tracking system. In *Pacific Rim International Conference on Artificial Intelligence*, páginas 778–788, 2000.
- [dSM01] Flávio Soares Corrêa da Silva e Eudênia Xavier Meneses. Integração de agentes de informação. In *Anais do XXI Congresso da Sociedade Brasileira de Computação, I Jornada de Atualização em Inteligência Artificial*, volume 3, páginas 209–253, 2001.
- [Eco97] Umberto Eco. *Obra Aberta*. Editora Perspectiva, oitava edição, 1997.

- [Fer92] I. A. Ferguson. *TouringMachines: An Architecture for Dynamic, Rational, Mobile Agents*. PhD thesis, University of Cambridge—UK, Clare Hall, 1992.
- [Fer95] Jacques Ferber. *Les Systèmes Multi-Agent: Vers une Intelligence Collective*. InterEditions, Paris, 1995.
- [Fer99] Jacques Ferber. *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*. Addison-Wesley, Reading (MA), 1999. Tradução inglesa da edição em francês.
- [FFMM94] T. Finin, R. Fritzson, D. McKay, e R. McEntire. KQML as an agent communication language. In *Proceedings of the 3rd International Conference on Information and Knowledge Management (CIKM'94)*, páginas 456–463, Gaithersburg, Maryland, 1994. ACM Press.
- [FG96] Stan Franklin e Art Graesser. Is it an agent, or just a program? a taxonomy for autonomous agents. In *Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages*. Springer Verlag, 1996.
- [FMP96] Klaus Fischer, Jörg P. Müller, e Markus Pischel. A pragmatic BDI architecture. *Intelligent Agents II: Agent Theories, Architectures, and Languages*, páginas 203–218, 1996.
- [FN71] R. E. Fikes e N. Nilsson. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 5(2):189–208, 1971.
- [GdA01] Didier Guigue e Fábio Gomes de Andrade. Earle Brown's "25 pianos": a web interactive implementation. In *Anais do VIII Simpósio Brasileiro de Computação Musical, XXI Congresso Nacional da Sociedade Brasileira de Computação—SBC*, volume 1, páginas 776–783, 2001.
- [GM95] Masataka Goto e Yoichi Muraoka. Beat tracking based on multiple-agent architecture: A real-time beat tracking system for audio signals. In Victor Lesser, editor, *Proceedings of the First International Conference on Multi-Agent Systems—ICMAS'96*, páginas 103–110. MIT Press, 1995.

- [Gui97a] Didier Guigue. Sonic object: A model for twentieth-century music analysis. *Journal of New Music Research*, Vol. 26(Nº 4):346–375, 1997.
- [Gui97b] Didier Guigue. *Une étude “pour les Sonorités Opposées” : Pour une analyse orientée objets de l’œuvre pour piano de Debussy et de la musique du XXèsiècle*. Presses Universitaires du Septentrion, Villeneuve d’Ascq, 1997.
- [Had99] Karim Haddad. *OpenMusic: User’s Manual and Reference*. Paris, 1999.
- [HL84] Lejaren Hiller e Burt Levy. General systems theory as applied to music analysis - part i. In M. Baroni e L. Callegari, editores, *Musical Grammars and Computer Analysis*, páginas 295–316. Leo Olschki Editore, Florença, 1984.
- [HS98] Michael N. Huhns e Munindar P. Singh. *Readings in Agents*. Morgan Kaufmann, San Francisco, 1998.
- [Hur99] David Huron. *Music Research Using Humdrum: A User’s Guide*. CCARH, Menlo Park, 1999. Disponível em <http://www.music-cog.ohio-state.edu/Humdrum/guide.toc.html>.
- [Isa97] Eric Isaacson. Neural network models for the study of post-tonal music. In Marc Leman, editor, *Music, Gestalt and Computing: Studies in Cognitive and Systematic Musicology*, páginas 237–250. Springer Verlag, Berlin, 1997.
- [Jen93] N. R. Jennings. Specification and implementation of a belief desire joint-intention architecture for collaborative problem solving. *Journal of Intelligent and Cooperative Information Systems*, 2(3):289–318, 1993.
- [Les99] Victor R. Lesser. Cooperative multiagent systems: A personal view of the state of the art. *IEEE Transactions on Knowledge and Data Engineering*, 11(1):133–142, Jan/Fev 1999.
- [LO98] Danny B. Lange e Mitsuru Oshima. *Programming and Deploying Java Mobile Agents with Aglets*. Addison-Wesley, Reading (MA), 1998.
- [Mae91] P. Maes. The agent network architecture (ANA). *SIGART Bulletin*, 2(4):115–120, 1991.

