MATHEMATICAL MODEL OF INTERACTIVE PROGRAMMING GUIDE

Yakov Kamen
Entelel Corporation

Abstract

The classic interactive programming guide (IPG) was designed over 20 years ago using a grid data-presentation model. This design was perfectly suitable for a small number of homogeneous video channels and a short (few - hour-long) schedule. Today's IPG must manage over 300 heterogeneous video, PPV, VOD, and music channels in a two week schedule. It also has to manage time shifting (PVR) capabilities. The classic grid-based IPG was never designed to handle these tasks, and has to be significantly modified to reflect this new reality. The big question is how to modify the IPG so that it wins consumers' minds and solves the new problems? A mathematical model of the IPG is necessary to make the right decision.

This article describes the first mathematical model of the IPG based on the cognitive information theory. Different popular IPG solutions are analyzed and compared based on the proposed model.

IPG COMPONENTS AND STRUCTURE

TV Event Descriptions

The Interactive programming guide (or IPG) allows the viewer to view and manipulate TV schedule data directly on the TV screen. The schedule data can be described as a structured set $\{E\}$ of TV event metadata or events. Each event E consists of a channel ID that defines the event's channel, starting time, event length, event name, and event description. Formally an event E is defined as a structure:

$$E = < C_{ID}, T_S, \Delta T, E_N, E_D >, \qquad (1)$$

where

- E - is an event description,
- C_{ID} - is a channel ID, that may include the channel name, number, ID, etc.
- T_S - is an event's start time (time stamp)
- ΔT - is an event's length (in minutes)
- E_N - event's name (can be empty)
- E_D - event's description (can be empty or can be a complex structure of different multi-resolution description representations)

Note, that according to definition (1), when the same TV program is shown on different channels or at different times, it is considered two different TV events.

TV Channels

There are two types of channels in the IPG: regular (TV) channels and special "on-demand" (OD) channels. In the case of TV channels, all events are linearly ordered by time and can not intersect in the time domain. In the case of OD channels, events are not linearly ordered by time. In this article we consider all channels to be TV channels.

Major Components of IPG

Each IPG represents schedule data differently, but there are common rules that affect the guide's logical structure. For example, descriptions of events that have already passed are never shown on the screen. The conventional IPG that follows existing rules consists of the following components (Fig.1): sorting and searching control component, date and time, event listing (which consists of a subset of TV events described by their names), time and channel IDs, and the description of the highlighted event or dynamically updating help information [Kam01].

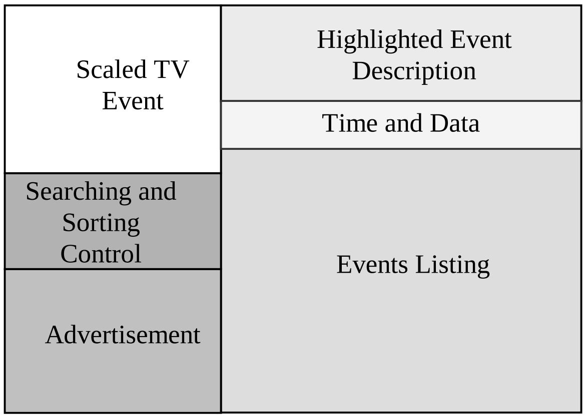

Fig1. Typical Components of an IPG

The Event listing is the most important component of the IPG, and therefore we will concentrate on its modeling and optimization. The "surfability" of the schedule data is the second most important component of the guide.

IPG OPTIMIZATION CRITERIA

Criteria specification

It is very difficult to define numerical IPG optimization criteria.

First of all, there are different formidable traditions of event listing presentation in different countries.

Second, depending on subscription package and location, different users have access to different channel packages, and, as a result, have different needs in an optimized IPG. For instance a user that watches 12 public broadcast channels would be satisfied with any IPG. However a user that is subscribed to 300+ channels and actively uses different recording devices (PVR, DVD, VCR) is significantly more sensitive to the IPG's efficiency.

Third, the description language significantly affects IPG event listing design, because a hieroglyphic language demands a different data presentation esthetic than alphabetic languages.

Fourth, different viewers have different TV-watching habits. The same person usually has different behavior patterns on working days, weekends, on vacation, and on holidays. These patterns are not stable. They tend to change over the years depending on health, family, and living conditions.

Fifth, optimization criteria must be "computable" and verifiable. This means that a criteria like "create an event listing such that all users will be happy" would not satisfy the goal of this work.

Sixth, the IPG user interface is, after all, a work of art. This means that the best and most useful solution may not be the most practical or ergonomic.

With all of the considerations above, the proposed mathematical model is based on a synthetic criteria C that consists of two separate criteria C_1 and C_2. The first criteria C_1, called "maximum listing information" criteria, estimates event listing information value. The second criteria C_2, called "minimum energy surfing" criteria, estimates the effort a user has to exert to find a TV program he would like to watch or record.

First we define criteria C_1 :

Formally, each event name listing (event listing) is a projection P of a subset of events on the screen. Each projected event in the listing is represented as

$$P(E) = < C_{ID}, T_S, \Delta T, P(E_N) >, \qquad (2)$$

where

$P(E_N)$ - is a projection of the event's name.

The criteria C_1 is defined as the maximization criteria comparing event listings by total information projected on the screen:

$$C_1 = \max_P \sum_1^{N(P)} P(E_i), \qquad (3)$$

where

$P(E_i)$ - is a projection of the i-event to the screen;

$N(P)$ - is the number of event names on the screen projected by P.

Now we define criteria C_2 .

The criteria C_2 is a minimization criteria that compares different IPGs by the average energy the user has to spend to go from an event E_0 to an event E_1. In this article we define an energy unit as a single key press of the remote controller. As a result the criteria C_2 allows us to find an IPG that requires a minimal average number of key presses to go from one arbitrary event name to another. To formalize criteria C_2 we define a distance function R in the event space such that $R(E_i, E_j)$, or the number of remote controller key presses needed to move the focus from event name E_i to the name E_j, is minimal. Lets assume that A(.) is an averaging operator as it has been defined in [Kam94]. The average distance between all pairs of events we will call "the IPG surfing diameter" or just IPG diameter. It is a good measure of IPG surfing energy. Formally, Criteria C_2 is an IPG diameter minimization criteria described as

$$C_2 = \min_R \underset{(i,j)}{A}(R(E_i, E_j)), \tag{4}$$

where

$A(x_1,\ldots x_n)$ - is an average function between $x_1,\ldots x_n$;

$R(x,y)$ - is the distance between objects x and y.

In the proposed mathematical model all IPG solutions are measured by the criteria C_1 and C_2.

ASSUMTIONS AND CONSTRAINTS

Any mathematical model is based on a set of basic assumptions and constraints that allow one to make non-trivial general conclusions.

Described below are the major assumptions and constraints of our IPG mathematical model.

- Homogeneous Event Value. All TV events in the schedule have the same priority value for all users. In the real world this assumption is not correct.
- Transmission Continuity. The current model assumes that all channels are always transmitted without interruption 24 hours per day, 7 days per week.
- Channel Structure. In the current model we assume that all available channels are TV channels where the events are linearly ordered. OD-channels do not exist in this model.
- Channel Distribution. A real user has his own list of "informative" channels and "non-informative", "noisy" or "garbage" channels. In the model below we assume that all channels are equally informative.
- Event Independence. Information located inside two arbitrary event descriptions is independent, i.e. for every two events E_1 and E_2 information I located in the pair of events is equal to the sum of information located in each event: $I(E_1 + E_2) = I(E_1) + I(E_2)$.
- Channel Independence. Information located inside two arbitrary channels is independent, i.e. for every two channels c_1 and c_2 information I located in the pair of channels is equal to the sum of information located in each channel: $I(c_1 + c_2) = I(c_1) + I(c_2)$.
- Semantic Equivalence. All descriptions that consist of the same number of symbols have equal amounts of information
- Event Information Equivalence. All event names viewed at the same time are equally important for users and consist of an equal amount of information

EVENT LISTING INFORMATION

Event Listing Modeling

There are numerous event listing models. The event listing information criteria C_1 measures the quantity of the information not its quality. From C_1 point of view all event names viewed at the same time are equally important for a user and consist of an equal amount of information I_0. For simplicity we set $I_0 = 1$. Each event name is projected on the screen into the event listing's fixed-sized "cell". If the cell is smaller than the event name, the event name is truncated and it looses some amount of information. The same video content has a different value to the user depending on whether it is already in progress, starting now, or will be playing in the future, because the starting time matters. Obviously, an event has the maximum value for the user if it is starting now. However it is a fairly rare case: usually events have already started (currently playing event) or will start in the future (future event). Both playing and future events have less information for the user than “starting now” events.