- [Mal96] Mikhail Malt. *Introduction to Patchwork*. IRCAM, Paris, segunda edição, April 1996.
- [Mee94] Nicolas Meeùs. De la forme musicale et de sa segmentation. *Musurgia*, Vol. 1(Nº1):7–23, 1994.
- [Mes94] Marcel Mesnage. Techniques de segmentation automatique en analyse musicale. *Musurgia*, Vol. 1(Nº 1):39–49, 1994.
- [Met96a] Meta Software Corporation. *Design/CPN Reference Manual for X-Windows*, 1996.
- [Met96b] Meta Software Corporation. *Design/CPN Tutorial for X-Windows*, 1996.
- [Meu00] Benoît Meudic. *OpenMusic: Bibliothèque Kant*. Paris, 2000.
- [MH69] John McCarthy e Patrick J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer e D. Michie, editores, *Machine Intelligence*, volume 4, páginas 463–502. Edinburgh University Press, Edinburgh, 1969.
- [Mur89] Tadao Murata. Petri nets: Properties, analysis and applications. *Proc. of the IEEE*, 77(4):541–580, Abril 1989.
- [NS76] A. Newell e H. A. Simon. Computer science as empirical enquiry. *Communications of the ACM*, 19:113–126, 1976.
- [R+96] Curtis Roads et al. *The Computer Music tutorial*. The MIT Press, Cambridge (MA), 1996.
- [Ram97] Geber Lisboa Ramalho. *Construction d'un Agent Rationnel Jouant du Jazz*. PhD thesis, Universidade Paris VI, 1997.
- [Rat95] Miguel Ratton. *Criação de Música e Sons no Computador*. Campus, Rio de Janeiro, 1995.
- [Rei85] W. Reisig. *Petri Nets: An Introduction.*, volume 4 de *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, 1985.

- [RG91] Anand S. Rao e Michael P. Georgeff. Modeling rational agents within a BDI-architecture. In *Proc. of the International Conference on Principles of Knowledge Representation and Reasoning*, páginas 473–484, 1991.
- [RG95] Anand S. Rao e Michael P. Georgeff. BDI-agents: From theory to practice. In *Proceedings of the First International Conference on Multiagent Systems*, San Francisco, 1995.
- [RK93] Elaine Rich e Kevin Knight. *Inteligência Artificial*. Makron Books, 1993.
- [RN95] Stuart J. Russell e Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Englewood-Clifs (NJ), 1995.
- [Sea69] John R. Searle. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press, Cambridge, 1969.
- [Tei97] Luciênio de Macêdo Teixeira. Da Representação do Conhecimento Musical ao Esboço Conceitual de uma Sociedade de Agentes em Harmonia. Dissertação de mestrado, UFPB/CCT/COPIN, Campina Grande, 1997.
- [TGF00a] Ernesto Trajano, Didier Guigue, e Edilson Ferneda. Automatic Segmentation of Musical Flows: a Rational Agents Approach. In *Anais do VII Simpósio Brasileiro de Computação Musical, XX Congresso Nacional da Sociedade Brasileira de Computação—SBC*, volume 1. Champagnat, 2000.
- [TGF00b] Ernesto Trajano, Didier Guigue, e Edilson Ferneda. Segmentação automática de fluxos musicais: uma abordagem via agentes racionais. *Revista Eletrônica de Musicologia*, 5(2), 2000.
- [TL91] Peter Todd e D. Gareth Loy, editores. *Music and Connectionism*. MIT Press, Cambridge(MA), 1991.
- [Ull98] Jeffrey D. Ullman. *Elements of ML Programming*. Prentice Hall, Upper Saddle River, 1998.
- [VW97] Michael Van Wie. Thesis proposal: Observation and cooperation, 1997. Disponível em <http://www.cs.rochester.edu/u/vanwie/paper/paper.html>.

- [WFN<sup>+</sup>01] Rodolfo Daniel Wulfhorst, Luciano Vargas Flores, Lauro Nakayama, Cecília Dias Flores, Luís Otávio Campos Alvares, e Rosa Maria Vicari. An open architecture for a musical multi-agent system. In *Anais do VIII Simpósio Brasileiro de Computação Musical, XXI Congresso Nacional da Sociedade Brasileira de Computação—SBC*, volume 1, páginas 873–878, 2001.
- [WJ95] Michael Wooldridge e Nicholas R. Jennings. Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2):115–152, 1995.

# Apêndice A

## MIDI

Este apêndice explica, em linhas gerais, o protocolo MIDI, que foi utilizado nesta dissertação como linguagem de representação da música<sup>1</sup>.

### A.1 Introdução

Criado no início da década de 1980 com o propósito de controlar em tempo real equipamentos musicais, o protocolo MIDI—*Musical Instrument Digital Interface*—pode ser descrito de diversas maneiras: como esquema de interconexão entre instrumentos e computadores, como um conjunto de diretrizes para a transferência de dados entre instrumentos musicais ou como uma linguagem para a transmissão de partituras entre computadores e sintetizadores. Ele se baseia na codificação da informação em mensagens, que são transmitidas de um aparelho para outro. Exemplos de mensagem podem ser o início ou o término de uma nota.

Apesar de ter sido projetado para ser um protocolo para a transmissão de dados entre equipamentos musicais, o protocolo MIDI tem sido utilizado ao longo do tempo como um formato de arquivo. Para tanto, em 1988, foi padronizada uma extensão à especificação MIDI original chamada *Standard MIDI Files*—SMF. A principal diferença do SMF é que nele todo tipo de dado armazenado é temporizado (*time-stamped*). Isto significa que toda mensagem MIDI possui um tempo associado, uma posição no arquivo. Este rótulo de tempo é especificado em *clock ticks*, que é um número que representa a diferença temporal entre a mensagem corrente e a mensagem anterior. Isto possibilita que programa que está lendo o

---

<sup>1</sup>Para uma descrição completa do protocolo *vide* [Rat95; R<sup>+</sup>96]

arquivo saiba quando uma mensagem deve ser executada.

## A.2 Classificação das Mensagens MIDI

Como dito, o protocolo MIDI codifica a informação em mensagens, também chamada de evento MIDI. Devido à variedade no conteúdo das mensagens, estas são classificadas de acordo com o tipo de dado que elas representam:

- mensagens de canal: estas mensagens representam os eventos relacionados diretamente com a execução musical. Elas podem ser de dois subtipos:
  1. mensagens de voz: são as mensagens utilizadas para transmitir informações sobre notas, pedais, modificações de volume, etc.
  2. Mensagens de modo: são utilizadas para modificar o modo de operação dos equipamentos (passar do modo monofônico para o polifônico, por exemplo).
- Mensagens de sistema: este tipo mensagens representa informações não-musicais. Apesar de não tratarem de conteúdo musical, elas são importantes para o funcionamento de um sistema MIDI. Elas são dos seguintes subtipos:
  1. mensagens de tempo real: são as mensagens que servem para a sincronização dos equipamentos.
  2. Mensagens comuns: servem para posicionar os equipamentos no mesmo ponto, permitindo, por exemplo, que um teclado e uma bateria executem juntos a mesma seqüência rítmica.
  3. Mensagens exclusivas: espaço para que fabricantes especifiquem mensagens exclusivas para seus equipamentos (transferência de dados responsáveis pela geração do som, por exemplo). Além disso, elas têm sido utilizadas para a extensão das aplicações do protocolo, com controle da iluminação e de equipamentos de áudio e vídeo.