A major assumption of this mathematical model is that the total information value of the event listing is a sum of the information values of all projected events (2), and each projected listing event can be completely described as follows:

$$I_{P(E)} = I_{\mathrm{E}}(\mathrm{s}_{\mathrm{E}}, \mathrm{s}_{\mathrm{C}}) \cdot G(T_0, \Delta T, T_c) \cdot I_0,$$
$$I_{\mathrm{E}}(\mathrm{s}_{\mathrm{E}}, \mathrm{s}_{\mathrm{C}}) \in [0,1]; G(T_0, \Delta T, T_c) \in [0,1], \qquad (5)$$
$$I_0 = 1,$$

where

$I_E(s_E, s_C)$ - is a name value function that describes the amount of information that has been “left” in the original name after the projection P;

$G(T_0, \Delta T, T_c)$ -is a time value function that describes the information value that the current event has compared to the information value it would have if the event started immediately;

s_E -is the size of the event name;

s_C -is the size of the cell.

Most Popular Event Listing Models

Three examples below describe the most popular event listing organization schemes.

Example1. Grid based event listing. Grid data representation is the most popular listing design approach in the US. It consists of a set of time-proportional rectangular cells that are used to show event names. A simple example of a grid listing is shown in Fig. 2. An abstract description of the grid listing page is shown on Fig.3.

Fig.2 Grid based listing page

	T^1	T^2	T^3	T^4
C_1	$E_k(C_1)$		$E_{k+1}(C_1)$	$E_{k+2}(C_1)$
C_2	$E_k(C_2)$	$E_{k+1}(C_2)$	$E_{k+2}(C_2)$	
C_3	$E_k(C_3)$	$E_{k+1}(C_3)$	$E_{k+2}(C_3)$	$E_{k+2}(C_3)$
C_4	$E_k(C_4)$			
C_5	$E_k(C_5)$		$E_{k+1}(C_5)$	
C_6	$E_k(C_6)$			$E_{k+1}(C_6)$
C_7	$E_k(C_7)$		$E_{k+1}(C_7)$	

Fig. 3 Formal model of the grid listing

In the formal model (Fig.3) $C_1,\ldots, C_7$ are sequential channel IDs; $T^1,\ldots, T^4$ are standard time intervals (usually 30 minutes). $E_k(C_i)$ is the name of the event that is being transmitted on channel “i” during the time interval T^1. $E_{k+l}(C_i)$ is the name of the event that will be transmitted on channel “i” after the event with the name $E_k(C_i)$ at the “l”-step.

Example2. Link-list based event listing. The link-list listing shows the maximum number of events of the currently highlighted channel on the same screen. The Link-list solution is very useful for digital video recording (DVR or PVR) enabled systems. In this example we defines two link-list schemes based on wide and narrow cells. The first scheme (Fig. 4) is using wide cells to present event listings, the second scheme (Fig.5) is using narrow cells to present event names.

	T^1	C_3 page $_1$
C_1	$E_k(C_1)$	$E_{k+1}(C_3)$
C_2	$E_k(C_2)$	$E_{k+2}(C_3)$
C_3	$E_k(C_3)$	$E_{k+3}(C_3)$
C_4	$E_k(C_4)$	$E_{k+4}(C_3)$
C_5	$E_k(C_5)$	$E_{k+5}(C_3)$
C_6	$E_k(C_6)$	$E_{k+6}(C_3)$
C_7	$E_k(C_7)$	$E_{k+7}(C_3)$

Fig. 4 Link-list based event listing (wide cells)

	T^1	C_9 page $_1$	T^1	
C_1	$E_k(C_1)$	$E_{k+1}(C_9)$	$E_k(C_1)$	C_8
C_2	$E_k(C_2)$	$E_{k+2}(C_9)$	$E_k(C_2)$	C_9
C_3	$E_k(C_3)$	$E_{k+3}(C_9)$	$E_k(C_3)$	C_{10}
C_4	$E_k(C_4)$	$E_{k+4}(C_9)$	$E_k(C_4)$	C_{11}
C_5	$E_k(C_5)$	$E_{k+5}(C_9)$	$E_k(C_5)$	C_{12}
C_6	$E_k(C_6)$	$E_{k+6}(C_9)$	$E_k(C_6)$	C_{13}
C_7	$E_k(C_7)$	$E_{k+7}(C_9)$	$E_k(C_7)$	C_{14}

Fig.5 Link-list based event listing (narrow cells)

Example 3. Event Matrix Listing. This type of data representation is popular in some European countries (Fig.6).

	C_1		C_2
$t(E_k(C_1))$	$E_k(C_1)$	$t(E_k(C_2))$	$E_k(C_2)$
$t(E_{k+1}(C_1))$	$E_{k+1}(C_1)$	$t(E_{k+1}(C_2))$	$E_{k+1}(C_2)$
$t(E_{k+2}(C_1))$	$E_{k+2}(C_1)$	$t(E_{k+2}(C_2))$	$E_{k+2}(C_2)$
$t(E_{k+3}(C_1))$	$E_{k+3}(C_1)$	$t(E_{k+3}(C_2))$	$E_{k+3}(C_2)$
$t(E_{k+4}(C_1))$	$E_{k+4}(C_1)$	$t(E_{k+4}(C_2))$	$E_{k+4}(C_2)$
$t(E_{k+5}(C_1))$	$E_{k+5}(C_1)$	$t(E_{k+5}(C_2))$	$E_{k+5}(C_2)$
$t(E_{k+6}(C_1))$	$E_{k+6}(C_1)$	$t(E_{k+6}(C_2))$	$E_{k+6}(C_2)$

Fig. 6 Formal model of the matrix listing

In the figure above C_1 and C_2 are two sequential channel IDs; $E_k(C_i)$ is the name of the event that is being transmitted on channel "*i*" during the time interval T^1. $E_{k+m}(C_i)$ is the name of the event that is transmitted on channel "*i*" at the "*m*"-step; $t(E_k\ (C_i))$ is the starting time of the event "*k*" on the channel C_i.

How would one decide which name listing representation is more informative? This can be done by comparing information presented on the "average" page of the listing using formula (5) and its realization described below.

Name Value

At first glance it is beneficial to show as many event names on the same listing as possible. However, screen space is always limited and the visible part of the name inevitably shrinks when new "cells" are added to the screen.

The name value function (name value) estimates the amount of information left in an event name of size s_E after its projection into an event listing cell of size s_C.

Below we will define the name value as a monotonic function of two variables

$$I_E(s_E, s_C) = \begin{cases} 1, & \frac{s_C}{s_E} \geq 1 \\ 0 < f(\frac{s_C}{s_E}) < 1, & a < \frac{s_C}{s_E} < 1, \\ 0 & \frac{s_C}{s_E} < a \end{cases} \quad (6)$$

where

f(x) -is a monotonically increasing function;

a -is a threshold parameter $a \in [0,1]$ that defines the average loss of information that transforms data into noise.

For simplicity we will linearly approximate *f(x)* as:

$$f(d) = \frac{1}{1-a}d - \frac{a}{1-a}, \quad d = \frac{s_C}{s_E}, \quad (7)$$

As a result, formula (6) will look like the following (Fig.7):

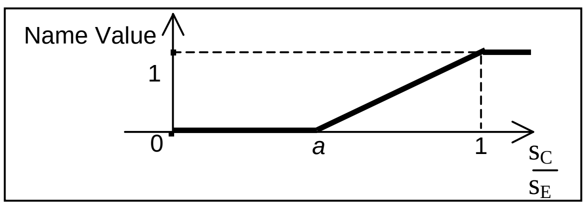

Fig.7 Name value function approximation

According to experimental data analysis a = 0.6 is a good approximation of the name threshold value for English-language based US TV schedule.

Time Value

As mentioned above, the information value of an event name in the event listing depends on the event's starting time. In the chosen model, the time value for future events monotonically decreases over time. The time value of the currently playing event is a monotonically growing function and as a result, the longer the event has been playing, the less value it has to the viewer.

The exact functional tie between the event starting time and its time value depends on many parameters, including subjective characteristics of the user, event's genre, structure, etc. Schematically the typical function may look like the following (Fig. 8)

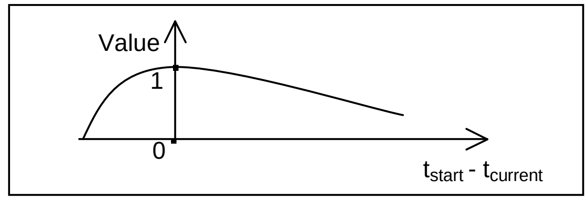

Fig.8 Time value

For simplicity of the model we assume that future events' time value decreases with a constant speed. Accepting this assumption, $G(T_0, \Delta T, T_c)$ can be approximated with the formula (8) below:

$$G(T_0, \Delta T, T_c) = \begin{cases} \left(\frac{T_c - T_0}{\Delta T}\right)^m \\ b^k, & k = \frac{T_c - T_0}{\Delta T} \end{cases} \quad (8)$$

where

m -is a parameter that defines time value degradation speed for currently playing events;

b -is a parameter that defines time value degradation speed for future events.