## A.3 Conteúdo das Mensagens MIDI

As mensagens MIDI classificadas na Seção A.2, são codificadas através de *bytes* que podem ser de dois tipos: *byte de status* e *byte de dados*. O *byte de status* codifica uma função em particular (ativação de uma nota, por exemplo) e é identificada através de seu primeiro *bit* que é o dígito 1. Já o *byte de dados* começa com o dígito 0 e provê os valores do *byte de status* ao qual ele está associado.

A Figura A.1 ilustra a anatomia de uma mensagem MIDI, a que codifica a ativação de uma nota (*note-on*). Ela consiste de três *bytes*: o primeiro (**Status**), um *byte de status* uma vez que ele se inicia com 1, indica a função desta mensagem. Este *byte* é lido da seguinte maneira: os quatro *bits* iniciais especificam a função da mensagem (*note-on*, neste caso) e os *bits* restantes especificam o canal MIDI onde a informação se encontra<sup>2</sup> (0000 = canal 1). Os dois *bytes* seguintes (**Dados1** e **Dados2**) são *bytes de dados*, pois se iniciam com o dígito 0. A quantidade de *bytes de dados* depende da função codificada no *byte de status*. Assim, uma mensagem pode possuir de dois a nenhum *byte de dados*. Neste caso, o primeiro *byte de status* codifica o valor da nota a ser ativada (no exemplo, o número MIDI 64, que corresponde à frequência de 330 Hz. ou mi 4, na notação americana) e o segundo codifica a intensidade ou velocidade com a qual a nota deve ser tocada (no exemplo, o valor MIDI 18).

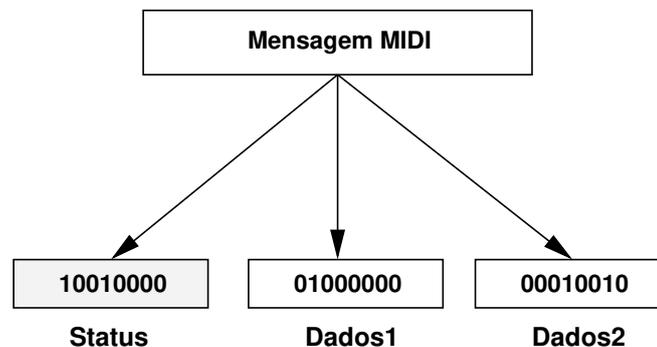


Figura A.1: Uma mensagem MIDI

Segue abaixo uma descrição das mensagens utilizadas nesta dissertação.

- Mensagens de canal—mensagem de voz:

<sup>2</sup>No protocolo MIDI é possível que os dados estejam distribuídos em 16 canais, possibilitando, por exemplo, a separação de vozes por canal.

- `note-on`: mensagem que indica a ativação de uma nota. Possui três *bytes*: um *status* e dois de dados referentes ao valor da nota e à velocidade da nota.
- `note-off`: mensagem que indica a desativação de uma nota. Também possui três *bytes*: um de *status* e dois de dados referentes ao valor da nota e à velocidade da nota<sup>3</sup>.
- `Control Change`: mensagem que indica a ativação ou desativação de um evento de controle (pedal, por exemplo). Possui três *bytes*: um de *status* e dois de dados, um representando o número do controlador a ser utilizado e o outro o valor<sup>4</sup>.

---

<sup>3</sup>Alguns sintetizadores, no lugar de um `note-off`, representam a desativação de uma nota através de um `note-on` com velocidade = 0.

<sup>4</sup>No sistema desenvolvido, o controlador utilizado é o controlador número 64, que é o valor MIDI para o pedal forte do piano *sustain pedal*. Convencionou-se que um valor = 0 representa o pedal desativado e que qualquer outro valor indica que o pedal se encontra ativado.