In practice, both information degradation speed parameters *m* and *b* would vary per user, time of day, or type of equipment (PVR). Based on experiments we found that m belongs to the interval [0.2, 1] and *b* belongs to the interval [0.7, 0.9]. Assuming that the currently airing event's start time is distributed uniformly we approximate (8) with:

$$G(T_0, \Delta T, T_c) = \begin{cases} 0.5^m \\ b^k, & k = \text{int}(\frac{T_c - T_0}{\Delta T}) \end{cases}, \quad (9)$$

where

int(x) -is the integer part of x.

Comparison of Models

With a few additional assumptions formulas (5)- (9) allow us to compare different models of event listings. In this article we compare models described in the examples above: “Grid”, “Link-list wide”, “Link-list narrow”, “Matrix wide”, and “Matrix narrow” . We will compute the average listing information values for the four cell sizes: 8, 10, 12, and 14. For comparison we assume that the standard time interval T is 30 minutes, and the number of visible cells is 28 (7x4). In the model comparison process we used empirical data collected from real US TV schedules (English language). This data includes a tabulated average name value function for each cell size, called a cell power table. A fragment of the cell power table is presented in Table 1.

Table1. Cell power table (fragment)

	cell size (symbols)				
	8	9	10	11	12
Name value	0.24	0.30	0.38	0.41	0.50

It also includes a tabulated histogram of event duration distribution (see Table2).

Table2. Event duration histogram

Event duration (min)	Percent (%)
1-30	56.8
31-60	20.7
61-120	10.4
>120	12.1

Final comparison results are presented in Table 3 below

Table3. Models Comparison

Listing type	cell size (symbols)			
	8	10	12	16
Grid	3.98	5.07	5.84	6.83
Link-list wide	4.09	5.64	6.40	7.58
Link-list narrow	2.29	3.96	6.49	9.17
Matrix wide	4.03	5.37	6.15	6.89
Matrix narrow	0.80	1.92	4.12	7.13

Table3 shows that there is no event listing model that is “the best” for all cell sizes.

GUIDE SURFING

Surfing Control

The minimum energy surfing criteria C_2 would benefit IPG solutions that use a lot of special keys that “short cut” the most popular step sequences. But the idea of improving the surfing experience by adding special keys does not work. First, screen space and remote controller buttons are limited. Second, it is impossible to convince a user to learn an “F-16 cockpit” style remote controller to surf TV in the dark. To make the minimal energy surfing criteria meaningful, we assume that all designed models must use the same minimal set of surfing keys: up, down, left, right, select, and ten digits 0-9.

Channel and Time Distance

Without limitations we would consider that the distance $R(E_i, E_j)$ between two arbitrary events E_i and E_j, used in formula (4), is a “manhattan” metric in the channel/time coordinate space. In other words,

$$R(E_i, E_j) = R^C(E_i, E_j) + R^T(E_i, E_j), \qquad (10)$$

where

$R^C(E_i, E_j)$ -is the distance between the channels of events E_i and E_j ;

$R^T(E_i, E_j)$ -is the distance between the times of events E_i and E_j.

In formula (10) $R^C(.)$ is called a channel distance and $R^T(.)$ is called a time distance.

Users’ Tasks and Models

Users surf the IPG to solve three main tasks:

Task A. Find something to watch now.

Task B. Find something to watch soon.

Task C. Find something to record or to watch in the future.

In the case of task A, time distance is equal to zero and only the channel distance has to be

estimated. When the user knows the channel number, the optimal surfing solution is dialing that channel number. In this case, the IPGs diameter is equal to the average number of digits in a channel number.
When the desired channel number is not known but the channel name is, task A is to tune to the channel based on its name with the minimal number of key presses. Assuming that channels are uniformly distributed, and that the name listing is the only surfing solution, the expected IPG diameter is approximated with the following formula:

$$A_{(i,j)}(R(E_i,E_j)) = A_{(i,j)}^{C}(R(E_i,E_j)) \approx \frac{0.5K}{N} + 0.5N, \qquad (11)$$

where
K -is the total number of channels;
N -is the number of channels visible on the event listing.

Function (11) achieves its minimum when $N = \sqrt{K}$ and it is equal to N. When N is close to $\sqrt{K}$, simple event listing scrolling is the optimal surfing method in the channel domain. However when the difference is large enough, there are additional opportunities to minimize the channel diameter by implementing multi-resolution data representation modules. The simplest idea of multi-resolution channel list representation is the idea of a “channel matrix” (Fig.7, courtesy iSurfTV Corporation).

Fig.7 Channel Matrix example

The channel matrix module uses screen space to show the maximum number (L) channel IDs (in visual or textual format) on the screen ($L>N$). The channel matrix (Fig. 7) has almost no information about the playing event names. This means that the user has to surf inside the matrix page to check several channels before he will make his decision to switch to a channel. The channel diameter of an IPG that includes a channel matrix module can be approximated with the formula:

$$A_{(i,j)}^{C}(R(E_i,E_j)) \approx \frac{0.5K}{L} + 0.5L, \qquad (12)$$

where
L -is the number of channel IDs in the channel matrix ($L>N$).

Note that diameter (12) is smaller than diameter (11) only if L is closer to $\sqrt{K}$ than N, i.e. if $N < L < 2\sqrt{K} - N$. This means that the matrix module would improve the surfing experience only in the case of a large number of playing TV channels.
Another solution is to create a new module that stores channel IDs alphabetically in the “notebook” style. Using the “optimal” notebook module channel diameter can be decreased to:

$$A_{(i,j)}^{C}(R(E_i,E_j)) \approx \sqrt{K} + 2 \qquad (13)$$

Let us now analyze tasks B and C. Both of them require the user to surf in the time domain. Below we will compute the time diameter in both tasks B and C. As with (11) we will approximate the time diameter of the linear event listing, when surfing in the time domain, with the following formula:

$$A_{(i,j)}^{T}(R(E_i,E_j)) \approx \frac{0.5K}{N} + 0.5N, \qquad (14)$$

where
K -is the total length of the schedule in hours (K varies from 1 to 720 hours);
N -is the average period of time visible at the event listing (N varies from 0.75 to 6 hours).

Formula (14) allows us to formalize the concept of "playing soon" events as events that would start playing in the time interval when the linear time surfing is the most optimal solution. Formally this time interval is defined as the interval $[T_C; T_C+N^2+N]$, where T_C is the current time.

Based on the definition of "playing soon" events we will estimate the time dimension in task B based on formula (14).

The optimal solution of task C is based on a multi-resolution time representation. In this model we will analyze three competing implementations.

The first implementation is a homogeneous grid that positions days on the first dimension and time intervals on the second dimension. The time diameter in this approach could be approximated with the formula:

$$A^T_{(i,j)}(R(E_i,E_j)) \approx \frac{K}{48}+\frac{12}{N}+1 \tag{15}$$

The second implementation is a set of two screens: a day listing screen, and a screen with 12 one-hour intervals. The time diameter is approximated by:

$$A^T_{(i,j)}(R(E_i,E_j)) \approx \frac{K}{336}+13 \tag{16}$$

The third implementation is a set of three layers: day, part of the day (morning, day time, prime time, evening, etc.), and one hour time intervals inside each part of the day. The time diameter in this implementation is fractionally smaller than diameter (16).

Comparing formula (14), (15) and (16) we can conclude that solution (14) is preferable for a very short schedule (less than 2 days), solution (15) is preferable when the schedule fluctuates between 2 and 10 days and solution (16) is preferable when the schedule is longer than 10 days.

FUTURE DEVELOPMENT

In this article we presented the first mathematical model of an interactive programming guide. A lot of assumptions and constrains make the usability of this model limited in practice. Therefore many of these constrains can be waved without serious complications. A model's complication would be compensated by its improved practicality. The maximal event listing information and minimal energy surfing criteria also can be generalized and improved. For instance, a channel's probability of being watched can be added to the event listing information criteria. In the minimal energy surfing criteria we can replace the "key press", as a measurement unit, with time.

Several important questions had not been discussed in this article. For example a model's robustness to the parameter's variation is extremely important for practical implementation.

REFERENCES

[Kam 01] Yakov Kamen, "Components and Advanced Functionalities of the Next Generation Interactive Programming Guide," Proc. "The 2nd Annual Digital TV Application Software Environment (DASE) Symposium 2001:End-to-End Data Services, Interoperability & Applications", Gaithersburg, June19-20, 2001, p.307-316

[Kam 94] Yakov Kamen and Sergei Ovchinnikov "Meaningful Means on Ordered Sets", Proc. "Fifth International Conference IPMU", Paris, July 4-8, 1994, p.1009-1013